

UNIVERSIDADE ESTADUAL DE MONTES CLAROS

Roberto César Faria e Silva

Desenvolvimento e validação de *softwares* para análise estatística em Ciências da Saúde

Montes Claros – Minas Gerais

2022

Roberto César Faria e Silva

Desenvolvimento e validação de *softwares* para análise estatística em Ciências da Saúde

Exame de defesa da tese de Doutorado, apresentada ao Programa de Pós-graduação em Ciências em Saúde (PPGCS) da Universidade Estadual de Montes Claros (Unimontes), como parte das exigências para a obtenção do título de Doutor em Ciências da Saúde.

Área de Concentração: Mecanismo e Aspectos clínicos das doenças.

Orientador: Prof. Dr. Renato Sobral Monteiro Júnior
Coorientadora: Prof.^a Dra. Marise Fagundes Silveira

Montes Claros – Minas Gerais

2022

Silva, Roberto César Faria e.

S586d Desenvolvimento e validação de softwares para análise estatística em Ciências da Saúde [manuscrito] / Roberto César Faria e Silva. – Montes Claros (MG), 2022.
1513 f. : il.

Inclui bibliografia.

Tese (Doutorado) - Universidade Estadual de Montes Claros - Unimontes, Programa de Pós-Graduação em Ciências da Saúde /PPGCS, 2022.

Orientador: Prof. Dr. Renato Sobral Monteiro Júnior.

Coorientadora: Profa. Dra. Marise Fagundes Silveira.

1. Software - Desenvolvimento. 2. Software - Validação. 3. Análise estatística.
4. Ciências médicas. I. Monteiro Júnior, Renato Sobral. II. Silveira, Marise Fagundes.
III. Universidade Estadual de Montes Claros. IV. Título.

UNIVERSIDADE ESTADUAL DE MONTES CLAROS-UNIMONTES

Reitor: Antônio Alvimar de Souza

Vice-reitora: Ilva Ruas de Abreu

Pró-reitora de Pesquisa: Clarice Diniz Alvarenga Corsato

Coordenadoria de Acompanhamento de Projetos: Virgílio Mesquita Gomes

Coordenadoria de Iniciação Científica: Sônia Ribeiro Arruda

Coordenadoria de Inovação Tecnológica: Sara Gonçalves Antunes Souza

Pró-reitor de Pós-graduação: André Luiz Sena Guimarães

Coordenadoria de Pós-graduação *Lato-sensu*: Allysson Steve Mota Lacerda

Coordenadoria de Pós-graduação *Stricto-sensu*: Marcos Flávio Silveira Vasconcelos

D'Ângelo

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS DA SAÚDE

Coordenadora: Cristina Andrade Sampaio

Subcoordenador: Renato Sobral Monteiro Júnior



GOVERNO DO ESTADO DE MINAS GERAIS

Universidade Estadual de Montes Claros

Mestrado e Doutorado em Ciências da Saúde

Universidade Estadual de Montes Claros

Anexo nº Folha Aprovação - Roberto César Faria e Silva/UNIMONTES/PRPG/PPGCS/2022

PROCESSO Nº 2310.01.0010198/2022-05

FOLHA DE APROVAÇÃO

Data da Defesa: 16/12/2022 - webconferência, via plataforma "Meet"

NOME DO(A) DISCENTE: ROBERTO CÉSAR FARIA E SILVA

() Mestrado Acadêmico em Ciência Da Saúde

(x) Doutorado Acadêmico em Ciências Da Saúde

TÍTULO DO TRABALHO DE CONCLUSÃO DE CURSO (TCC):

"DESENVOLVIMENTO DE FERRAMENTAS PARA ANÁLISE ESTATÍSTICA EM CIÊNCIAS DA SAÚDE"

ÁREA DE CONCENTRAÇÃO: Mecanismos e aspectos clínicos das doenças

LINHA DE PESQUISA: Clínica, Diagnóstico e Terapêutica das Doenças

BANCA (TITULARES)

Prof. Dr. Renato Sobral Monteiro Júnior por videoconferência)	ORIENTADOR	(participação à distância
Prof. ^a Dr. ^a Marise Fagundes Silveira videoconferência)	COORIENTADOR	(participação à distância por
Prof. Dr. Marcos Flavio Silveira Vasconcelos D'angelo videoconferência)		(participação à distância por
Prof. ^a Dr. ^a Sara Gonçalves Antunes de Souza videoconferência)		(participação à distância por
Prof. ^a Dr. ^a Paula Cristina Pelli Paiva videoconferência)		(participação à distância por
Prof. Dr. Rodolpho César dos Reis TININI videoconferência)		(participação à distância por

BANCA (SUPLENTES)Prof.^a Dr.^a Ana Carolina de Mello Alves Rodrigues

Prof. Dr. Carlos Renato Theóphilo

Prof. Dr. João Marcus Oliveira Andrade

A análise realizada pelos membros examinadores da presente defesa pública de TCC teve como resultado parecer de:

APROVAÇÃO REPROVAÇÃO

A efetiva conclusão do curso está sujeita à entrega da versão final da tese com as alterações recomendadas pela banca avaliadora no prazo de 30 dias.

	Documento assinado eletronicamente por Renato Sobral Monteiro Junior, Professor de Educação Superior, em 28/12/2022, às 11:00, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 47.222, de 26 de julho de 2017 .
	Documento assinado eletronicamente por Marcos Flavio Silveira Vasconcelos Dange, Professor de Educação Superior, em 28/12/2022, às 11:56, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 47.222, de 26 de julho de 2017 .
	Documento assinado eletronicamente por Marise Fagundes Silveira, Professora de Educação Superior, em 28/12/2022, às 16:19, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 47.222, de 26 de julho de 2017 .
	Documento assinado eletronicamente por Rodolpho César dos Reis Tinini, Usuário Externo, em 29/12/2022, às 09:29, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 47.222, de 26 de julho de 2017 .
	Documento assinado eletronicamente por Paula Cristina Pelli Paiva, Usuário Externo, em 16/02/2023, às 09:13, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 47.222, de 26 de julho de 2017 .
	Documento assinado eletronicamente por Sara Goncalves Antunes de Souza, Professor(a), em 16/02/2023, às 20:44, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 47.222, de 26 de julho de 2017 .
	A autenticidade deste documento pode ser conferida no site http://sei.mg.gov.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador 56855867 e o código CRC C5DFC983 .

Dedico esse trabalho a todos os que contribuíram de forma direta ou indireta para a conclusão deste, e àqueles que se beneficiarão dos produtos aqui desenvolvidos.

AGRADECIMENTOS

À minha esposa, Dayane, e meus filhos João Lucas e Antônio por estarem ao meu lado, apoiando-me nos momentos mais difíceis.

Aos meus pais, Expedito e Ana Maria, que me incentivaram no caminho do estudo.

Ao meu irmão Dr. André Luís, que, com seu exemplo, incentivou-me no estudo acadêmico.

Ao meu amigo e irmão Ismael por todo apoio de que precisei.

Ao meu orientador, Dr. Renato Sobral Monteiro Júnior, pela coragem em aceitar um projeto desafiador e fora da sua principal área de pesquisa e por ter me aceitado nesse processo.

A minha coorientadora Marise Fagundes Silveira pelo apoio e ensinamentos estatísticos que foram fundamentais para esse trabalho.

Aos membros de minha banca, que contribuíram para este trabalho com seus olhares críticos e repletos de experiência.

Aos meus colegas de doutorado e do grupo GENESEs, pelo convívio e amizade.

A todos os meus professores, que tanto me ensinaram.

“A genialidade é 1% inspiração e 99% transpiração.”

Thomas Edison

APRESENTAÇÃO

Cálculo sempre foi o meu forte. Desde criança gostava de números e, na escola, desempenhava a função de monitor dos professores de matemática. Ainda criança, estudei computação na Escola Técnica de Montes Claros, onde aprendi linguagens de programação como DOS, Basic e Cobol. Foi nesse período que também aprendi sobre lógica de programação, o que me foi muito útil para o desenvolvimento deste trabalho.

Minha formação acadêmica se iniciou em 1994, no curso de Administração da Universidade Estadual de Montes Claros (Unimontes), no período noturno, o qual concluí em 1998. Em 1995, iniciei também na Unimontes o curso de Ciências Contábeis, no período matutino. Até o ano de 1998, fazia os dois cursos e trabalhava no período da tarde em uma instituição financeira. Em 1999, recebi uma proposta para trabalhar viajando e acabei abandonando o curso de Ciências Contábeis, que só retomei em 2013 e concluí em 2014.

Em 2001, concluí uma especialização em Gestão Estratégica (Administração) e em 2012 concluí meu mestrado, também em Administração. Em 2009, comecei a lecionar na Faculdade Prisma e Faculdade do Norte de Minas – Funorte. Trabalhei nas Faculdades Pitágoras, Santo Agostinho e entrei na Unimontes no Ano de 2012 como professor designado. Em 2014, fui aprovado no concurso público e tomei posse em 2015, como professor concursado na área financeira do curso de Administração, onde leciono até o momento.

Em 2018, conheci o Dr. Renato Sobral Monteiro Junior e, na ocasião, comecei a cursar disciplinas isoladas no Programa de Pós-graduação em Ciências da Saúde – PPGCS (Unimontes) e também a participar do grupo de pesquisa GENESEs (Grupo de Estudos e Pesquisas em Neurociência, Exercício, Saúde e Esporte). Nesse grupo de pesquisa, me foi dada a função de auxiliar os demais alunos no tratamento de dados e nos cálculos estatísticos. Assim, surgiu a ideia de desenvolver minha tese em uma solução estatística, nascendo então a proposta do Oráculo Estatístico.

No segundo semestre de 2019, entrei para o PPGCS através do fluxo contínuo e tive como Orientador o Professor Dr. Renato Sobral Monteiro Junior. No primeiro semestre de 2021, durante as aulas de Bioestatística I, com as professoras Dra. Marise Fagundes

Silveira (Coorientadora deste trabalho) e Dra. Desirée Sant’Ana Haikal, foi desenvolvido uma planilha para cálculo amostral que, por sugestão do meu orientador, se tornou outro produto desta tese, o PPGCS *Sample Size*.

Assim, nesse trabalho apresento o resultado do meu doutorado, que resultou no desenvolvimento de dois *softwares* estatísticos. O primeiro, intitulado “Oráculo Estatístico”. Trata-se de um *software* desenvolvido com a finalidade de realizar testes de hipóteses para comparação de médias. O segundo produto, denominado “PPGCS *Sample Size*” é uma calculadora desenvolvida para calcular o tamanho amostral para comparação de médias e de proporções.

RESUMO

O uso das ferramentas estatísticas é de fundamental importância para validação e associação de eventos nas mais diferentes áreas do conhecimento. Dentre elas a da saúde, onde se é possível verificar a eficiência de tratamentos ou procedimentos. Contudo, a utilização de *softwares* torna-se limitada em detrimento de interfaces pouco intuitivas, altos custos dos programas disponíveis, bem como a dificuldade dos usuários na escolha das análises adequadas ao delineamento, isso associado à dificuldade na interpretação dos dados. Nessa perspectiva, torna-se oportuno a criação de *softwares* mais intuitivos que auxiliam na tomada de decisão de seus usuários e garantem a qualidade dos dados obtidos. Neste sentido, diante da lacuna existente, o presente estudo objetivou desenvolver *softwares* intuitivos voltados para análises estatísticas em ciências da saúde. Para isso, foram desenvolvidos dois *softwares* estatísticos. Um deles, intitulado Oráculo Estatístico, permite resolver problemas relacionados a testes de hipóteses. O outro, denominado PPGCS *Sample Size*, possibilita calcular o tamanho da amostra mínima necessária para estudos que envolvem médias e proporção. A metodologia utilizada para o desenvolvimento dos *softwares* seguiu a abordagem para o ciclo de vida, o *Rational Unified Process* (RUP) e prototipagem. Os requisitos necessários foram documentados utilizando a notação proposta pela *Unified Modeling Language* (UML). O Oráculo Estatístico foi desenvolvido empregando a linguagem C#, enquanto o PPGCS *Sample Size* foi desenvolvido na linguagem PHP (*Hypertext Preprocessor*) - camada visual HTML 5 (*Hypertext Markup Language*). Os dois *softwares* foram testados por meio da comparação dos resultados das análises obtidas pelas ferramentas propostas com aqueles obtidos por outros *softwares*, como o SPSS, o R e o Excel, tendo sido observada concordância de 100%. Em teste de usabilidade comparado com o PSPP (Free Software Foundation), os usuários do Oráculo Estatístico conseguiram resolver o problema proposto com um tempo 46% menor em relação ao PSPP. Ademais o índice de acerto com o Oráculo Estatístico foi de 82% contra 12% do PSPP, além de receber também uma nota melhor nos quesitos: aprendizado, eficiência, segurança e satisfação. Os programas criados apresentam interfaces intuitivas para usuários, além de possuírem janelas educativas com justificativas do uso do teste recomendado, bem como suas referências. É importante destacar, que o programa estatístico Oráculo Estatístico tem como controle de qualidade das análises a retenção na tomada de decisões, ou seja, impede que o usuário faça escolhas erradas impedindo análises em desacordo com o delineamento recomendado para a amostra em questão. Para acesso e difusão, os programas se encontram disponíveis no site do PPGCS pelo endereço <http://www.ppgcs.unimontes.br>. Conclui-se que as ferramentas criadas são fidedignas para as análises propostas, de fácil manuseio e práticas.

Palavras-chave: *Software*. Análise. Interpretação Estatística de Dados.

ABSTRACT

The use of statistical tools is of fundamental importance for the validation and association of events in different areas of knowledge. In the health area, it is possible to verify the efficiency of treatments and procedures. However, machine use is limited because there are few unintuitive interfaces, high costs of available programs, and users' difficulty in choosing the appropriate analysis. From this perspective, it is opportune to create more intuitive software to assist users' decision-making, ensuring the data's quality. Given the existing gap, the present study aimed to develop tools for statistical analyses in Health Sciences. For this means, two statistical software were developed. One of them, entitled Statistical Oracle, solves problems related to hypothesis tests. The other one, PPGCS Sample Size, allows calculating the minimum sample size required for studies involving means and proportion. The methodology for software development followed the lifecycle approach, Rational Unified Process (RUP), and prototyping. The requirements were documented with the notation proposed by the Unified Modeling Language (UML). The Statistical Oracle was developed with C# language, while the PPGCS Sample Size was developed in the PHP (Hypertext Preprocessor) language - visual layer HTML 5 (Hypertext Markup Language). The two software scans were tested by comparing the results of the analyses by the tools proposed with those obtained by other softwares, such as SPSS, R, and Excel - 100% agreement was observed. In a usability test compared to the PSPP (Free Software Foundation), the users of the Statistical Oracle were able to solve the proposed problem with a 46% lower time compared to the PSPP. In addition, the success rate with the Statistical Oracle was 82% against 12% for the PSPP, in addition to also receiving a better score in the categories: learning, efficiency, safety and satisfaction. The programs present intuitive interfaces for users; besides, it has educational windows that justify the use of a recommended test and its references. It is essential to highlight that Oracle has quality control of the analyses that may influence the researcher's decision-making. That prevents the user from making wrong choices in the analysis when it is not following the recommended design for the sample. For access and dissemination, programs are available at the PPGCS website at the <http://www.ppgcs.unimontes.br> . It is concluded that the tools created are reliable for the analyses, easy to handle and practical.

Keywords: Software. Analysis. Data Interpretation, Statistical.

LISTA DE ILUSTRAÇÕES

Produto 1 – Software: PPGCS Sample Size

Figura 1 - Protótipo.	28
Figura 2 - Trello.....	29
Figura 3 – Linguagens utilizada na programação.....	30
Figura 4 – Parte do código fonte.	30
Figura 5 – Git.....	31
Figura 6 – Explicação de uso.....	32
Figura 7 – Dica de tela.....	33
Figura 8 – Vídeo com instruções de uso do ppgcs sample size.....	33
Figura 9 – Protótipo desenvolvido no excel.	35
Figura 10 – PPGCS <i>Sample Size</i> – tela inicial.	36
Figura 11 – PPGCS <i>Sample Size</i> – tela inicial (continuação i).	37
Figura 12 – PPGCS <i>Sample Size</i> – tela inicial (continuação ii).	37
Figura 13 – PPGCS <i>Sample Size</i> – estimativa para média – população infinita.	38
Figura 14 – PPGCS <i>Sample Size</i> – caixa com explicação.	39
Figura 15 – PPGCS <i>Sample Size</i> – cálculo.....	40
Figura 16 – PPGCS <i>Sample Size</i> – aba sobre.	41
Figura 17 – PPGCS <i>Sample Size</i> – referências.....	42
Figura 18 – Cálculo amostral no excel.	43

Produto 2 – Software: Oráculo Estatístico

Figura 1 - Rational Unified Process (RUP).	48
Figura 3 – Teste de unidade para o teste de mann whitney.....	53
Figura 4 – Trello.....	55
Figura 5 – C#.....	55
Figura 6 – Git.....	56
Figura 7 – Vídeo com instruções de uso.....	57
Figura 8 - Logomarca do software oráculo estatístico.	58
Figura 9 – Tela inicial do software.....	63
Figura 10 – Tela inicial de dados.	64
Figura 11 – Importação de arquivos do excel.....	65

Figura 12 – Escolha da planilha.	66
Figura 13 – Mensagem de configuração dos dados.....	67
Figura 14 – Dados importados.	68
Figura 15 – Mensagem de confirmação.	69
Figura 16 – Dados para análise.....	70
Figura 17 – Organização dos dados.....	71
Figura 18 – Analisar.	72
Figura 19 – Análise de uma variável.	73
Figura 20 – Resultado da análise de uma variável.	74
Figura 21 – Resultado da análise de uma variável.	75
Figura 22 – Análise de mais de uma variável.....	76
Figura 23 – Resultado da análise de três variáveis (parte 1).	77
Figura 24 – Resultado da análise de três variáveis (parte 2).	78
Figura 25 – Resultado da análise de três variáveis (parte 3).	79
Figura 26 – Resultado da análise de três variáveis (parte 4).	80
Figura 27 – Resultado da análise de três variáveis (parte 5).	81
Figura 28 – Resultado da análise de três variáveis (parte 6).	82
Figura 29 – Explorando a aba sobre.	83
Figura 30 – Teste de normalidade no r.	88
Figura 31 – Teste de igualdade de variância no r.	89
Figura 32 – ANOVA no r.	90
Figura 33 – ANOVA no r.	92

Produto 3 – Oráculo Estatístico: uma nova ferramenta de análise

Figura 1. Comparação entre os tempos gastos	99
Figura 2. Percentual de acertos entre os usuários do programas avaliados pspp e oráculo estatístico.	100

LISTA DE TABELAS

Produto 1 – Software: PPGCS Sample Size

Quadro 1 – cálculos amostrais que o PPGCS <i>Sample Size</i> Realiza.	31
--	----

Produto 2 – Software: Oráculo Estatístico

Tabela 1 – descrição das classes do sistema.....	51
Tabela 2 – Validação de testes de hipótese do software Oráculo Estatístico com o R e o PASW.....	85
Tabela 3 – Dados utilizados para a análise de validação do Oráculo Estatístico.	85
Tabela 4 - Estatística descritiva realizada no Excel.	86
Tabela 5 - Estatística descritiva realizada no <i>software</i> Oráculo Estatístico.	87
Tabela 5 - Teste de normalidade realizado no <i>software</i> Oráculo Estatístico.....	88
Tabela 7 – Teste de igualdade de variância realizado no <i>software</i> Oráculo Estatístico..	89
Tabela 8 – Teste de igualdade de variância no PASW.....	89
Tabela 9 - ANOVA no Oráculo Estatístico.	89
Tabela 10 - ANOVA no Excel.	90
Tabela 11 – ANOVA no PASW.....	91
Tabela 12 - Teste de Bonferroni no Oráculo Estatístico.	91
Tabela 13 - Teste de Bonferroni no PASW.....	92
Tabela 14 - Teste de Bonferroni no Excel.	93

Produto 3 – Oráculo Estatístico: uma nova ferramenta de análise

Quadro 1. Comparação de resultados entre os <i>softwares</i>	98
Tabela 2. Comparação dos critérios de usabilidade entre os softwares PSPP e oráculo	101

LISTA DE ABREVIATURAS E SIGLAS

CSS	Do inglês, <i>Cascading Style Sheets</i> , traduzido para o português como estilo de folhas em cascata.
CSV	Do inglês, <i>Character-separated values</i> , traduzido para o português como valores separados por caracteres.
GENESEs	Grupo de Estudos e Pesquisa em Neurociência, Exercício, Saúde e Esporte.
HTML	Do inglês, <i>Hypertext Markup Language</i> , traduzido para o português como linguagem de marcação de texto.
IDE	Do inglês, <i>Integrated Development Environment</i> , traduzido para o português como ambiente de desenvolvimento integrado.
IHC	Interação Humano-Computador.
IMRS	Índice mineiro de responsabilidade social.
INPI	Instituto Nacional da Propriedade Industrial.
NIT	Núcleo de Inovação Tecnológica.
ORCID	<i>Open Researcher and Contributor ID</i> .
PASW	Do inglês, <i>Predictive Analytics Software</i> (PASW), traduzido para o português como software de análise preditiva.
PHP	Do inglês, <i>Hypertext Preprocessor</i> , traduzido para o português como processador de hipertexto.
PPGCS	Programa de Pós Graduação em Ciências da Saúde.
RUP	Do inglês, <i>Rational Unified Process</i> , traduzido para o português como processo racional unificado.
SPSS	Do inglês, <i>Statistical Package for the Social Science</i> , traduzido para o português como pacote estatístico para ciências sociais.
UML	Do inglês, <i>Unified Modeling Language</i> , traduzido para o português como linguagem de modelagem unificada.
UNIMONTES	Universidade Estadual de Montes Claros

SUMÁRIO

1 INTRODUÇÃO	19
2 OBJETIVO	24
2.1 Objetivo geral	23
2.2 Objetivos específicos	23
3 PRODUTOS TÉCNICOS CIENTÍFICOS.....	24
3.1 Produto 1 – Calculadora: PPGCS SAMPLE SIZE.....	26
INTRODUÇÃO.....	26
Objetivo	26
Público-alvo.....	27
METODOLOGIA.....	27
Abordagem	27
Identificação de requisitos	28
Linguagem de Programação	29
Aperfeiçoamento e registro	30
Testes	31
Instalação e instruções de uso.....	32
RESULTADOS E DISCUSSÃO	34
Apresentação	34
Desenvolvimento	34
Testes Realizados	42
CONCLUSÃO.....	44
3.2 Produto 2 – SOFTWARE: ORÁCULO ESTATÍSTICO	45
INTRODUÇÃO.....	45
Objetivo	47
Público-alvo.....	47
METODOLOGIA.....	47
Abordagem	47

Identificação dos requisitos	48
Linguagem de programação	54
Aperfeiçoamento e registro	55
Testes	56
Instalação e instruções de uso.....	57
RESULTADOS E DISCUSSÃO	58
Apresentação do Software	58
Desenvolvimento do Oráculo Estatístico	59
Telas do oráculo estatístico.....	63
Relatório do Oráculo Estatístico.....	83
Validação do oráculo estatístico	84
CONCLUSÃO.....	93
3.3 Produto 3 – Artigo: Oráculo estatístico: uma nova ferramenta de análise	94
1 INTRODUÇÃO.....	95
2 MÉTODO	95
2.1 Procedimentos	96
2.2 Questionários	96
2.3 Análise dos dados	97
3 RESULTADOS	97
3.1 Software Oráculo Estatístico	97
3.2 Tempo utilizado na Análise.....	99
3.3 Usabilidade dos <i>softwares</i>	100
4 DISCUSSÃO	102
5 CONCLUSÃO.....	103
REFERÊNCIAS	103
4 CONSIDERAÇÕES FINAIS.....	109
REFERÊNCIAS	111
ANEXOS	115
Anexo A – Certificado do Registro do Oráculo Estatístico Junto ao INPI.	115
Anexo B – Certificado do Registro do PPGCS Sample Size Junto ao INPI.	116
Anexo C – Dados do Pesquisador O.T.S. (grupo GENESEs).....	117

Anexo D – Dados do Índice Mineiro de Responsabilidade Social.....	118
Anexo E: Declaração de revisão linguística	139
Apendice A : Diagrama de Classes	140
Apendice B : <u>Bibliometrics of CNS & Neurological Disorders – Drug Targets: An International Evolution Along Time</u>	141
Apendice C : Código Fonte PPGCS Sample Size	141
Apendice D : Código Fonte do Oráculo Estatístico	280

1 INTRODUÇÃO

A história da estatística remonta desde 2000 a.C. com a catalogação de bens e de povos pelos chineses bem como os egípcios e astecas (1). Desde então, aconteceu um grande desenvolvimento da complexidade dos cálculos em diferentes áreas do conhecimento. Dentre essas, está a área da saúde que despontou como ciência por volta do século XVII e utiliza modelos estatísticos como instrumentos para análise de fenômenos biológicos (2). Outro momento que indica o avanço na área na bioestatística, foi a revolução tecnológica que foi iniciada por meio da criação dos computadores e supercomputadores, bem como de *softwares* para análise de grandes bancos de dados (3). Contudo, a sua difusão se deu principalmente pelo movimento intitulado de Medicina Baseada em Evidências (MBE) acrescentou novos argumentos sobre o ensino e a prática da especialidade em análises estatísticas (4). Esse movimento nasceu da associação dos métodos da epidemiologia com a pesquisa clínica e coroou conceitos fundamentais da Medicina e da Saúde com vistas à eficácia, efetividade, eficiência e segurança para a tomada de decisões (5).

Estimulada pelos desafios estatísticos específicos às ciências da saúde, surgiu uma nova disciplina, a bioestatística, que estabeleceu um referencial teórico próprio e criou uma grande variedade de conceitos, métodos e técnicas de análise (3). Por convenção definiu-se “Bioestatística” como a utilização de um conjunto de métodos estatísticos para o planejamento e análise de dados na área da saúde. Ela possui várias utilidades, como a avaliação de testes laboratoriais, definição de faixas de referências, investigação da eficiência de tratamentos, fatores de risco e ocorrência de desfechos (6–8).

O raciocínio estatístico é um elemento crítico de muitas pesquisas empíricas na área da saúde. Estudos incorporam dados bioestatísticos em vários aspectos, desde o desenho do estudo até a análise de dados a geração de relatórios (9). É valoroso que pesquisadores e estudantes de ciências da saúde sejam educados em bioestatística, pois esta é frequentemente empregada em pesquisas com fins de resumo, análise e relato de dados (10). A bioestatística opera em todos os campos das ciências da saúde e assegura que os resultados sejam apresentados de forma adequada em artigos e publicações, por meio de métodos estatísticos e análises de hipóteses adequadas aos dados (4).

Com o advento da revolução tecnológica e a chegada dos computadores pessoais e o surgimento de *softwares* computacionais, o emprego da estatística se tornou mais acessível, pois os cálculos que outrora eram manuais passaram a ser realizados com a utilização de *softwares*, aumentando assim o emprego dos métodos estatísticos e sua utilização (11–13). O desenvolvimento tecnológico, em particular dos computadores pessoais, tem disponibilizado aos pesquisadores um rápido acesso a recursos de resolução de problemas estatísticos, propiciando a difusão e criação de *softwares* estatísticos. (14).

Os *softwares* desempenham um papel de destaque em inúmeras áreas da tecnologia, no mundo dos negócios e nas ciências da Saúde, isso devido a agilidade e confiabilidade aumentos no processamento de grandes bancos de dados em um curto prazo, aliado a robustez dos programas criados. Devido à crescente demanda por programas de computador, houve um constante desenvolvimento dos sistemas de informação, e, com isso, o surgimento de diferentes métodos e modelos de desenvolvimento de *software* (15).

Apesar de existirem *softwares* que realizam análises estatísticas, suas interfaces nem sempre são simples e intuitivas. A maioria dos *softwares* foi desenvolvida focada inteiramente nas especificações técnicas e processuais. Em detrimento disso, os programas perdem aceitabilidade e muitas vezes são usadas de maneira inadequada (16).

Mesmo com o uso e aprimoramento da informática, erros estatísticos nos trabalhos científicos são comumente relatados. Um crescente corpo de literatura aponta para erros persistentes, falhas e deficiências presentes em grande parte das publicações (17,18). A estatística é parte integrante de alguns tipos de pesquisa e erros estatísticos são comuns, o que pode legitimar falsamente os dados (19). A literatura relata o uso indevido de conceitos, métodos e testes estatísticos. Fatores esses que afetam pesquisas de forma significativa (20).

Vários erros na apresentação dos resultados de trabalhos científicos são ligados a fatores como: qualidade dos dados, escolha apropriada e implementação de métodos de análise estatística, avaliação rigorosa e interpretações dos resultados (21,22). Dentre os principais problemas encontrados são relatados: o tamanho da amostra, distribuição dos dados, medição de sumarização incorreta, métodos de teste estatístico equivocados,

especialmente para medidas repetidas, desconsideração à suposição para teste t ou teste ANOVA, falta de ajuste para comparação múltipla (20).

Frequentemente são encontradas avaliações estatísticas realizadas de forma incorreta para um determinado delineamento de pesquisa e/ou tipo de dados. Testes paramétricos básicos são usados com frequência, embora os dados não estejam normalmente distribuídos (23,24).

Costumeiramente os ensaios randomizados de bioestatística empregam testes t para amostras pareadas quando o mesmo paciente foi avaliado antes e depois da intervenção, ou no caso de um estudo experimental versus estudo controle que aplicam testes t para amostras independentes. Esses testes têm como pressupostos a distribuição normal da amostra, mas os dados biomédicos nem sempre seguem essa distribuição, gerando a necessidade de testes não paramétricos (25).

O uso indevido de testes paramétricos quando o correto seria a utilização de seus equivalentes não paramétricos possivelmente origina resultados equivocados, e isso pode ocorrer por descuido ou desconhecimento por parte do pesquisador (23). Em literaturas revisadas por pares constatou-se que cerca de metade das pesquisas clínicas contém um ou mais erros estatísticos, alguns dos quais interferiram nos resultados e na interpretação dos resultados (26), que por conseguinte pode levar a inferências errôneas, bem como prejuízo a saúde do seus beneficiários do conhecimento gerado.

A estatística é uma ferramenta de auxílio aos pesquisadores, e o seu uso de forma adequada requer conhecimento de conceitos teóricos e aplicados (1). A falta desse conhecimento traz problemas, tanto na coleta de dados, na utilização dos cálculos quanto na interpretação dos resultados. Comumente existem falhas na interpretação de indicadores estatísticos, e esse tem sido um dos problemas relatados na literatura das ciências médicas nos últimos anos (27). Isso demonstra o quanto é importante uma abordagem pedagógica para a utilização racional da estatística, e o quanto um *software* prático e educativo é importante para a minimização de erros por parte do usuário, bem como, consolidar os conhecimentos acerca do método estatístico.

Essa quantidade de erros estatísticos fez com que alguns periódicos se atentassem ao ensino e aprendizado da estatística. A estatística é tão importante para a área de saúde que o Conselho Editorial da JSE (*Journal of Statistics Education*) criou uma seção da revista voltada para a formação estatística em ciências da saúde, com o objetivo de “divulgar conhecimento para a melhoria da educação estatística” (28). Promover o pensamento estatístico e relatar sobre a importância da estatística no processo científico tornou-se algo corriqueiro nas ciências (4).

O “distanciamento” entre estudantes da área da área da saúde ao entendimento dos métodos estatísticos se dá em detrimento da linguagem complexa usada pelos estatísticos, bem como *softwares* pouco intuitivos. Tais fatores poderiam justificar certo desinteresse de uma parte considerável dos usuários (29). Mediante a essa lacuna e necessidade é imperativa a criação de programas mais intuitivos. Softwares podem também ser ambientes de aprendizado e consolidação dos conhecimentos acerca da bioestatística. Tal perspectiva promove uma maior qualidade dos estudos e conhecimentos gerados e a minimização de erros.

Tendo em vista a lacuna apresentada, a formulação de um programa de análises, ensino e aprendizagem deve atender à necessidade básica de seus usuários, ou seja, deve conter análises frequentemente utilizadas no contexto da saúde. Nesse sentido, em um estudo com artigos publicados entre os anos de 1995 e 2009 em revistas de odontologia, foram levantados a utilização de /testes estatísticos em 307 trabalhos. Pôde-se constatar que a estatística descritiva representava 40% do total dos testes utilizados, seguido pelos testes de hipóteses para média (paramétricos) e seus respectivos testes não paramétricos (38%). Se somados os dois representam 78% dos métodos estatísticos utilizados nos artigos analisados (26).

Mediante a importância do exposto, o presente trabalho objetivou desenvolver um *software* que realiza estatística descritiva e testes de hipóteses para a média e seus respectivos testes não paramétricos. Além disso, foi idealizada a criação de um *software* para cálculo amostral faz-se necessária a seleção de amostra adequada e representativa por meio de um “n” amostral que permita fazer inferências na população amostrada. Para a criação dos *softwares*, os princípios norteadores foram, deste modo, a simplicidade, a intuição, acessibilidade, praticidade e didática.

2 OBJETIVOS

2.1 Objetivo geral

Desenvolver *softwares* intuitivos voltados para análises estatísticas aplicados em ciências da saúde.

2.2 Objetivos específicos

- Desenvolver um *software* estatístico para a solução de problemas que envolvem testes de hipótese de comparação de médias (testes paramétricos) e seus respectivos testes não paramétricos, favorecendo o aprendizado e a interpretação dos resultados;
- Desenvolver um *software* para estimar o tamanho da amostra mínima necessária para estudos que envolvam médias e proporção;

3 PRODUTOS TÉCNICOS CIENTÍFICOS

3.1 Produto 1 - *Software*: PPGCS *Sample Size*

O PPGCS *Sample Size* é um *software* desenvolvido no formato de calculadora, para calcular o tamanho amostral mínimo necessário para estimar parâmetros populacionais ou testar hipóteses sobre médias e proporções. Esse *software* integrará a página do PPGCS/Unimontes e poderá ser acessado via computador ou celular.

Esse *software* foi registrado no INPI (Instituto Nacional da Propriedade Industrial) sob o processo: BR512022001979-5 (Anexo A).

3.2 Produto 2 - *Software*: Oráculo Estatístico.

O Oráculo Estatístico é *software* desenvolvido para realizar testes de hipótese por meio da comparação de médias (testes paramétricos) e seus respectivos testes não paramétricos. Os testes são calculados de maneira rápida analisando todos os pressupostos necessários para cada teste, e o resultado é demonstrado através de tabelas e com uma explicação que ajuda o usuário a entender os passos realizados, assim como, na interpretação das resoluções. Ele oferece também estatística descritiva dos grupos analisados e pode ser copiado para ferramenta de textos, como o Microsoft Word.

Esse *software* foi registrado no INPI (Instituto Nacional da Propriedade Industrial) sob o processo: BR512022001978-7 (Anexo B).

3.3 Produto 3 - Oráculo estatístico: uma nova ferramenta de análise

Trata-se de um artigo que avalia o desempenho do Oráculo estatístico frente ao PSPP, um *software* estatístico gratuito concorrente do PASW (antigo SPSS). Foi realizado um teste com 11 pessoas, que usaram os dois softwares para análises estatísticas, e os resultados foram apresentados neste artigo.

3.4 Produto 4 – Bibliometrics of CNS & Neurological Disorders – Drug Targets: An International Evolution Along Time

Este artigo foi desenvolvido na disciplina Special Topics in Neuroscience & Health, trata-se de uma pesquisa bibliométrica e foi publicada na revista CNS & Neurological Disorders - Drug Targets, Volume 18, Issue 3, 2019, page: [239 - 244], **DOI:** [10.2174/1871527318666181227123924](https://doi.org/10.2174/1871527318666181227123924).

3.1 Produto 1 – Calculadora: PPGCS SAMPLE SIZE

INTRODUÇÃO

As pesquisas clínicas abrangem um universo grande de elementos que são impossíveis serem analisados em sua totalidade. Por esse motivo, se trabalha com amostras, ou seja, uma parte representativa dessa totalidade de elementos (30). Para isso, é fundamental na elaboração de um projeto de pesquisa uma especificação amostral apropriada e adequada, ao mesmo tempo que representativa da população estudada para que as inferências observadas não sejam comprometidas (31,32). Um estudo com uma amostra que não foi calculada utilizando-se de procedimentos estatísticos adequados não pode ter seus resultados extrapolados, ficando restrita apenas a determinado grupo.

A maioria dos testes estatísticos pressupõe que a amostra estudada seja probabilisticamente representativa da população (31). Para se fazer alguma inferência estatística e generalizar os resultados da pesquisa, um dos preceitos básicos é o cálculo do tamanho amostral mínimo. Assim, cabe ao pesquisador definir, além do tipo de pesquisa, a precisão desejada para seu estudo.

Nesse contexto surgiu a necessidade de se desenvolver um *software* que permite aos pesquisadores estimarem a amostra mínima necessária para determinados tipos de estudos. O *PPGCS Sample Size* é um *software* calculadora desenvolvido para auxiliar o pesquisador a mensurar a amostra mínima necessária para sua pesquisa, e foi projetada para ajudar o pesquisador em estudos onde se deseja comparar médias ou proporções. Além do cálculo amostral, o *software* auxilia na compreensão do delineamento para se amostrar um população desejada, isso por meio de interface didática e intuitiva com explicação dos elementos envolvidos para se estimar um “n” amostral adequado.

Objetivo

O Objetivo do *PPGCS Sample Size* é estimar a amostra mínima necessária para estudos que envolvam médias e proporção.

Público-alvo

Destina-se a todos os pesquisadores que desejem calcular a amostra mínima necessária para pesquisas que envolvam comparação de média ou proporção.

METODOLOGIA

Nessa secção é apresentada a metodologia de estudo, em que constam as descrições das fases de engenharia de software. Essa secção é dividida nos seguintes tópicos: abordagem, identificação de requisitos, linguagem de programação, aperfeiçoamento e registro, testes e instalação e instrução de uso.

Abordagem

Para o desenvolvimento do PPGCS *Sample Size* utilizou-se da prototipagem. Prototipagem pode ser entendida como uma técnica que consiste em desenvolver um sistema experimental (protótipo) para avaliar seus requisitos e funcionalidades (33). Protótipo é um modelo operacional do *software* que ao ser desenvolvido, representa uma versão simplificada, com certos aspectos e funcionalidades. Quanto mais se aproxima de um *software* real mais fiel será o protótipo (34,35).

A prototipação deve começar pela inserção dos requisitos mais bem definidos, e, conforme o processo de construção ocorre, são implementados novos requisitos. Foi utilizada a prototipação evolucionária nesse projeto, pois, trata-se de um modelo de prototipação que permite melhor adaptação às mudanças de requisitos (36).

No primeiro momento foi desenvolvido um protótipo (Figura 1) utilizando o Microsoft Excel (37). Nesse protótipo foram inseridos os cálculos e observada a funcionalidade. O protótipo foi desenvolvido durante a disciplina de Bioestatística I do PPGCS, e usado no seu decorrer para cumprir com os exercícios em sala propostos. O desenvolvimento concomitante à disciplina pode proporcionar ganhos, pois o desenvolvimento e testagem ocorreram frente às explicações e exercícios propostos e permitiram validar os dados gerados.

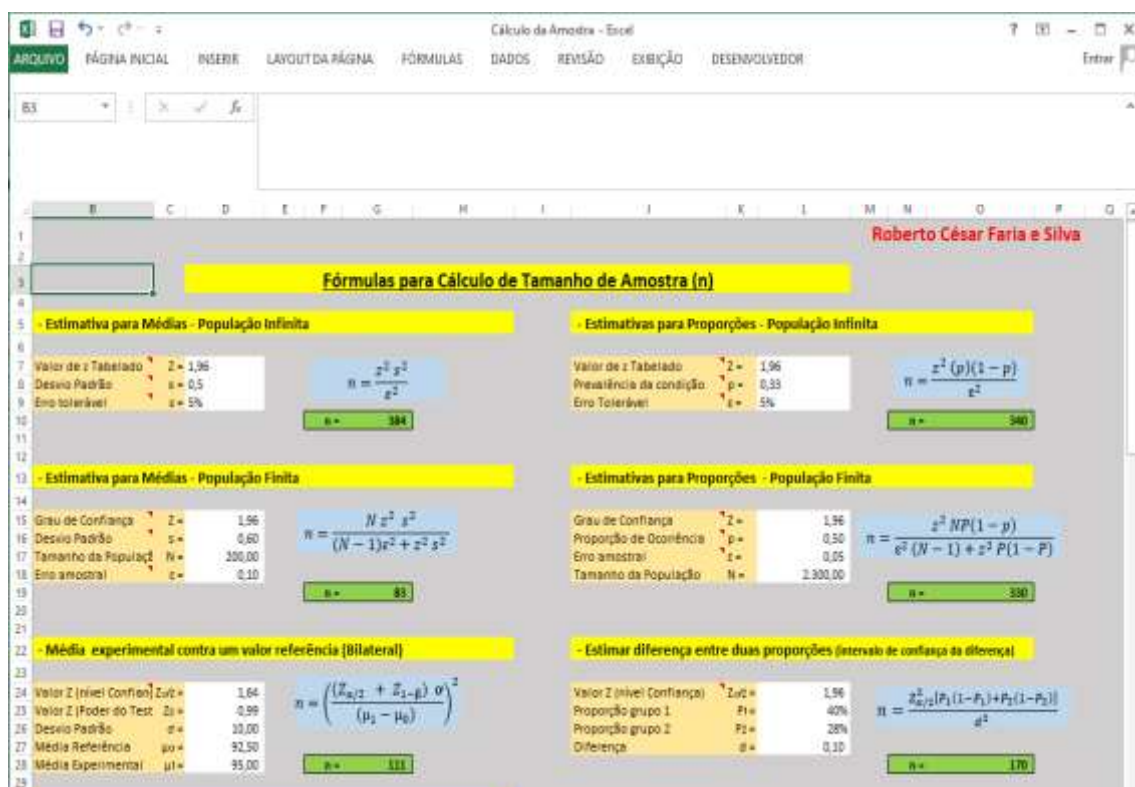


Figura 1 - Protótipo.

Fonte: O próprio autor.

Com base nesse protótipo foi criado o PPGCS *Sample Size*, com o intuito de integrar a página do Programa de Pós-graduação em Ciências da Saúde (PPGCS). A ferramenta foi disponibilizada para pesquisadores realizarem cálculos amostrais.

Identificação de requisitos

Com o objetivo de avaliar e aperfeiçoar o *software*, itens fundamentais para seu melhoramento foram identificados e implementados gradualmente. Periodicamente uma nova versão do programa era disponibilizada para testes pelos pesquisadores. Nas reuniões era reavaliada a forma como os requisitos foram implementados ou mesmo outros novos que foram identificados, o que é previsível em projetos que adotam abordagem evolucionária. Aspectos vinculados à interação, como disposição de elementos na interface gráfica, erros e mensagens de erro foram também analisados, cabendo aos desenvolvedores decidirem quais requisitos eram prioritários e quais seriam incluídos ou modificados na próxima iteração.

Para a gestão dos requisitos, utilizou-se a ferramenta online Trello (Figura 2). Nela, foram enumerados os requisitos não implementados (backlog) e foi realizado o

acompanhamento dos requisitos em implementação em cada iteração e o registro também dos requisitos já implementados. Dessa forma, a atualização constante dos *softwares* foi realizada, até que a versão final de ambos foi alcançada.

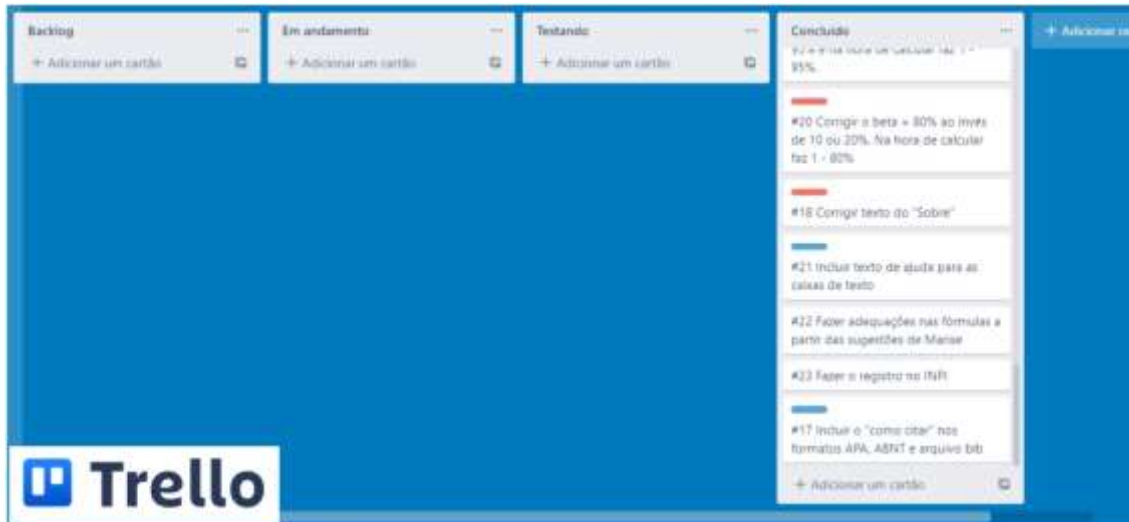


Figura 2 - Trello.

Fonte: O próprio autor.

Linguagem de Programação

No desenvolvimento deste *software* foi utilizada a linguagem PHP (*Hypertext Preprocessor*) versão 7.0, que é uma linguagem de programação fácil e robusta voltada para ambiente Web. Sua camada visual utiliza o HTML 5 (*Hypertext Markup Language*). Essa é uma linguagem comumente usada para a publicação de conteúdo de textos e imagens na *web*. O Java Script é uma linguagem que permite que os conteúdos sejam atualizados instantaneamente e o CSS 3 (*Cascading Style Sheets*) fornece estilos de formatação visual (Figura 3).

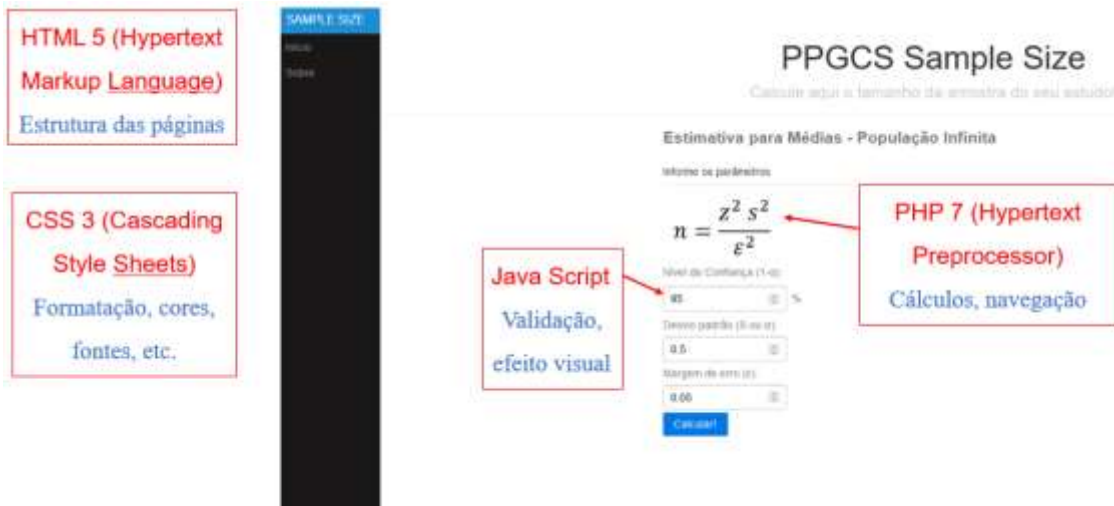


Figura 3 – Linguagens utilizada na programação.
 Fonte: O próprio autor.

A Figura 4 demonstra uma parte do código fonte utilizada na programação.

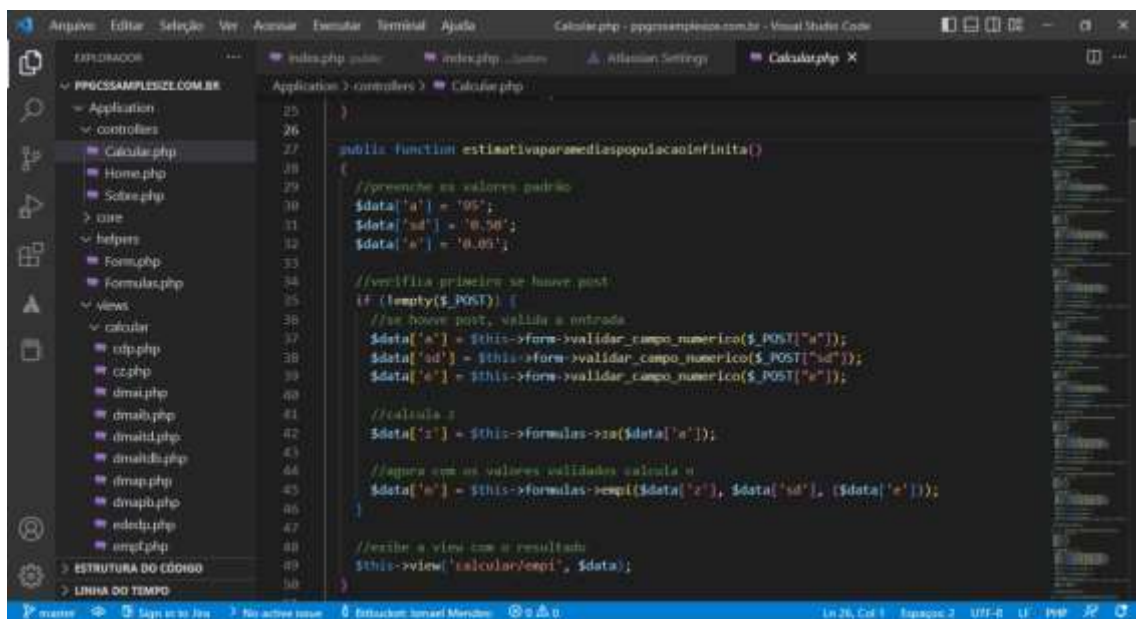


Figura 4 – Parte do código fonte.
 Fonte: O próprio autor.

Aperfeiçoamento e registro

A gestão da configuração e aperfeiçoamento e registro do *software* se deu pelo uso do sistema de versionamento de código Git. Um repositório online, replicado localmente no

computador utilizado para o desenvolvimento. Ela mantém todas as versões do código desde o início do projeto (Figura 5).

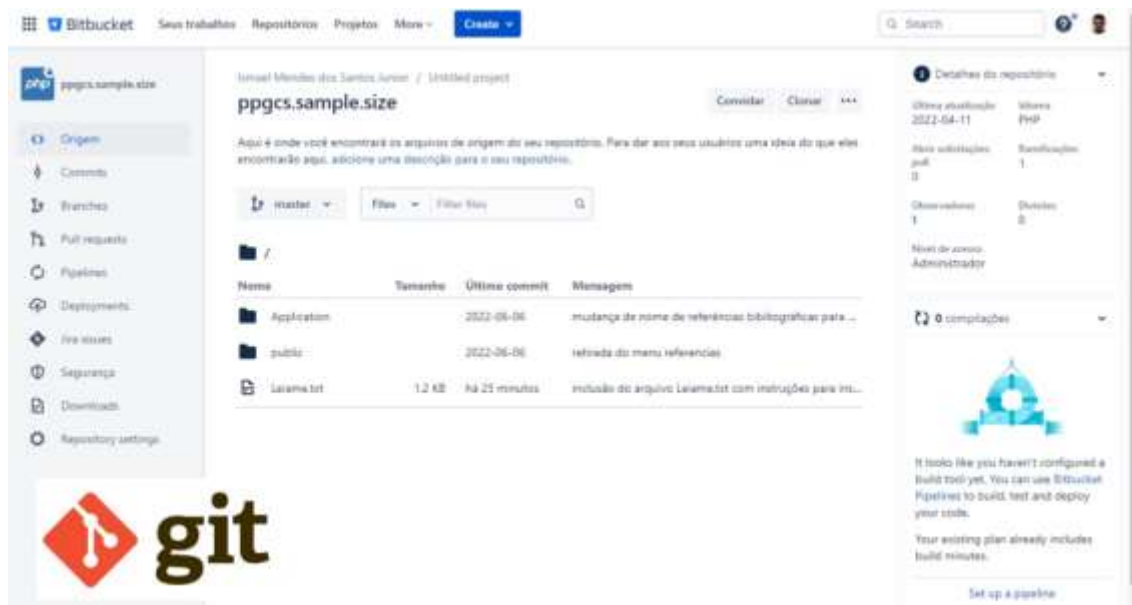


Figura 5 – Git.

Fonte: O próprio autor.

Testes

Todos os requisitos implantados foram testados através de comparações, seja com o protótipo que já tinha sido aferido durante a disciplina de bioestatística, e com cálculos manuais. Estes testes são importantes para validar os resultados encontrados no PPGCS *Sample Size*. Todos os registros foram feitos no Trello que serviram de check list para a checagem de implantação e teste de requisitos.

O Quadro 1 ilustra os cálculos amostrais inseridos no Trello e check list da implantação individual. No Trello o acompanhamento é mais completo pois para cada cálculo amostral constam os campos e as informações a serem preenchidas.

Quadro 1 – Cálculos amostrais que o PPGCS *Sample Size* realiza.

Cálculos amostrais que o PPGCS *Sample Size* realiza.

Estimativa para Médias - População Infinita
Estimativa para Médias - População Finita
Média experimental contra um valor referência - Teste Unilateral
Média experimental contra um valor referência - Teste Bilateral
Duas Médias amostrais pareadas - Teste Unilateral
Duas Médias amostrais pareadas - Teste Bilateral
Duas Médias amostrais independentes de mesmo tamanho - Teste Unilateral
Duas Médias amostrais independentes de mesmo tamanho - Teste Bilateral
Duas Médias amostrais independentes de tamanho diferente - Teste Unilateral
Duas Médias amostrais independentes de tamanho diferente - Teste Bilateral
Estimativas para Proporções - População Infinita
Estimativas para Proporções - População Finita
Estimar diferença entre duas proporções - intervalo de confiança da diferença
Comparar duas proporções - uma experimental com uma referência

Fonte: O próprio autor.

Instalação e instruções de uso

Por se, tratar de um programa em ambiente Web ele não precisa ser instalado. Para isso é necessário uma conexão com a internet. Muitos softwares necessitam de manual de instrução. No PPGCS Sample Size as instruções de uso são localizadas dentro do próprio programa. Na tela inicial são apresentados os cálculos amostrais disponíveis e uma breve explicação de quando se usa o cálculo (Figura 6).

[Estimativa para Médias - População Infinita](#)

Utiliza-se quando o pesquisador deseja estimar o intervalo de confiança para média populacional, considerando uma população infinita (número de elementos muito elevado. Ex: População de São Paulo, de um País).

Figura 6 – Explicação de uso.

Fonte: O próprio autor.

Após a escolha do cálculo amostral desejado, o software orienta sobre a ação a ser realizada. Todos os locais onde os dados devem ser inseridos contém uma explicação do uso por meio de uma dica de tela (tooltip) isso auxilia o usuário a preencher o valor na caixa de texto (Figura 7).

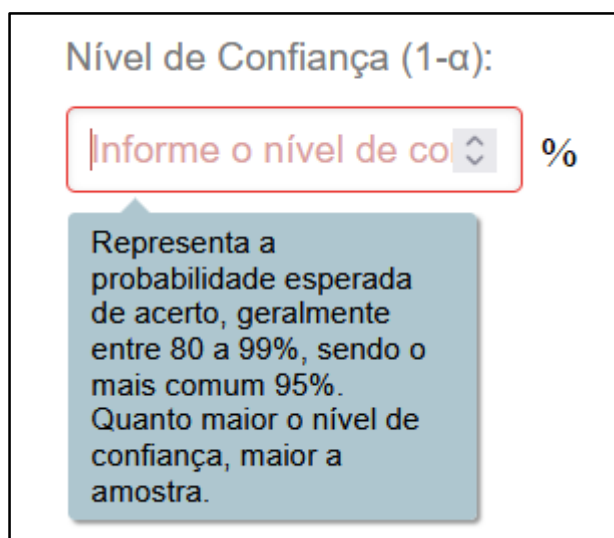


Figura 7 – Dica de tela.

Fonte: O próprio autor.

Uma nova tendência de mercado são os manuais de instrução no formato de vídeo, pois esses apresentam várias vantagens:

- Divulgação do produto na maior vitrine do mundo: a internet;
- Menor custo para produção;
- Maior facilidade de entendimento por parte do usuário.

Devido às vantagens apresentadas, optou-se por fazer o manual de instrução do PPGCS Sample Size no formato de vídeo, e o mesmo se encontra disponível no Youtube através do endereço: <https://youtu.be/vv22zv11Zqg>



Figura 8 – Vídeo com instruções de uso do PPGCS Sample Size.

Fonte: O próprio autor.

RESULTADOS E DISCUSSÃO

Apresentação

A ideia foi criar um *software* gratuito, no formato de calculadora, de fácil acesso e em ambiente web, que pudesse ser acessado de qualquer lugar, inclusive pelo celular. Esse *software* pode ser acessado pela página do programa, <http://www.ppgcs.unimontes.br/>, junto com os produtos produzidos pelo PPGCS. Ao clicar no produto PPGCS Sample Size o usuário será redirecionado para a página: http://www.ppgcs.unimontes.br/ppgcs/index.php/pt-br/?option=com_sppagebuilder&view=page&id=194 - clicar no botão calcular ou para começar a usar a ferramenta. O PPGCS *Sample Size* está registrado no INPI (Instituto Nacional da Propriedade Industrial) sob o processo: BR512022001979-5 (Anexo B).

Desenvolvimento

Primeiramente foi desenvolvida uma planilha de no Microsoft Excel (37) que serviu como protótipo para a criação do PPGCS *Sample Size* (Figura 9), nessa planilha estavam contemplados vários cálculos amostrais para comparar e estimar médias e proporções.

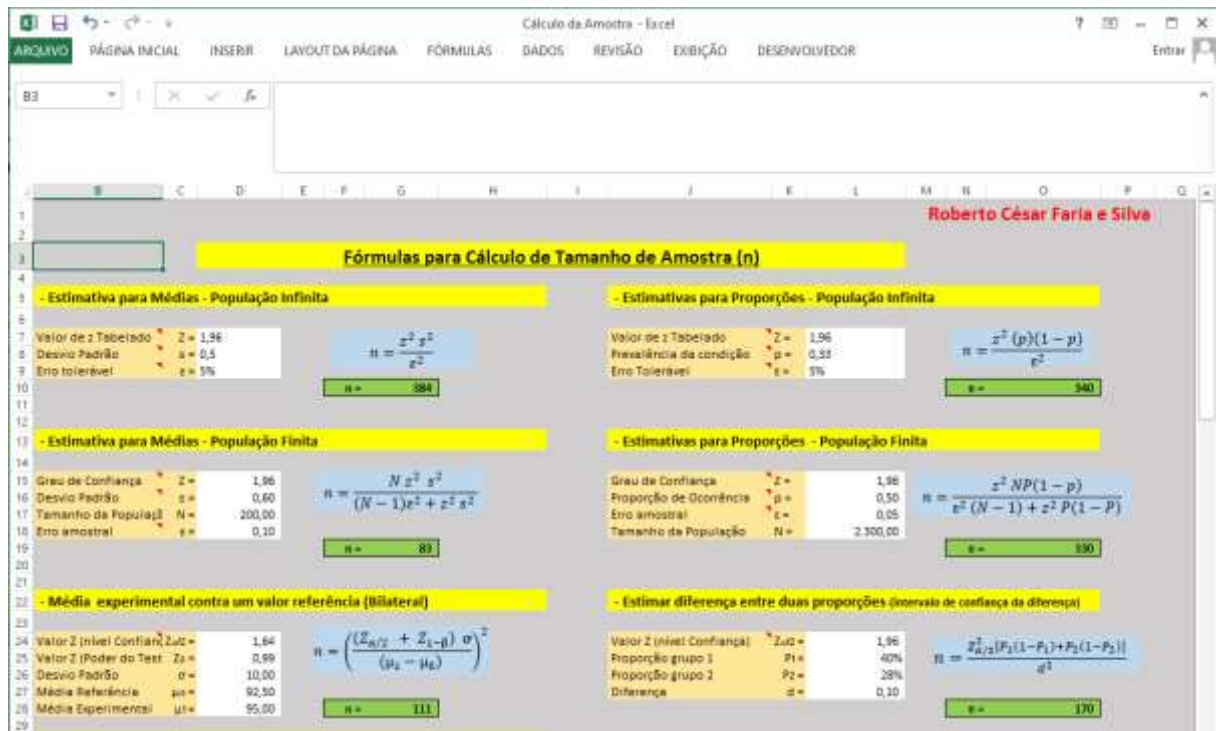


Figura 9 – Protótipo desenvolvido no Excel.
 Fonte: O próprio autor.

As telas do PPGCS *Sample Size* apresentam na aba lateral três menus: Início, Referências e Sobre. No menu Início consta a tela inicial do PPGCS *Sample Size* (Figuras 10, 11 e 12). Essa tela apresenta um leque de opções de cálculo amostral distribuído em dois grupos: a) Para estimar e comparar médias e b) para estimar e comparar proporções. Cada opção de cálculo possui um breve comentário sobre quando se deve utilizá-lo.



Figura 10 – PPGCS Sample Size – Tela Inicial.
Fonte: O próprio autor.

SAMPLE SIZE

Início
Sobre

[Duas Médias amostrais pareadas - Teste Unilateral](#)

[Duas Médias amostrais pareadas - Teste Bilateral](#)

Utiliza-se quando deseja-se comparar médias de um mesmo grupo avaliado em momentos distintos, ou por avaliadores/métodos distintos.

[Duas Médias amostrais independentes de mesmo tamanho - Teste Unilateral](#)

[Duas Médias amostrais independentes de mesmo tamanho - Teste Bilateral](#)

[Duas Médias amostrais independentes de tamanho diferente - Teste Unilateral](#)

[Duas Médias amostrais independentes de tamanho diferente - Teste Bilateral](#)

Utiliza-se quando deseja-se comparar médias de dois grupos distintos.

Para estimar e comparar proporções

Utilizado para estudos comparativos em ensaios clínicos, estudos observacionais do tipo coorte e caso controle, onde se pretende estudar a associação entre duas variáveis dicotômicas.

[Estimativas para Proporções - População Infinita](#)

Utiliza-se quando o pesquisador deseja estimar o intervalo de confiança para proporção populacional, considerando uma população infinita (número de elementos muito elevado. Ex: População de São Paulo, de um País).

Figura 11 – PPGCS Sample Size – Tela Inicial (Continuação I).
Fonte: O próprio autor.

SAMPLE SIZE

Início
Sobre

Para estimar e comparar proporções

Utilizado para estudos comparativos em ensaios clínicos, estudos observacionais do tipo coorte e caso controle, onde se pretende estudar a associação entre duas variáveis dicotômicas.

[Estimativas para Proporções - População Infinita](#)

Utiliza-se quando o pesquisador deseja estimar o intervalo de confiança para proporção populacional, considerando uma população infinita (número de elementos muito elevado. Ex: População de São Paulo, de um País).

[Estimativas para Proporções - População Finita](#)

Utiliza-se quando o pesquisador deseja estimar o intervalo de confiança para proporção populacional, considerando uma população finita (número de elementos limitado. Ex: Número de estudantes de uma escola, pacientes de um hospital).

[Estimar diferença entre duas proporções - intervalo de confiança da diferença](#)

Utiliza-se quando deseja-se estimar intervalo de confiança para a diferença entre as proporções de dois grupos distintos.

[Comparar duas proporções - uma experimental com uma referência](#)

Utiliza-se quando se deseja comparar a proporção obtida experimentalmente com uma proporção tida como referência (populacional).

Figura 12 – PPGCS Sample Size – Tela Inicial (Continuação II).
Fonte: O próprio autor.

Ao se escolher uma das opções de cálculo amostral, como por exemplo, “Estimativa para média – População Infinita”, será aberta uma outra tela com a fórmula utilizada para o cálculo e os dados a serem preenchidos pelo pesquisador. Após o preenchimento é necessário apertar o botão **calcular** (Figura 13).

The screenshot shows a web application interface for calculating sample size. On the left is a dark sidebar with 'SAMPLE SIZE' at the top and 'Inicio' and 'Sobre' below. The main content area has the title 'PPGCS Sample Size' and the subtitle 'Calcule aqui o tamanho da amostra do seu estudo!'. Below this is a section titled 'Estimativa para Médias - População Infinita' with the instruction 'Informe os parâmetros'. The formula
$$n = \frac{Z^2 S^2}{\epsilon^2}$$
 is displayed. There are three input fields: 'Nível de Confiança (1-α)' with the value '95', 'Desvio padrão (S ou σ)' with the value '0,5', and 'Margem de erro (ε)' with the value '0,05'. A blue 'Calcular!' button is at the bottom.

Figura 13 – PPGCS Sample Size – Estimativa para Média – População Infinita.

Fonte: O próprio autor.

Ao se clicar em uma das caixas para preencher o valor definido na pesquisa, como por exemplo, “Nível de confiança”, o PPGCS *Sample Size* mostra uma caixa de diálogo com uma pequena explicação sobre o campo preenchido (Figura 14).

SAMPLE SIZE

Início

Sobre

PPGCS Sample Size

Calcule aqui o tamanho da amostra do seu estudo!

Estimativa para Médias - População Infinita

Informe os parâmetros

$$n = \frac{z^2 s^2}{\epsilon^2}$$

Nível de Confiança (1-α):

95%

Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo a mais comum 90%. Quanto maior o nível de confiança, maior a amostra.

Calcular!

Figura 14 – PPGCS *Sample Size* – Caixa com explicação.
Fonte: O próprio autor.

Depois de preenchidos todos os parâmetros, clica-se em calcular, para que o *software* apresente o resultado (Tamanho da amostra (n)) e o valor de z tabelado (Figura 15). Apesar da fórmula do cálculo usar o valor do z tabelado, que é medido através do nível de confiança desejado, acredita-se que é mais simples para o pesquisador informar o nível de confiança do que o valor z. Por esse motivo optou-se por colocar na calculadora o preenchimento por parte do pesquisador - o nível de confiança e não o valor de z.

SAMPLE SIZE

Início

Sobre

PPGCS Sample Size

Calcule aqui o tamanho da amostra do seu estudo!

Estimativa para Médias - População Infinita

Informe os parâmetros

$$n = \frac{z^2 S^2}{\epsilon^2}$$

Nível de Confiança (1-α):

95 %

Valor de z tabelado (z):

1,96

Desvio padrão (S ou σ):

0,5

Margem de erro (ε):

0,05

Calcular!

Resultado:

Tamanho da amostra (n):

385

Figura 15 – PPGCS Sample Size – Cálculo.

Fonte: O próprio autor.

A Figura 16 apresenta a aba “Sobre”, onde é mostrada a equipe de desenvolvedores, com link para seus respectivos Currículos Lattes e ORCID, além de “Como Citar?”.



Figura 16 – PPGCS *Sample Size* – aba Sobre.

Fonte: O próprio autor.

A Figura 17, ainda na aba “Sobre”, apresenta também as referências bibliográficas utilizadas na calculadora.



Figura 17 – PPGCS *Sample Size* – Referências.

Fonte: O próprio autor.

As demais funcionalidades desta calculadora (PPGCS *Sample Size*), seguem o mesmo padrão do exemplo demonstrado.

Testes Realizados

Foram realizados vários testes para medir a acurácia do PPGCS *Sample Size*, aqui neste trabalho iremos usar como exemplo um exercício utilizado em uma aula ministrada no Programa de Pós-graduação em Ciências da Saúde na disciplina de Bioestatística I.

Nela desejava-se conhecer o número médio de dentes permanentes com experiência de cárie aos 12 anos em uma comunidade. Em estudo prévio, os valores encontrados foram: CPOD (dentes obturados, perdidos e cariados) médio 2,7 e desvio padrão 0,5. Quantas crianças devem ser examinadas se o erro tolerável é igual a 0,02 e o nível de confiança definido foi de 95% (38)?

Esse problema foi resolvido manualmente, mas para isso foi necessário encontrar o Z tabelado que para 95% correspondesse a 1,96.

$$n = \frac{z^2 s^2}{\varepsilon^2} \quad n = \frac{1,96^2 \cdot 0,5^2}{0,02^2} \quad n = \frac{3,8416 \times 0,25}{0,004} \quad n = \frac{0,9604}{0,0004} \quad n = 2.401$$

Além do cálculo manual foi realizado também com o auxílio de uma planilha do Excel (Figura 18). Esse mesmo cálculo foi realizado no PPGCS Sample Size, como demonstrado na figura 19.

- Estimativa para Médias - População Infinita	
Valor de z Tabelado	Z = 1,96
Desvio Padrão	s = 0,5
Erro tolerável	ε = 2%

$$n = \frac{z^2 s^2}{\epsilon^2}$$

n = 2.401

Figura 18 – Cálculo amostral no Excel.
Fonte: O próprio autor.

SAMPLE SIZE

Início
Sobre

PPGCS Sample Size

Calcule aqui o tamanho da amostra do seu estudo!

Estimativa para Médias - População Infinita

Informe os parâmetros

$$n = \frac{z^2 s^2}{\epsilon^2}$$

Nível de Confiança (1-α):
95 %

Valor de z tabelado (z):
1,96

Desvio padrão (S ou σ):
0,5

Margem de erro (ε):
0,02

Calcular!

Resultado:

Tamanho da amostra (n):
2401

Figura 19 – Cálculo amostral no PPGCS Sample Size.
Fonte: O próprio autor.

Pode-se perceber que os resultados foram os mesmos, o que mostra a exatidão no cálculo realizado pelo PPGCS *Sample Size*. Esses cálculos foram feitos em todas as

funcionalidades da calculadora e em todos os testes ele apresentou precisão em seus resultados, validando assim a mesma.

Conclusão

O *software* PPGCS *Sample Size* apresenta-se como uma calculadora amostral de qualidade, uma vez que pelos testes de validação demonstrou sua precisão nos cálculos do tamanho amostral, essa validação torna o PPGCS *Sample Size* apto para ser usado, podendo assim contribuir para todos os pesquisadores, por meio de uma plataforma intuitiva e educativa.

3.2 Produto 2 – SOFTWARE: ORÁCULO ESTATÍSTICO

Registrado no INPI (Instituto Nacional da Propriedade Industrial) sob o processo: BR512022001979-5 (Anexo B).

INTRODUÇÃO

Existem vários *softwares* e planilhas estatísticas comercialmente ou gratuitamente disponíveis, dentre os quais podem-se destacar o Microsoft Excel (37), R (39), SPSS (40), PSPP (41), Stata (42) e o Jamovi (43), dentre outros. Os programas possuem diferentes usuárias de acordo as demandas que cada um requer para a área de aplicação. Embora seja ampla a variedade de *softwares* como ferramenta estatística, há ainda, uma grande necessidade de programas que sejam mais intuitivos a públicos em formação na área de análise de dados. Nesse sentido, apresenta-se o Oráculo Estatístico, o qual é apresentado em paralelo aos programas existentes e suas vantagens que o mesmo supri dentre os oferecidos pelo mercado.

O Microsoft Excel é um *software* para criação de planilhas e gráficos desenvolvido pela Microsoft em 1987 e que é atualizado constantemente. Ele possui em seu pacote fórmulas predefinidas que ajudam os usuários a desenvolverem cálculos de distribuição, testes de hipóteses, estatística descritiva, criação de gráficos, dentre outras. (37,44). Outro *software* de destaque é o “R”, um *software* estatístico popular, de código aberto, livre e gratuito. Nele os usuários podem usar os pacotes livremente, desenvolver contribuições e incorporá-las ao programa. Isso pode ser feito a partir da criação de pacotes próprios (39). Dentre o *hall* de pacotes estatísticos mais usados em ciências da saúde, destaca-se o *software Predictive Analytics SoftWare* (PASW), antigo *Statistical Package for Social Science* (SPSS), que permite a utilização de dados em diversos formatos para gerar relatórios, calcular estatísticas descritivas, conduzir análises estatísticas avançadas e elaborar gráficos. Ele apresenta-se amplamente difundido devido apresentar uma gama de análises desde as mais simples envolvendo frequências até modelos equacionais complexos (40). Contudo, requer uma organização prévia dos dados para serem

importados, além de não possuir código aberto para aprimoramento de ações. Por outro lado, com menos recursos, mas de modo acessível, a *Free Software Foundation* (41) desenvolveu o PSPP, que é um *software* livre, similar ao PASW *Statistics*, disponibilizado de forma gratuita e não necessita de licença, sendo esta sua principal vantagem.

O Stata constitui um dos *softwares* que oferece um pacote completo e integrado para a manipulação de dados, cálculos estatísticos e visualização gráfica. Ele busca atender às necessidades dos seus usuários quanto ao tratamento de dados (42). Já o projeto Jamovi foi criado para desenvolver uma plataforma estatística gratuita e aberta, de uso intuitivo e capaz de fornecer um conjunto completo de análises estatísticas (testes t, ANOVAs, correlação e regressão, testes não paramétricos, tabelas de contingência, confiabilidade e análise fatorial). No cerne da filosofia Jamovi, qualquer pessoa pode desenvolver e publicar análises e disponibilizá-las para um público amplo (43).

Apesar de existir uma variedade de *softwares*, tais programas e ferramentas pouco intuitivas e exigem do usuário domínio de estatística prévio mínimo, além de conhecimento específico de operacionalização do *software* ou planilha que pretende utilizar. Sendo assim, a criação de um *software* onde possa ser além de uma ferramenta de análise de dados, seja um ambiente de aprendizado e consolidação de conhecimentos torna-se apreciável.

Diante da lacuna encontrada, foi desenvolvido O Oráculo estatístico como forma de auxiliar o usuário a realizar testes de hipóteses para comparação de médias (testes paramétricos) e seus respectivos testes não paramétricos e contribuir para o aprendizado dessa técnica estatística. Este *software* processa os dados inseridos e gera os resultados na forma de relatório simples, apresentando o processo de análise e a interpretação de cada cálculo realizado, contribuindo para o aprendizado.

Objetivo

O Objetivo do Oráculo estatístico é de realizar os testes estatísticos para a solução de problemas que envolvem testes de hipótese, de médias (testes paramétricos) e seus respectivos testes não paramétricos. Ele também visa o favorecimento do aprendizado estatístico do usuário.

Público-alvo

Pesquisadores que desejam realizar testes estatísticos para comparação de médias (paramétricos) e seus respectivos testes não paramétricos.

METODOLOGIA

Nessa seção é apresentada a metodologia de estudo, em que consta a descrição das fases de engenharia de software. Essa seção é dividida nos seguintes tópicos: abordagem, identificação de requisitos, linguagem de programação, aperfeiçoamento e registro, testes e instalação e instrução de uso.

Abordagem

Para o desenvolvimento do Oráculo Estatístico adotou-se como abordagem para o ciclo de vida o *Rational Unified Process* (RUP). Trata-se de uma perspectiva iterativa e incremental que visualiza seis disciplinas de projeto e quatro fases para o ciclo de vida de um software. As disciplinas são: Modelagem de negócio; Requisitos; Análise e design; Implementação; Teste; Implantação. As quatro fases são a concepção, a elaboração, a construção e a transição (Figura 1). Essas fases são sequenciais, mas as disciplinas são

executadas de forma paralela ou sequencial, dependendo do que se propõe a cada iteração (34,36,45).

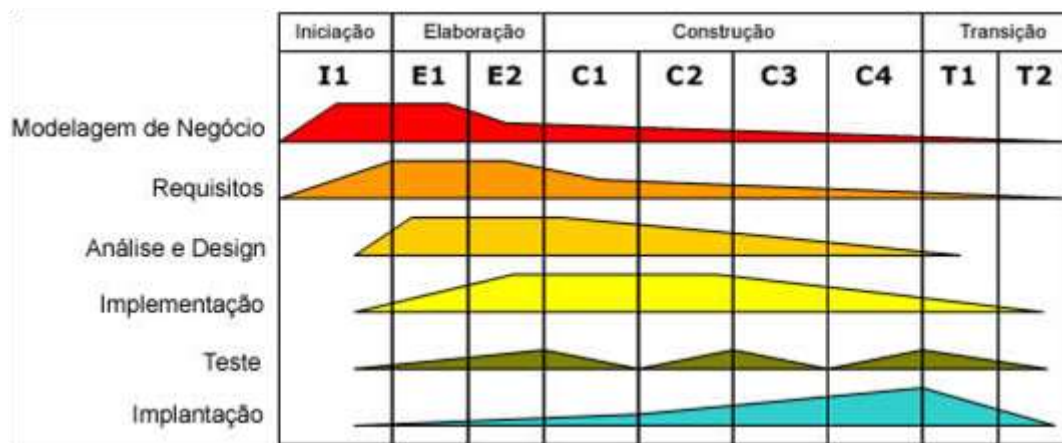


Figura 1 - *Rational Unified Process* (RUP).
 Fonte: Pressman (2021) (34).

Fase de iniciação é a fase planejamento, é onde se delinea o projeto em termos de custos, prazos, levantamento de requisitos, definição do escopo do projeto e é estabelecido o caminho a seguir. Na fase de elaboração da modelagem do software se deve analisar o domínio do problema e verificar os riscos do projeto. Já na fase de construção o maior foco é a implementação do software e na fase de transição o foco é a entrega do software ao usuário (34).

Identificação dos requisitos

Procurou-se compreender as necessidades dos usuários, identificar os requisitos do sistema, documentá-los e decidir as tecnologias utilizadas para o projeto. Os requisitos foram documentados usando-se a notação proposta pela *Unified Modeling Language* (UML). A UML é uma família de notações gráficas que ajudam na descrição e no projeto de sistemas de *software*, permitindo modelar o sistema nas perspectivas contextual, estrutural, interativa e comportamental (46). Foram utilizados os diagramas de casos de uso, de classes e de sequência.

O sistema Oráculo Estatístico foi pensado para um usuário, que acessa em um único ambiente integrado todas as funcionalidades do sistema necessárias para realizar suas

análises. Para identificação dessas funcionalidades, foi criado um diagrama de casos de uso por meio do *software* Astah Community 7.0 (**Erro! Fonte de referência não encontrada.** 2). Esse diagrama tem papel fundamental na organização e modelagem dos componentes de um sistema, ilustrando a visão estática do caso de uso, através dos relacionamentos entre os autores e os casos de uso (BOOCH; RUMBAUGH; JACOBSON, 2005).

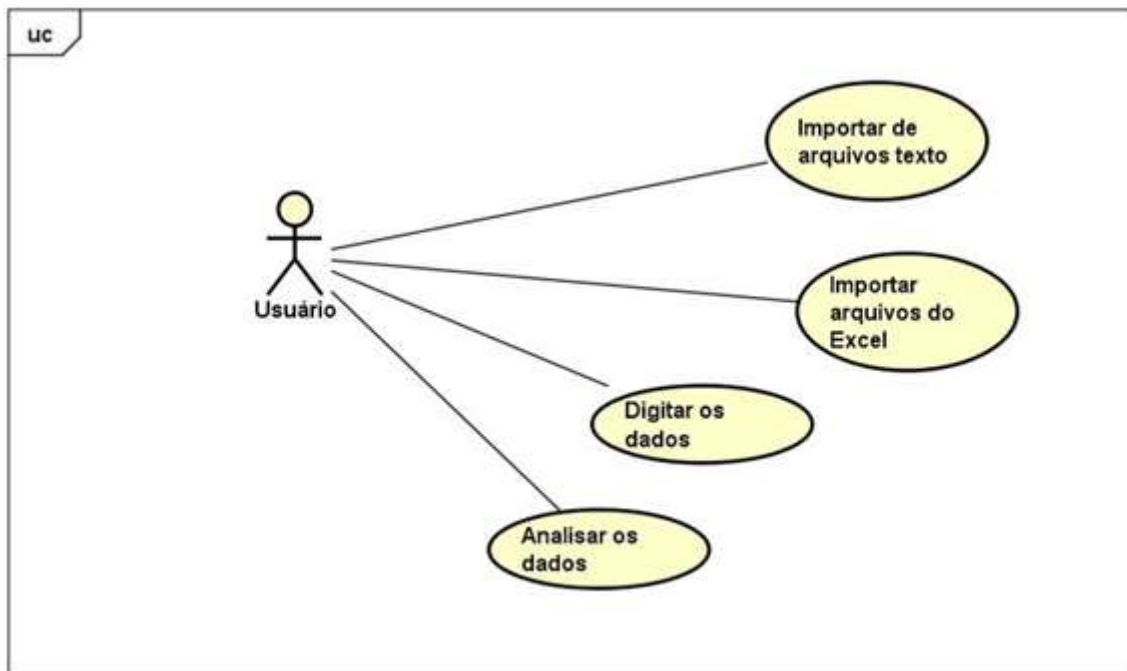


Figura 2 – Diagrama de caso de uso
Fonte: O próprio autor.

A primeira funcionalidade, “Importar de arquivos de texto” consiste em uma interface gráfica para que o usuário faça a importação dos dados previamente digitados que foram criados em arquivos de texto separados por vírgula ou *comma separated values* (CSV). A interface gráfica em formato de assistente (*wizard*) solicita ao usuário o arquivo desejado e o auxilia na importação das informações para o formato tabular (linhas e colunas) utilizado no sistema.

A funcionalidade “Importar arquivos do Excel” provê uma interface gráfica semelhante à funcionalidade previamente descrita para que o usuário importe os dados anteriormente digitados em formato de planilha eletrônica do Microsoft Excel (extensão xls ouxlsx).

Caso não queira importar os dados, o usuário pode digitá-los diretamente, assim como, alterá-los. A terceira funcionalidade - “Analisar os dados” - oferece uma interface gráfica para edição dos dados que permite incluir linhas, colunas, alterar dados de um registro (célula) em particular.

Uma vez que os dados estejam digitados e prontos para a análise, o usuário poderá realizá-la. A funcionalidade “Analisar os dados” é implementada no aplicativo por meio de uma interface gráfica também em forma de assistente (*wizard*) em que o usuário informa que tipo de análise deseja e quais colunas (campos) irá utilizar na análise. A partir das características da análise é exibido um relatório em texto contendo parágrafos e tabelas com os resultados dos testes estatísticos escolhidos.

Finalmente, a funcionalidade “Conhecer sobre o aplicativo” oferece ao usuário informações sobre a equipe de desenvolvimento. Os participantes são listados e, além das suas atribuições, são disponibilizados *links* para seus currículos. Para organizar o código de forma a atender aos casos de uso propostos, foi criado um diagrama de classes e relacionamentos utilizando funcionalidade do Visual Studio 2019 para tal (Apêndice A). O sistema possui ao todo 68 classes, das quais 36 classes são utilizadas nos cálculos realizados pelo sistema (caso de uso “Analisar os dados”).

A Tabela 1 reúne em uma listagem uma breve descrição do papel de cada classe. As classes que implementam os testes estatísticos herdam as propriedades e métodos da classe base *StatisticalTest* e implementam o método “Run” (é o método acionado pelas instâncias da classe para realizar o cálculo). Essa classe base tem como principais propriedades:

- *AreEqual* (booleana): indica se os valores são iguais;
- *Stat*: traz a estatística do teste;
- *CriticalValue*: traz a estatística de valor crítico;
- *PValue*: o valor de p-valor;
- *Result*: o texto de análise do resultado do teste

Assim, por exemplo as classes “*ManWhitneyTest*” e “*TwoSampleTTestTest*” assim como todas as demais que implementam os testes (aparecem no diagrama com a indicação da herança por meio da seta) possuem essas propriedades e fazem suas implementações

do método “Run”. Na maioria dos casos o teste estatístico já tinha implementação no framework Accord.NET e é nesse método que é instanciado.

Entretanto, em alguns casos o framework ainda não tinha implementação dos testes necessários ou foi necessário complementá-lo com algum pós-teste. Assim, classes com os cálculos foram incluídas. Elas têm o nome do teste, mas não finalizam com o termo “Test”. É o caso da classe “MannWhitney”. Essa classe implementa o teste no método “MannWhitney” e tem como propriedades o valor crítico (CriticalValue), p-valor (PValue) e estatística de teste (Stat).

Tabela 1 – Descrição das classes do sistema

#	Nome da classe	Descrição
1	StatisticalTest	É a classe base para as classes que implementam os testes estatísticos.
2	ManWhitneyTest	Implementa o Teste de Man-Whitney.
3	MannWhitney	Implementa os cálculos para o teste de Man-Whitney
4	TwoSampleTTestTest	É a classe que implementa o teste Teste T para duas amostras
5	MauchlyTest	Implementa o Teste de Mauchly.
6	Mauchly	Implementa os cálculos para o Teste de Mauchly.
7	WelchTest	Implementa o Teste T de Welch.
8	Welch	Implementa os cálculos para o Teste T de Welch.
9	GreenhouseGeisserEpsilonTest	É a classe que implementa o Teste Greenhouse and Geisser Epsilon (EGG)
10	GreenhouseGeisserEpsilon	Implementa os cálculos do Teste EGG
11	WilcoxonTest	Implementa o Teste de Wilcoxon.
12	OneWayAnovaTest	Implementa o teste ANOVA
13	OneWayAnova	Implementa os cálculos do teste ANOVA
14	FDistribution	É a classe correspondente à distribuição estatística F

15	HuynhFeldtEpsilonTest	Implementa o teste Huynh-Feldt Epsilon (HFE)
16	HuynhFeldtEpsilon	Implementa o cálculo do teste HFE
17	DMS	Implementa o teste DMS para os grupos
18	RepeatedMeasuresAnovaTest	Implementa o teste ANOVA (medidas repetitivas) ou Repeated Measures Analysis of Variance (ANOVA).
19	RepeatedMeasuresAnova	Implementa os cálculos do teste ANOVA (medidas repetitivas).
20	PairedTTest	Implementa o teste T pareado.
21	BonferroniTest	Implementa o teste de Bonferroni.
22	Bonferroni	Implementa os cálculos do teste de Bonferroni.
23	KolmogorovSmirnovTest	Implementa o teste de normalidade de de Kolmogorov-Smirnov.
24	KruskalWallisTest	Implementa os cálculos do teste de Kruskal-Wallis.
25	KruskalWallis	Implementa o teste de Kruskal-Wallis.
26	ShapiroTest	Implementa o teste de Shapiro.
27	LeveneTest	Implementa o teste de Levene.
28	FriedmanTest	Implementa o teste de Friedman.
29	Friedman	Implementa os cálculos do teste de Friedman.
30	TTest	Implementa o Teste T.
31	DescriptiveTable	É a classe que recebe o resultado de um teste (do tipo StatisticalTest) e o exibe em formato de tabela para apresentação no relatório final.

Fonte: O próprio autor.

Durante o desenvolvimento foram criados casos de teste para validar os resultados das classes que foram implementadas (testes de unidade). A classe “StatisticsTestsTest” tem um método para cada caso de teste contendo a implementação desses testes. Por exemplo, o trecho de código do método “MannWhitneyTest” implementa um teste tendo como parâmetro dois grupos (A e D) com $n = 10$. (linhas 06 a 21). Depois que o teste de Mann Whitney é executado (linhas 23 e 24) as propriedades p-valor (linha 27) e estatística de teste (linha 28) são conferidas com os valores esperados (Figura 3).

```

01 public void MannWhitneyTest()
02 {
03     DataTable dt = new DataTable();
04     dt.Clear();
05
06     dt.Columns.Add("Grupo A", typeof(double));
07     dt.Columns.Add("Grupo D", typeof(double));
08
09     dt.Rows.Add(47, 48);
10     dt.Rows.Add(48, 48);
11     dt.Rows.Add(48, 48);
12     dt.Rows.Add(49, 48);
13     dt.Rows.Add(49, 48);
14     dt.Rows.Add(49, 48);
15     dt.Rows.Add(50, 48);
16     dt.Rows.Add(50, 51);
17     dt.Rows.Add(51, 51);
18
19     List<string> var = new List<string>();
20     var.Add("Grupo A");
21     var.Add("Grupo D");
22
23     double[][] dr = dt.ToBidimensionalDoubleArray(var);
24     MannWhitney m = new MannWhitney(dr, var);
25
26
27     Assert.AreEqual(0.368, Math.Round(m.PValue, 3));
28     Assert.AreEqual(31, Math.Round(m.Stat, 0));
29     ...

```

Figura 1 – Teste de unidade para o Teste de Mann Whitney
Fonte: O próprio autor.

As demais classes do sistema são utilizadas para implementar os demais casos de uso e operacionalizar a interface gráfica. Na linguagem C# as telas são também classes: “MainForm” é a classe para a tela principal e “EditForm” é a classe para as telas de edição. Os componentes inseridos nas telas também são classes e há também classes de apoio utilizadas para viabilizar o registro e a validação dos dados, bem como a interação com o usuário.

As anotações dos requisitos bem como o acompanhamento da programação foram registradas no Trello (Figura 4) que serviu também como check list do processo. A escolha do Trello como ferramenta se baseou no fato de se tratar de uma ferramenta visual que possibilita várias pessoas a gerenciar os trabalhos.

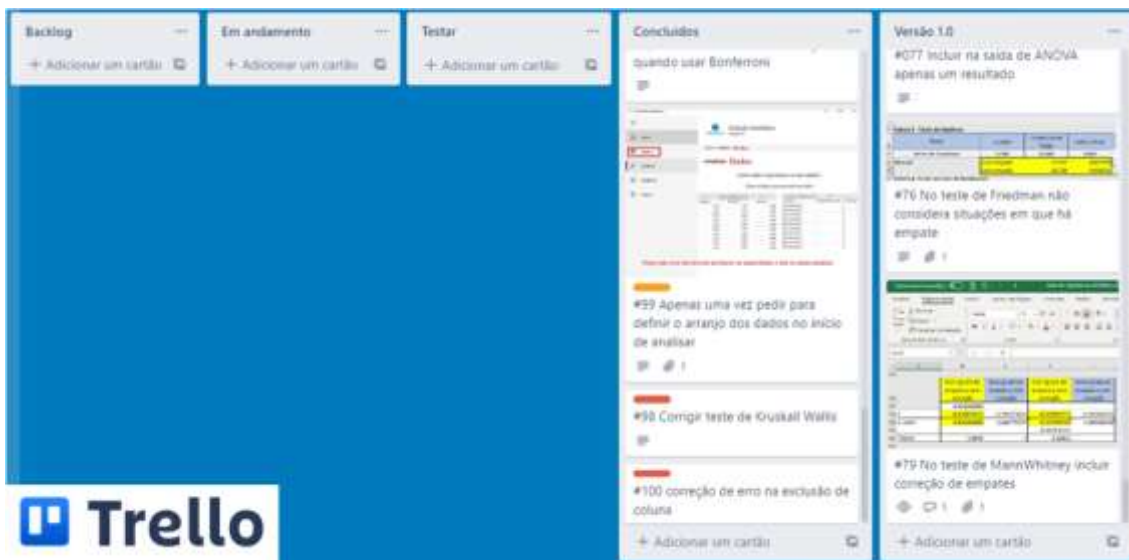


Figura 4 – Trello
Fonte: O próprio autor.

Linguagem de programação

A partir dos requisitos modelados, realizou-se análise das tecnologias disponíveis para implementação do *software* Oráculo Estatístico. Decidiu-se pelo desenvolvimento de uma aplicação desktop para ambiente Windows, desenvolvida por meio da linguagem C#, tendo em vista que o código poderá ser portado para outras plataformas (web e mobile). A IDE (*Integrated Development Environment*) utilizada para desenvolvimento é o *Microsoft Visual Studio 2017 Community Edition* - ferramenta gratuita que apoia o desenvolvimento do *software*.

O Oráculo estatístico foi desenvolvido em C# (C-Sharp), que é uma linguagem de programação multiplataforma desenvolvida pela Microsoft, usada para desenvolver aplicativos de desktop, web, jogos entre outros. Essa linguagem roda em ambiente Windows utilizando o .NET Framework (47). Trata-se de uma linguagem orientada para objetos e é fortemente tipada e altamente frequente em programação, o que lhe confere uma biblioteca robusta que permite desenvolver vários tipos de aplicativos (48,49) . Há

uma biblioteca estatística disponível (50), que é amplamente utilizada e confiável. Essa linguagem é bem estruturada, o que facilita a programação e garante uma maior confiabilidade de funcionamento, minimizando erros de sistema (51). No processo, foi utilizado o C# 5.0 e o .NET Framework 4.5 (Figura 5). A escolha dessa versão deve-se por ela ser considerada “robusta” e bem suportada pelas máquinas que operam com o sistema operacional Windows (51).

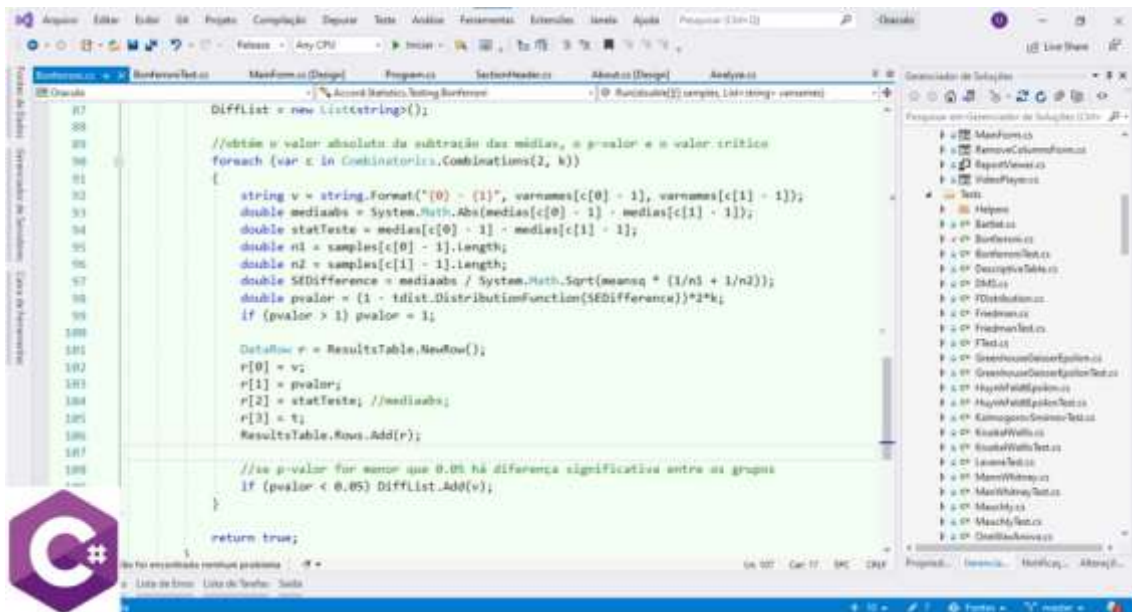


Figura 5 – C#.
Fonte: O próprio autor.

Aperfeiçoamento e registro

Para o aperfeiçoamento do *software*, necessidades foram identificadas e itens implementados gradualmente. Periodicamente uma nova versão era disponibilizada para testes pelos pesquisadores colaboradores. Em reuniões eram reavaliadas as implementações, o que é previsível em projetos que adotam abordagem evolucionária. Aspectos vinculados à interação, como disposição de elementos na interface gráfica, erros e mensagens de erro foram também analisados, cabendo aos desenvolvedores decidirem quais requisitos seriam prioritários, assim como inclusões ou modificações que seriam reapresentados na reunião seguinte.

A gestão da configuração do *software* se deu pelo uso do sistema de versionamento de código Git (Figura 6). Um repositório *online*, replicado localmente em um computador

destinado ao projeto que mantém todas as versões do código. Para a gestão dos requisitos, utilizou-se a ferramenta online Trello. Nela, foram enumerados os requisitos não implementados (*backlog*) e foi realizado o acompanhamento daqueles em implementação em cada iteração, assim como o registro dos já implementados. Dessa forma, a atualização constante do *software* foi realizada, até que a versão final foi alcançada.

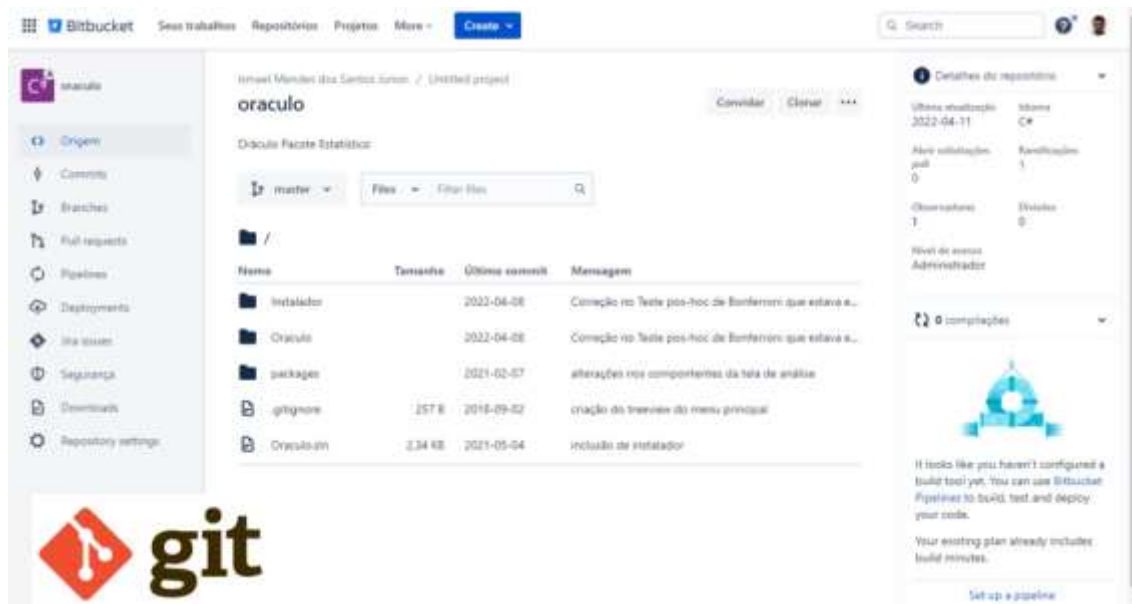


Figura 6 – GIT.

Fonte: O próprio autor.

Testes

Durante o processo de desenvolvimento do *software*, foram realizados testes de comparação onde se testou o Oráculo Estatístico e comparou seus resultados com os do Microsoft Excel (37), R (39) e o PASW *Statistics* (40). Foram utilizados bancos de dados, de vários tamanhos, inclusive dados de bases públicas como o IMRS (índice mineiro de responsabilidade social) da fundação João Pinheiro (52). O objetivo destes testes foi verificar se os resultados obtidos com Oráculo Estatístico eram os mesmos dos demais *softwares*.

Instalação e instruções de uso

O Oráculo estatístico deve ser instalado no computador do usuário, para isso basta entrar na página do PPGCS através do seguinte link: <http://www.ppgcs.unimontes.br/ppgcs>, nesta página há vários produtos desenvolvidos no programa de pós graduação, basta então clicar no botão Oráculo estatístico e será direcionado para outra página (http://www.ppgcs.unimontes.br/ppgcs/index.php/pt-br/?option=com_sppagebuilder&view=page&id=195). A partir do link é necessário o *download* do programa e, posteriormente, ativá-lo pelo ícone criado na área de trabalho. Caso o Windows instalado no computador do usuário esteja com o pacote .Net Framework desatualizado, será ofertada a atualização desse pacote como pré-requisito.

Apesar de ser considerado um software intuitivo, foi criado um vídeo com instruções de uso no youtube que pode ser acessado através do endereço eletrônico: <https://www.youtube.com/watch?v=l0MzREuu4eI> ou dentro do próprio software (Figura 7)



Figura 7 – Vídeo com instruções de Uso.
Fonte: O próprio autor.

RESULTADOS E DISCUSSÃO

Apresentação do Software

O nome Oráculo Estatístico surgiu da ideia de se prever o que o usuário deseja fazer de acordo com as características da amostra. Foi criada uma logomarca que busca materializar os sentimentos supracitados (Figura 8). O círculo simboliza uma bola de cristal, objeto usado para fazer previsões. A tonalidade em azul de seu *layout* foi escolhido intuitivamente por semioticamente transmitir tranquilidade e serenidade (53). Dentro do círculo há três figuras gráficas com cores diferentes, simbolizando três grupos diferentes com distribuição normal (Gaussiana). Uma das distribuições mais importantes na estatística.



Figura 8 - Logomarca do software Oráculo Estatístico.
Fonte: O próprio autor.

O Oráculo Estatístico foi desenvolvido inicialmente (versão 1.0) como um *software* estatístico para a solução de problemas que envolvem testes de hipótese de comparação de médias (testes paramétricos) e seus respectivos testes não paramétricos. Entretanto, pretende-se que esse seja um *software* evolutivo, onde outros testes sejam implementados de forma gradativa. Nesse sentido, estratégias são estudadas para sua difusão, bem como a disponibilização dos códigos para melhoria contínua por meio de seus usuários e de acordo com suas necessidades e demandas. Isso em consonância as melhorias contínuas do proponente idealizador.

Desenvolvimento do Oráculo Estatístico

No desenvolvimento deste *software* buscou-se primeiro compreender como funcionam os testes de hipóteses para média (testes paramétricos) e seus requisitos (pressupostos), assim como a quantidade de grupos que se deseja comparar, normalidade, igualdade de variância, esfericidade e os testes não paramétricos correspondentes.

Foi criado o Fluxograma 1 que demonstra o passo a passo necessário para a utilização de cada teste. Ele que serviu de base para o desenvolvimento da lógica do programa, fazendo com que, com base nos resultados encontrados, o Oráculo escolha o próximo passo de forma quase automática; com mínima intervenção humana.

Se por exemplo, um pesquisador desejar realizar um teste com três grupos, de acordo com o Fluxograma 1, o primeiro passo é escolher os três grupos para comparação. O sistema então automaticamente perguntará se os dados são “independentes” ou “dependentes”, cabendo ao pesquisador fornecer essa informação. Caso o pesquisador informe que é “independente”, o próximo passo é calcular a normalidade de cada grupo.

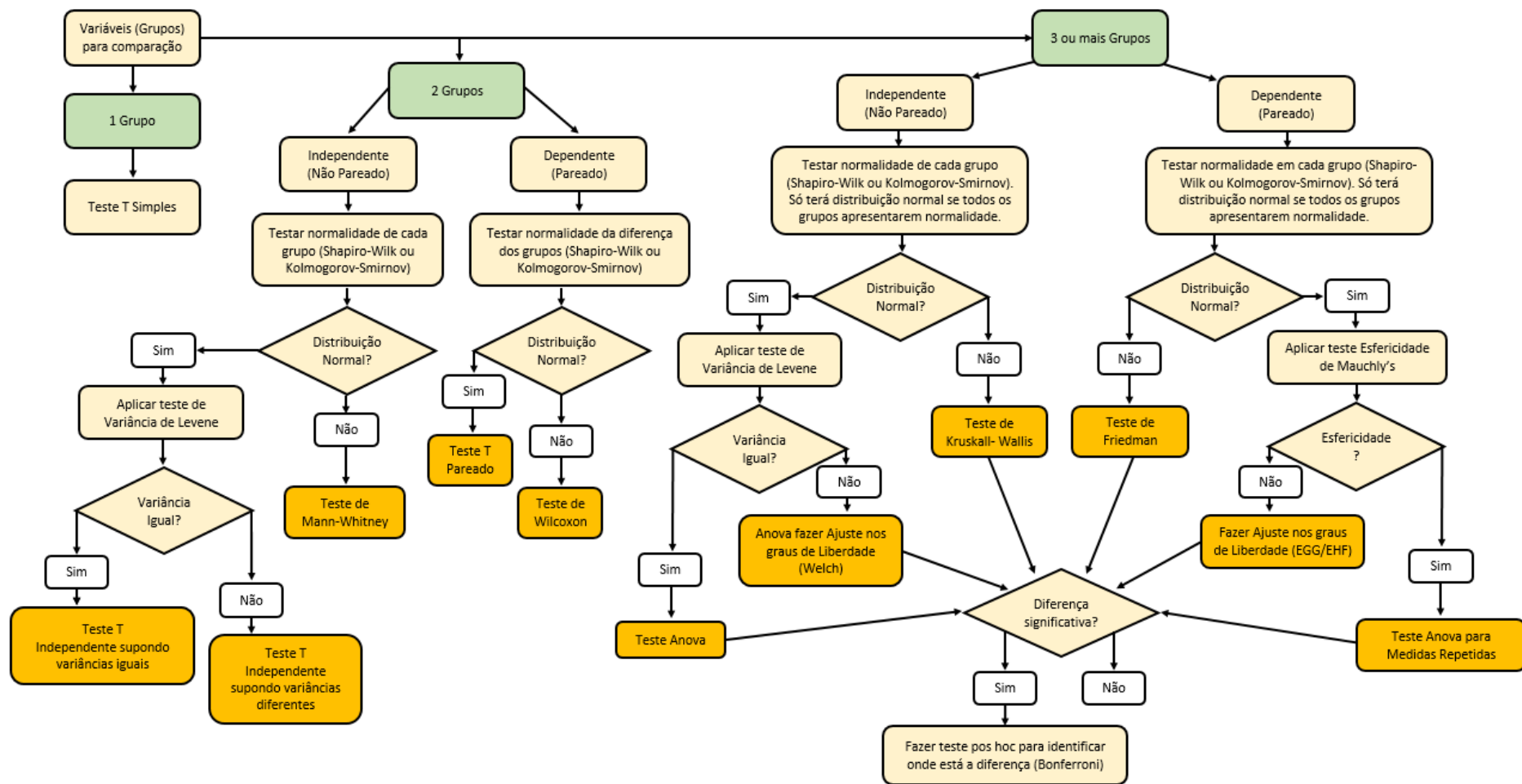
Pelo teorema do limite central, a distribuição amostral da média se aproxima da distribuição normal quando se tem uma amostra grande (54), entretanto, testes estatísticos de normalidade com amostras grandes demonstraram que nem sempre essas suposições de normalidade se confirmam (55). Para resolver esse problema deve-se verificar se a distribuição é ou não normal. Para isso utilizou-se os testes de Kolmogorov-Smirnov e o de Shapiro-Wilk, quando o n é menor ou igual a 50 utiliza-se o teste de Shapiro-Wilk e

se o n for maior que 50 utiliza-se o teste de Kolmogorov-Smirnov , os quais comparam escores de uma amostra com uma distribuição normal modelo (13,56).

Tendo em vista que o Oráculo Estatístico se utiliza dos testes de Kolmogorov-Smirnov e o de Shapiro-Wilk para calcular a normalidade, caso todos os grupos apresentem distribuição normal em todos os grupos, o *software* vai automaticamente para o passo seguinte: realizar o teste de igualdade de variância. Se a normalidade não for encontrada o *software* então direciona os dados para o teste de Kruskal-Wallis (teste não paramétrico).

A variância constante ou homogeneidade de variâncias (homocedasticidade) é um requisito necessário para se fazer a análise de variância (ANOVA) (57). Para se calcular a variância existem testes alternativos, como o Teste F e o Teste de Levene. O teste F é apropriado para variáveis com distribuição normal, enquanto o Teste de Levene pode ser usado com qualquer distribuição (58). Na aplicação do Teste de Levene, caso o resultado demonstre igualdade de variância, é realizado o teste ANOVA, caso contrário, realiza-se a ANOVA com ajustes nos graus de liberdade, como proposto por Welch (59). Realizado o teste ANOVA, se for encontrada uma diferença significativa, o *software* irá fazer um Teste post hoc para identificar onde estão situadas as diferenças. São realizadas comparações entre todos os pares possíveis entre os grupos. Para corrigir o problema do crescimento do erro de TIPO I, pode-se dividir o valor de α pelo número de comparações. Esse método é conhecido como correção de Bonferroni (13).

Toda a lógica de programação do Oráculo Estatístico foi baseada no Fluxograma 1. Nele estão indicadas as tarefas que o sistema deverá executar com base na informação recebida, seja por meio do usuário ou via parametrização do sistema.



Fluxograma 1 – Testes de Hipóteses.
 Fonte: O próprio autor

Para facilitar o uso deste *software* pelo usuário, buscou-se uma lógica na apresentação de telas, de modo a deixá-las mais intuitivas.

Antes da escrita do programa, foi desenhada uma lógica de programação baseada na interação com o usuário, ou seja, foi racionalizado como seria o *layout* das telas, seu conteúdo, sua apresentação e a sequência de cada uma, de acordo com a perspectiva do usuário. A interação com o computador é fundamental para o processo de criação de um *software* (60). Existe também outra face da lógica de programação, que é o que o *software* deve fazer para cada instrução recebida. O programador deve fazer com que o sistema entenda o que o usuário está pedindo, em alguns casos, até prever o que o usuário necessita. Essas duas partes do desenvolvimento devem ser realizadas em total coerência, para não haver conflito e mau funcionamento do *software*. A lógica é estruturada de modo que o usuário do *software* não precise pensar muito para utilizá-lo, mas alcance o resultado necessário (61).

Durante o desenvolvimento do Oráculo Estatístico buscou-se desenvolver telas simples e intuitivas, utilizando um conceito de telas com visualizações rápidas propostas por Krug (62). A parte de inserção de dados tem várias perguntas e confirmações para evitar algum erro nessa fase crucial.

A parte de análise dos dados seguiu a lógica proposta no fluxograma 1, onde o usuário escolhe os grupos que deseja comparar. O *software* conta a quantidade de grupos, e caso seja apenas um, ele entende que vai comparar aquele grupo com alguma média já conhecida. Ele então, solicita ao usuário para informar a média e já retorna o cálculo com o relatório.

Caso o usuário escolha mais de um grupo para análise o *software* pergunta se é pareado ou não e na caixa de pergunta aparece um ícone com uma explicação conceitual, caso o usuário tenha dúvida na opção a ser escolhida. A partir desse momento é só clicar no botão analisar que o *software* faz todos os cálculos necessários de forma automática retornando tabelas com a estatística descritiva dos dados, normalidade, homogeneidade/esfericidade, teste de hipótese adequado ao caso e até mesmo o teste pós hoc quando necessário.

Além das tabelas com os resultados calculados o Oráculo Estatístico traz um relatório explicativo que contribui para o aprendizado sobretudo de alunos da iniciação científica ajudando a entender o processo e a interpretar os resultados encontrados.

Telas do oráculo estatístico

Como todo programa de computador, o usuário deverá clicar em um ícone para abri-lo. Na primeira tela há uma mensagem de boas-vindas e feita uma apresentação do Oráculo Estatístico ao usuário. Logo abaixo há um botão para iniciar o *software* (Figura 9).

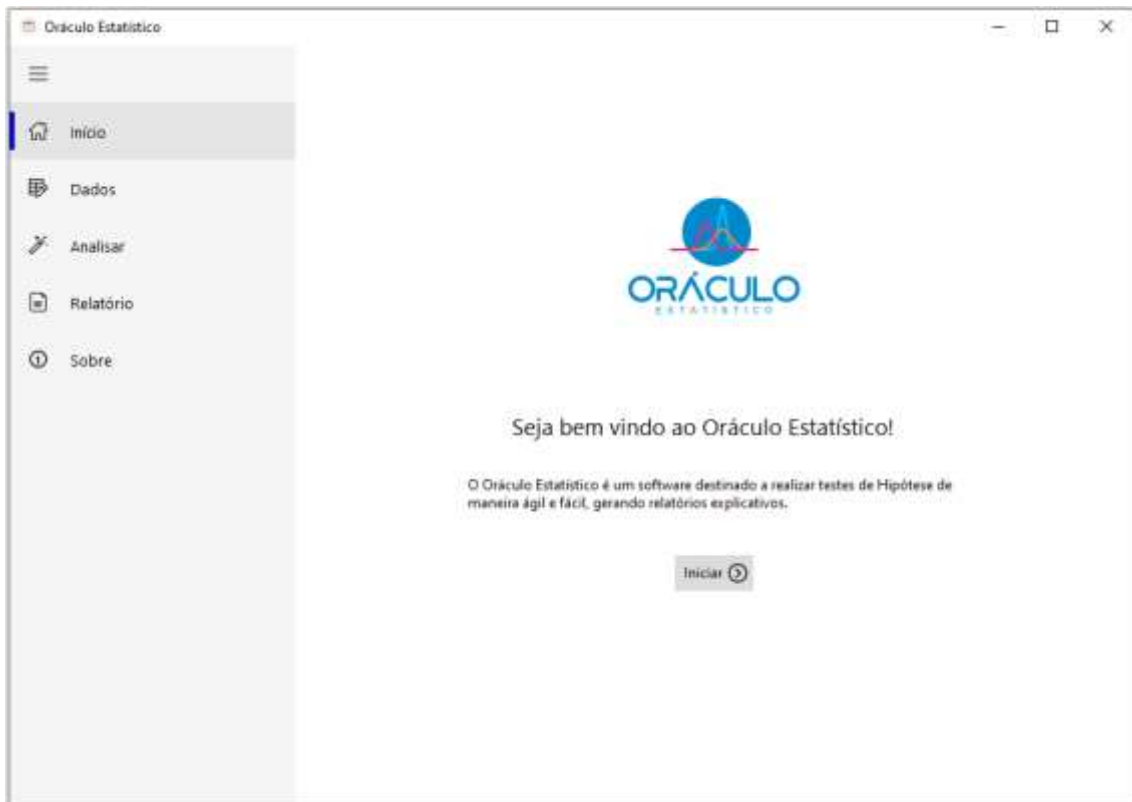


Figura 9 – Tela inicial do Software.
Fonte: O próprio autor.

Apesar de ser desenvolvido buscando-se uma boa usabilidade e o arranjo de suas telas ser autoexplicativo, optou-se, na segunda tela, por disponibilizar um vídeo demonstrando o uso do *software*. Caso o usuário não queira assistir o vídeo, ele pode inserir os dados que pretende trabalhar. Na aba lateral do *software*, denominada dados, o usuário pode importar arquivos nos formatos csv (valores separados por vírgula) ou xls (planilha do Excel). Caso prefira, o usuário pode optar por digitar os dados diretamente no sistema (Figura 10).

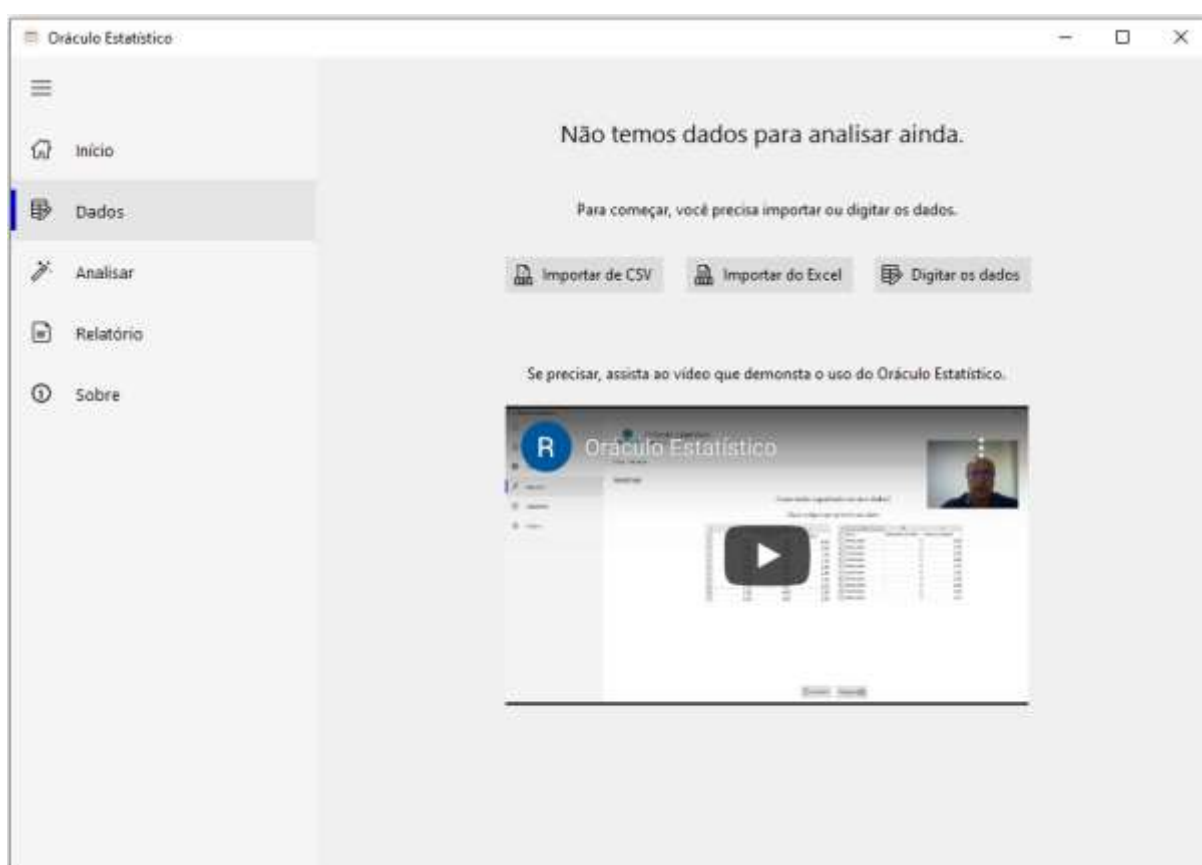


Figura 10 – Tela inicial de dados.
Fonte: O próprio autor.

Para inserir os dados de um arquivo (csv ou xls), o usuário pode localizar a pasta onde está o arquivo, selecioná-lo e clicar em “ok”, ou com a pasta já aberta, clicar no arquivo e arrastá-lo diretamente para dentro do *software* (Figura 4). Após isso, basta clicar em **Próximo ▶**. O usuário poderá também retornar para a tela anterior a qualquer momento, bastando para isso clicar no ícone **◀ Anterior**. Esses ícones foram pensados para facilitar a experiência do usuário com o Oráculo Estatístico, oferecendo maior usabilidade ao *software* (Figura 11).

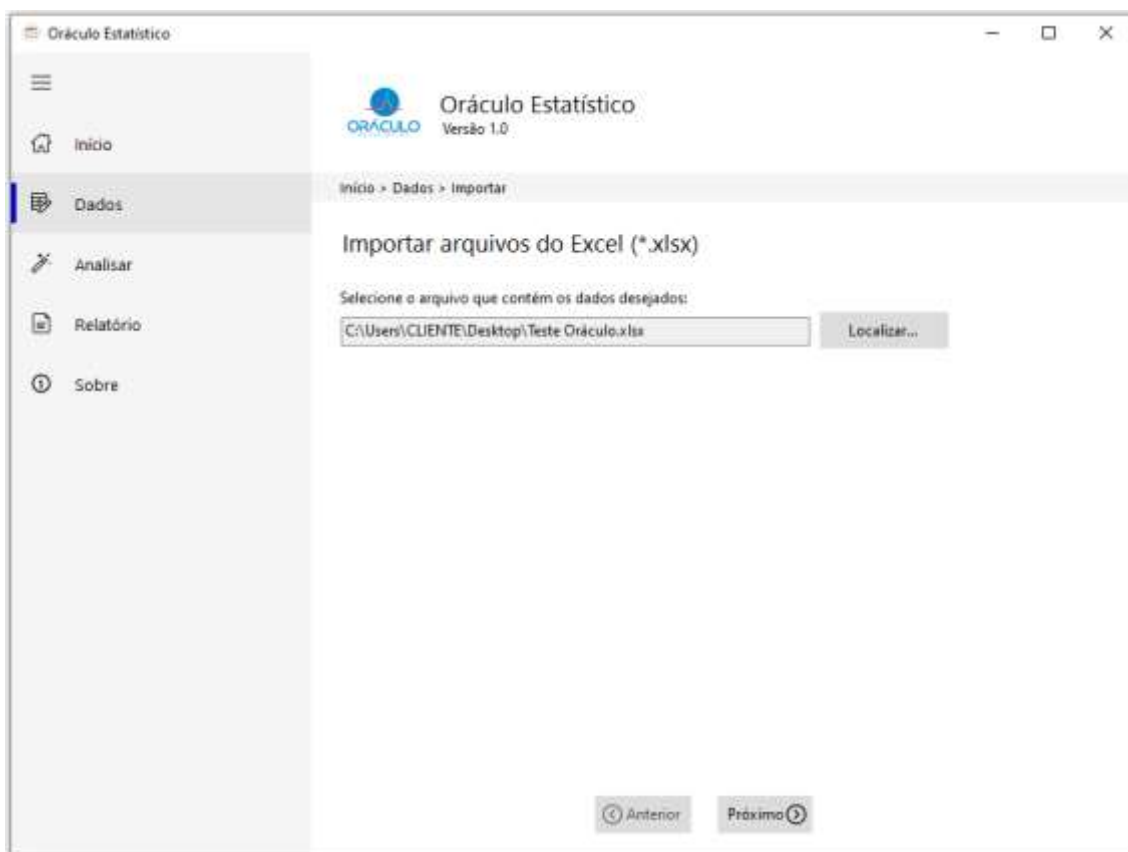


Figura 11 – Importação de arquivos do Excel.
Fonte: O próprio autor.

Caso o arquivo Excel escolhido tenha mais de uma planilha, o Oráculo Estatístico pede para escolher a planilha onde estão os dados que se deseja analisar. Basta clicar na planilha desejada e, posteriormente, clicar em **Próximo ▶** (Figura 12).

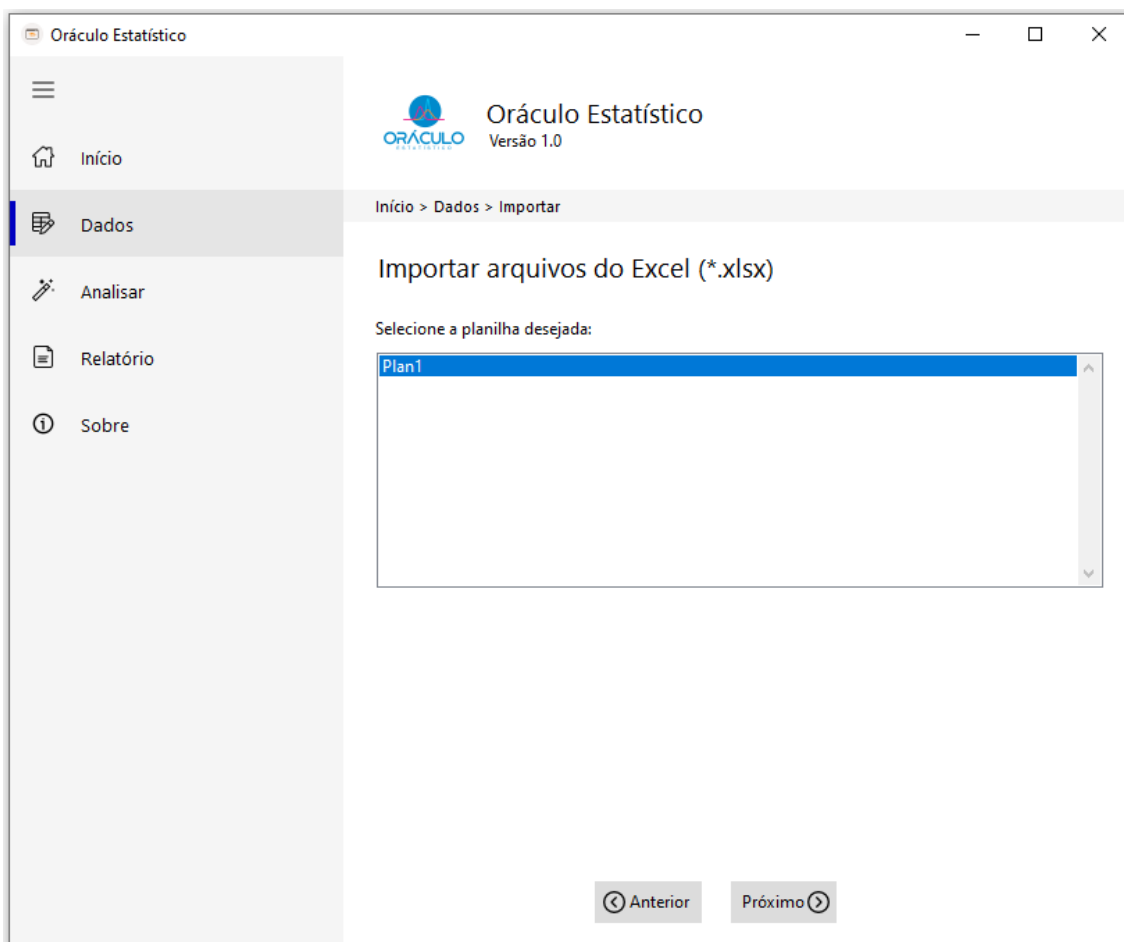


Figura 12 – Escolha da Planilha.
Fonte: O próprio autor.

Após escolhida a planilha que se deseja trabalhar, o *software* questiona se a primeira linha contém os títulos do campo (Figura 13). Caso a resposta seja **sim**, ele importa os dados considerando a primeira linha como o título de cada coluna. Caso o usuário selecione a opção **não**, o *software* irá nomear automaticamente as colunas; Coluna0, Coluna1, Coluna2 e assim sucessivamente.

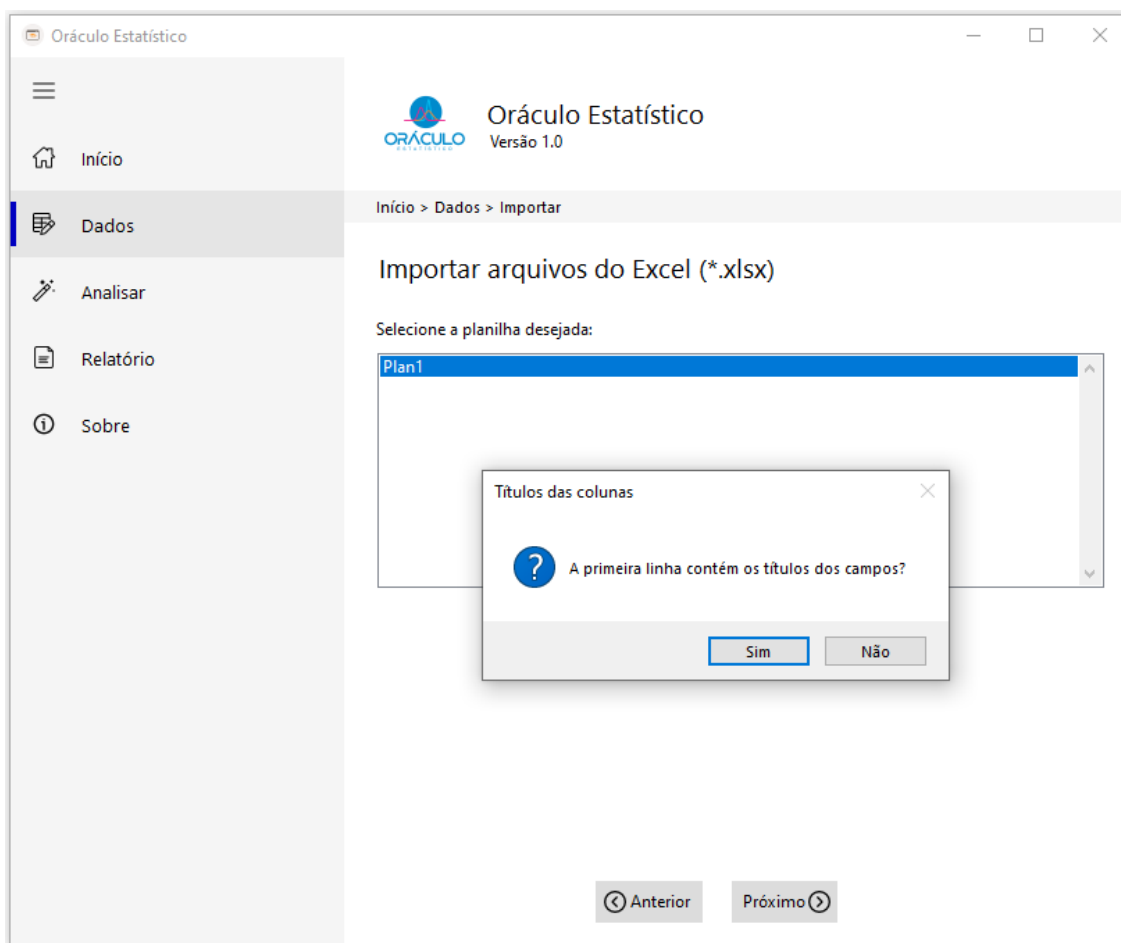


Figura 13 – Mensagem de configuração dos dados.
Fonte: O próprio autor.

Após a confirmação da primeira linha, como título ou não, os dados irão aparecer na tela em forma de tabela. Se os dados estiverem corretos, basta clicar em **Concluir ▶**, caso esteja errado, o usuário poderá clicar no ícone **◀ Anterior** (Figura 14).

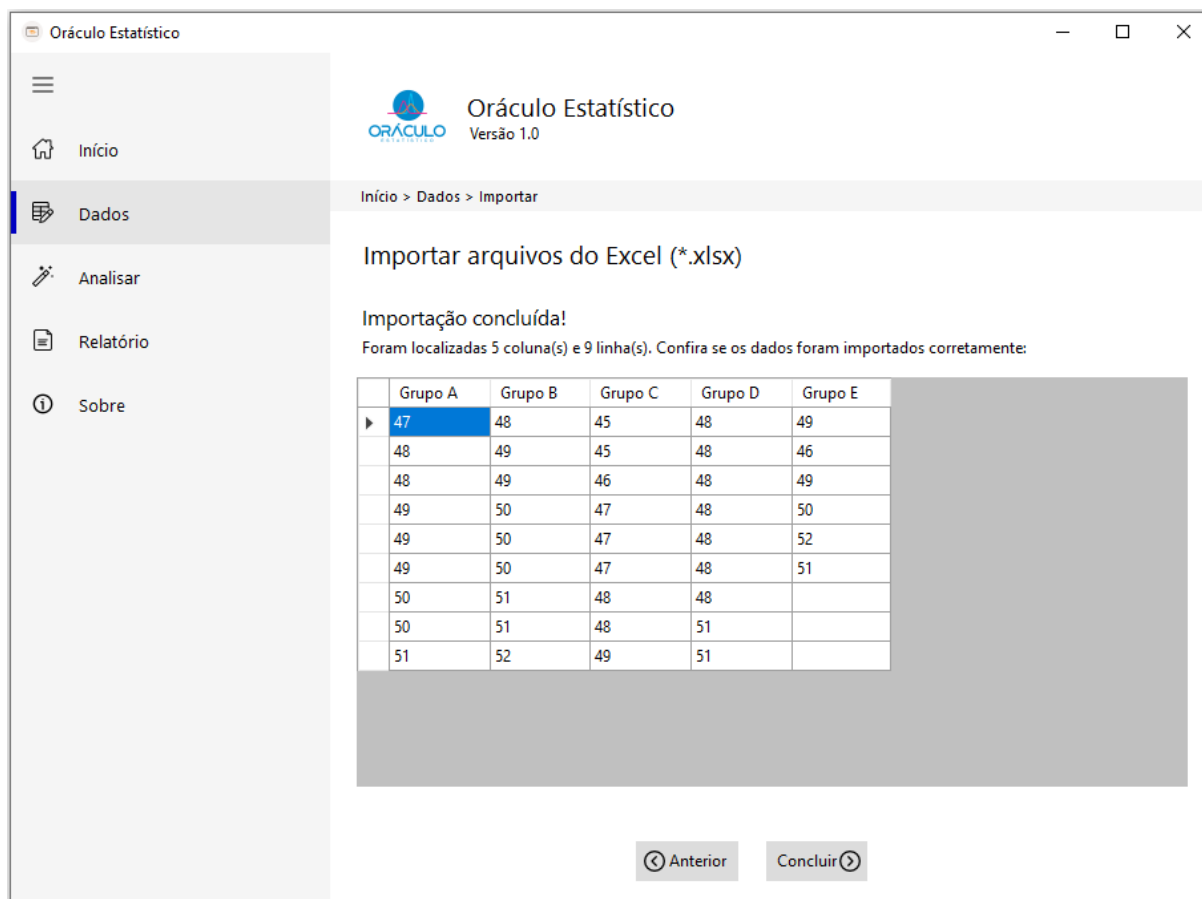


Figura 14 – Dados importados.
Fonte: O próprio autor.

Levando-se em consideração a importância da base de dados para a análise, e que qualquer erro nesta compromete os resultados, antes de seguir, o Oráculo Estatístico confirma novamente se os dados importados estão corretos. Essa medida de segurança ajuda a garantir que os dados estejam corretos. O usuário poderá confirmar clicando no ícone **Sim**, ou não confirmar clicando em **Não**, o que lhe permite revisar e corrigir qualquer erro. Após tudo confirmado, basta clicar em **Concluir** (Figura 15).

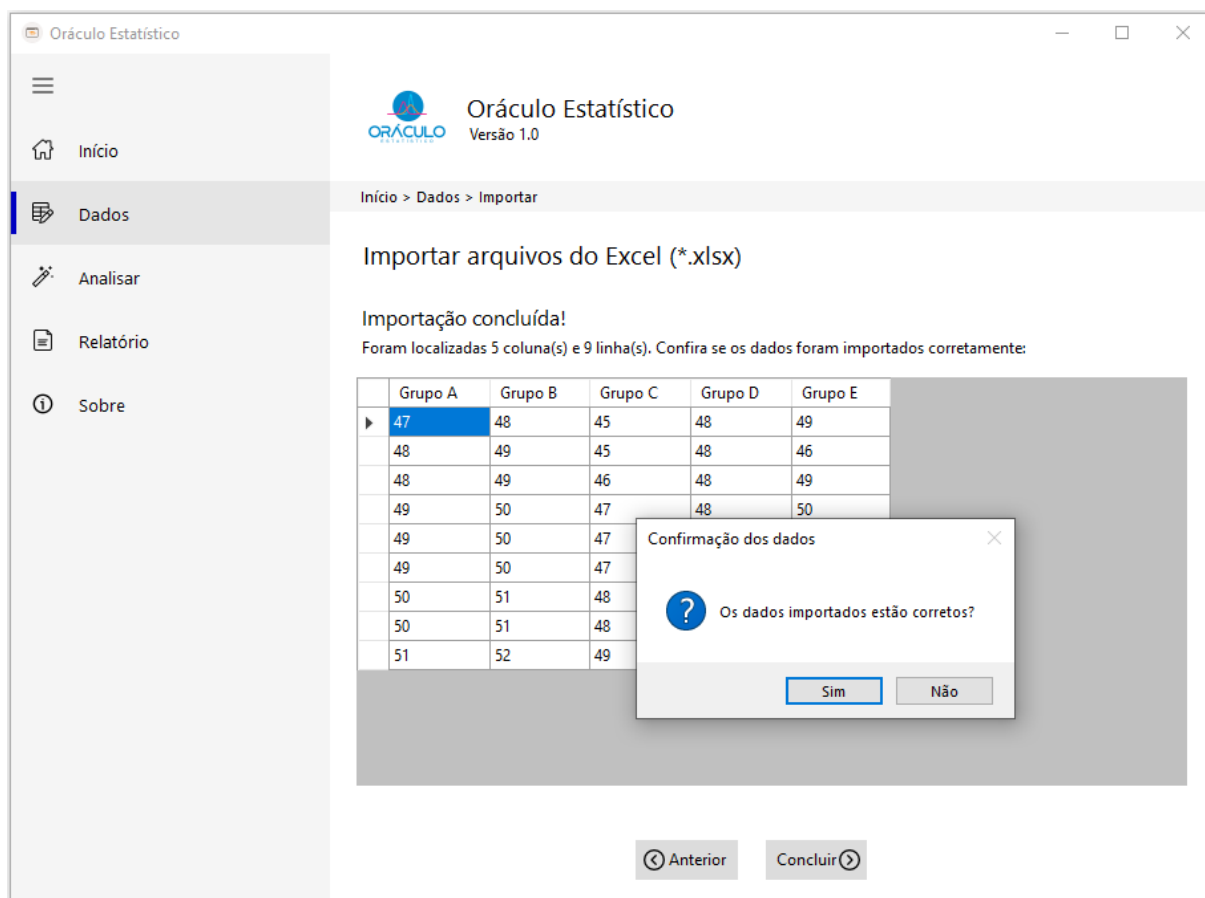


Figura 15 – Mensagem de Confirmação.
Fonte: O próprio autor.

Após a confirmação ou não dos dados, o sistema apresenta os ícones: **Adicionar Coluna**, **Remover Coluna** e **Remover Linhas**. Caso o usuário necessite acrescentar ou retirar algum dado, para adicionar linha basta preencher a linha em branco que aparece no final. Caso precise alterar valor de uma célula, basta clicar nela e mudar o valor. Quando tudo estiver correto o usuário pode fazer a análise clicando no botão **Analisar** (Figura 16).

Oráculo Estatístico

Oráculo Estatístico
Versão 1.0

Início > Dados

Meus dados

Adicionar Coluna... Remover Coluna Remover Linha(s)

	Grupo A	Grupo B	Grupo C	Grupo D	Grupo E
▶	47	48	45	48	49
	48	49	45	48	46
	48	49	46	48	49
	49	50	47	48	50
	49	50	47	48	52
	49	50	47	48	51
	50	51	48	48	
	50	51	48	51	
	51	52	49	51	
*					

Analisar...

Total: 9 linhas e 5 colunas.

Figura 16 – Dados para análise.
Fonte: O próprio autor.

A organização dos dados na forma de planilha pode se tornar um problema para o pesquisador. Alguns pesquisadores organizam os dados em grupos (colunas), outros organizam os dados como se fosse uma lista (em linhas). *Softwares* como o PASW Statistics, quando realiza a análise de dados pareados, exige que os dados estejam separados por coluna. Por outro lado, se os dados forem independentes, devem ser organizados em uma lista, de modo que as linhas sejam identificadas com o grupo a que pertencem. O Oráculo Estatístico trabalha com os dados em grupos, entretanto se o pesquisador tiver feito na forma de lista, basta clicar na opção lista (segunda planilha) e o *software* organizará os dados segundo o agrupamento desejado (Figura 17).

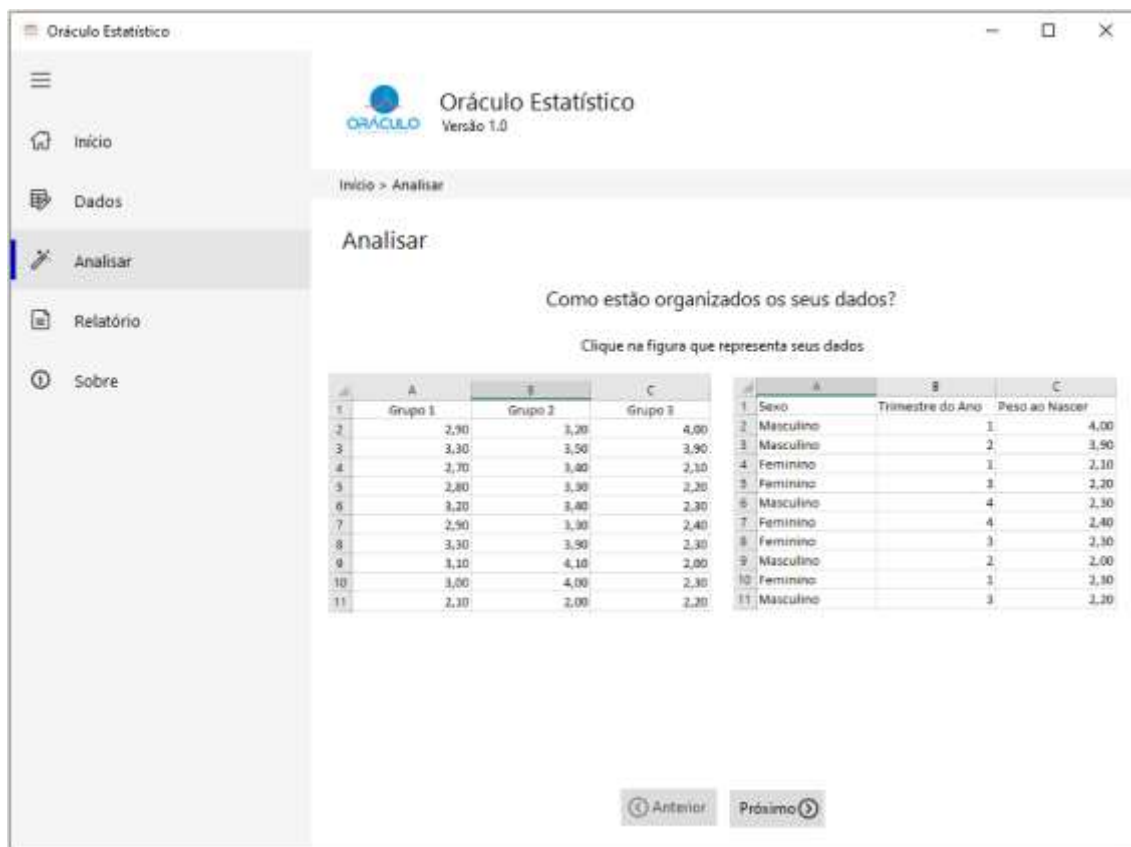


Figura 17 – Organização dos dados.
Fonte: O próprio autor.

Para analisar os dados é necessário escolher a variável que se deseja utilizar, para isso é necessário se transferir as variáveis do campo “Variáveis importadas” para o campo “Variáveis que se deseja utilizar”. Para mudar a variável dentre os quadros, é necessário marcar a variável e clicar na seta (> ou <). Depois de selecionada a variável, clica-se na seta (Figura 18).

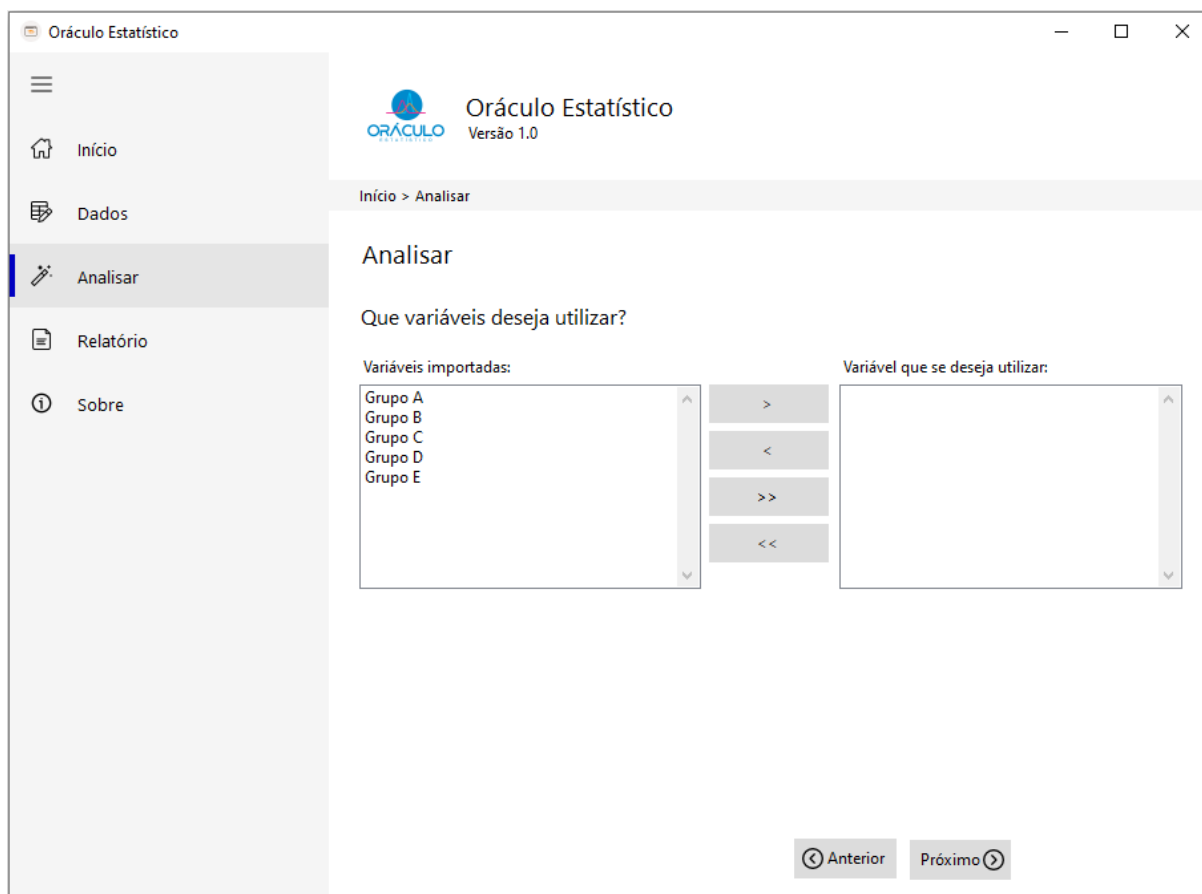


Figura 18 – Analisar.
Fonte: O próprio autor.

Caso o pesquisador escolha apenas uma variável, o *software* irá entender que ele pretende fazer um teste t para uma amostra, comparando o grupo amostral com uma média já estabelecida, podendo ser a média de um outro estudo, e irá abrir uma caixa para digitar a média populacional. Após inserir a média que se deseja fazer a comparação, é necessário clicar em **Concluir** ► (Figura 19).

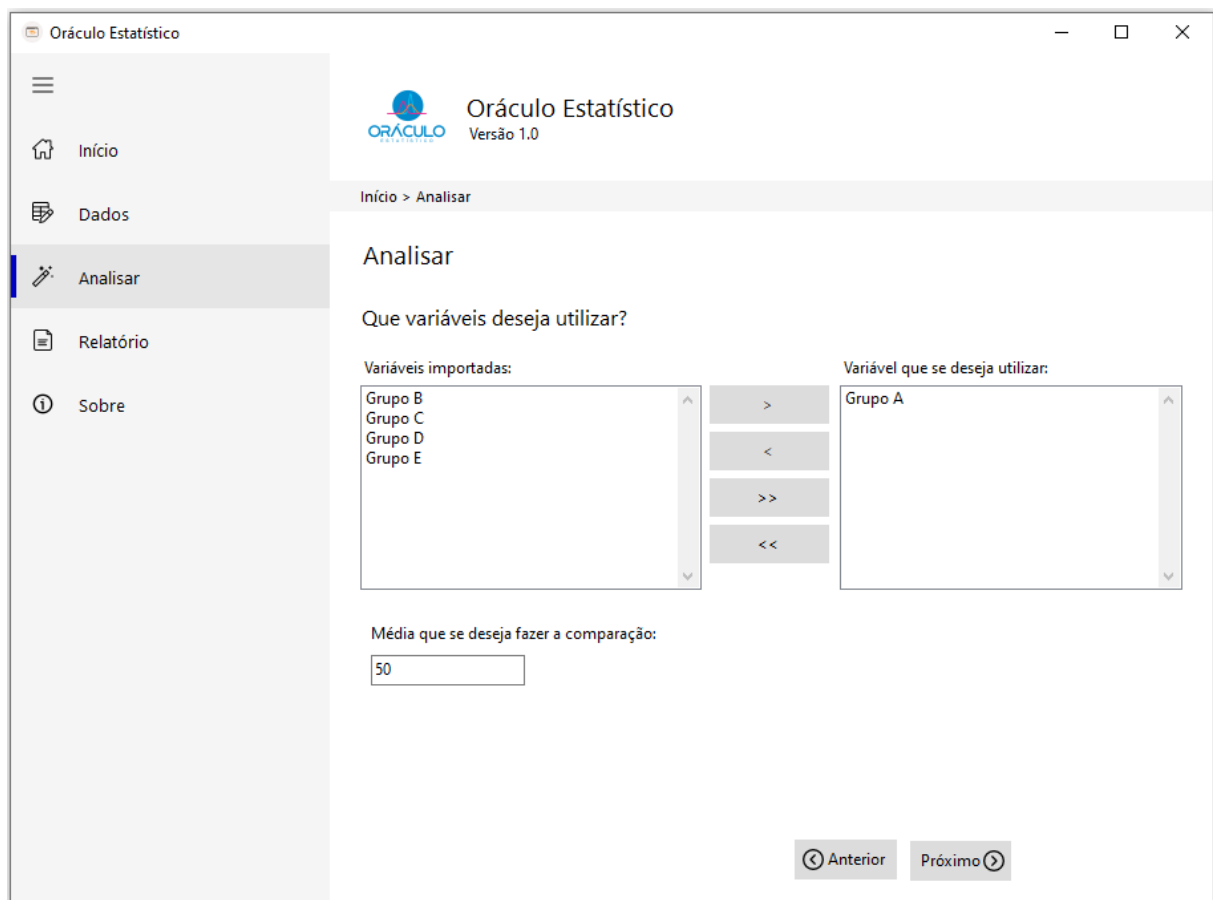


Figura 19 – Análise de uma variável.
Fonte: O próprio autor.

A Figura 20 mostra o resultado fornecido pelo Oráculo Estatístico, onde o resultado da Tabela 1 apresenta a estatística descritiva das variáveis do grupo A.

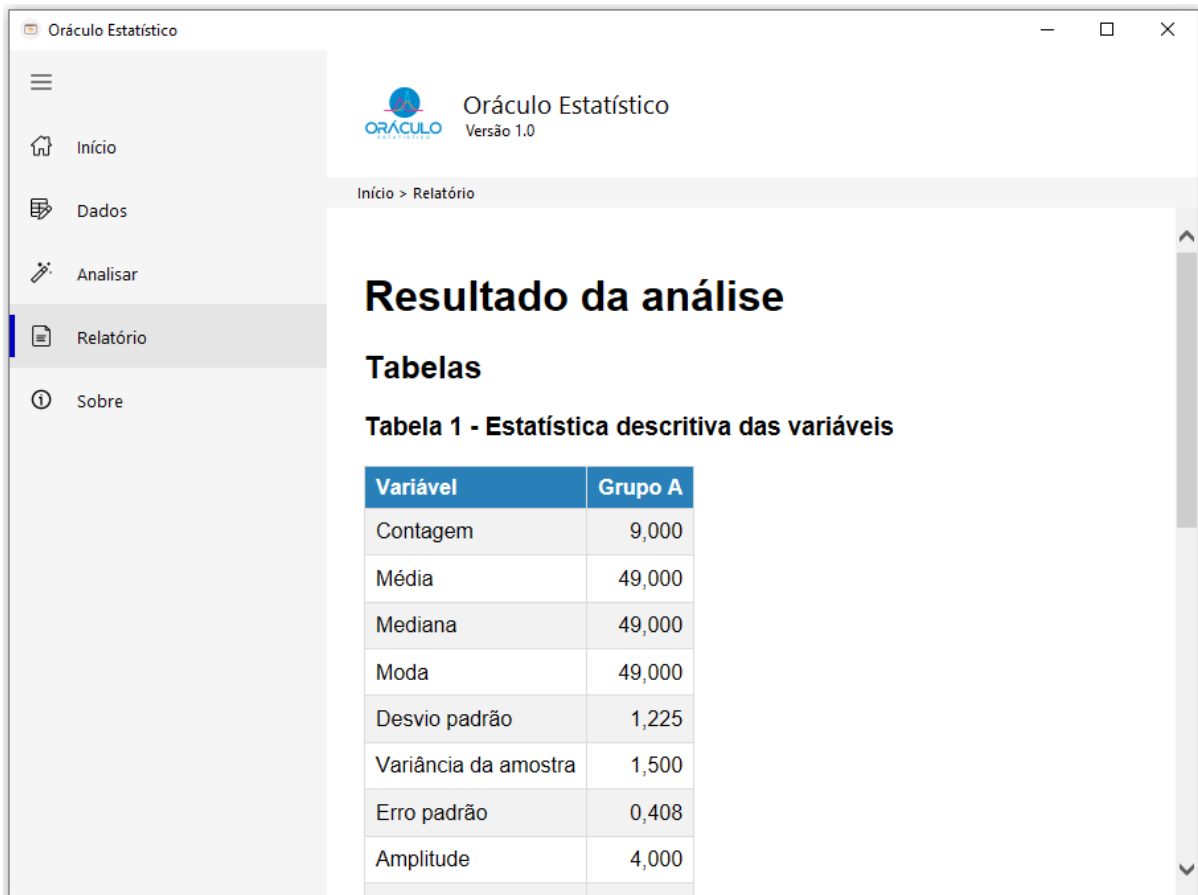


Figura 20 – Resultado da Análise de uma variável.
Fonte: O próprio autor.

A Figura 21 é a continuação do resultado da análise do grupo A, ela apresenta a Tabela 2, com resultado do teste de hipótese. O Relatório apresenta um texto explicativo, contendo o nome do teste que foi realizado, o resultado da análise, como reportá-lo e a forma de citação do *software*.



Oráculo Estatístico

Oráculo Estatístico
Versão 1.0

Início > Relatório

Assimetria	0,000
Soma dos valores	441,000

Tabela 2 - Teste de hipótese

Grupo	p-valor	Valor calculado	Valor crítico
Grupo A	0,040	2,449	2,306

Relatório

O teste escolhido foi o Teste T Simples. Este teste permite testar se a média de dois grupos é significativamente diferente.

O resultado obtido no Teste T foi um p-valor de 0,040. Portanto, menor que 0,05. Sendo assim, para um nível de confiança de 95%, pode-se dizer que existe diferença significativa para os dados analisados.

Como citar?

Silva, R. C. F.; Santos Junior, I. M.; Silveira, M. F.; Monteiro Junior, R. S. (2022). Oráculo Estatístico [Computer software].

Figura 21 – Resultado da Análise de uma variável.
Fonte: O próprio autor.

Caso o usuário escolha mais de um grupo de dados, o Oráculo Estatístico irá perguntar se os dados são pareados ou não pareados; caso o usuário não saiba a diferença, existe um ícone de informação (I). Tal ícone apresenta informações para ajudar o usuário na definição (conceito) de dados pareados e não pareados. Após escolher as variáveis (grupos que se deseja comparar) e definido se estes são pareados ou não, o usuário clica em **Próximo** (Figura 22).

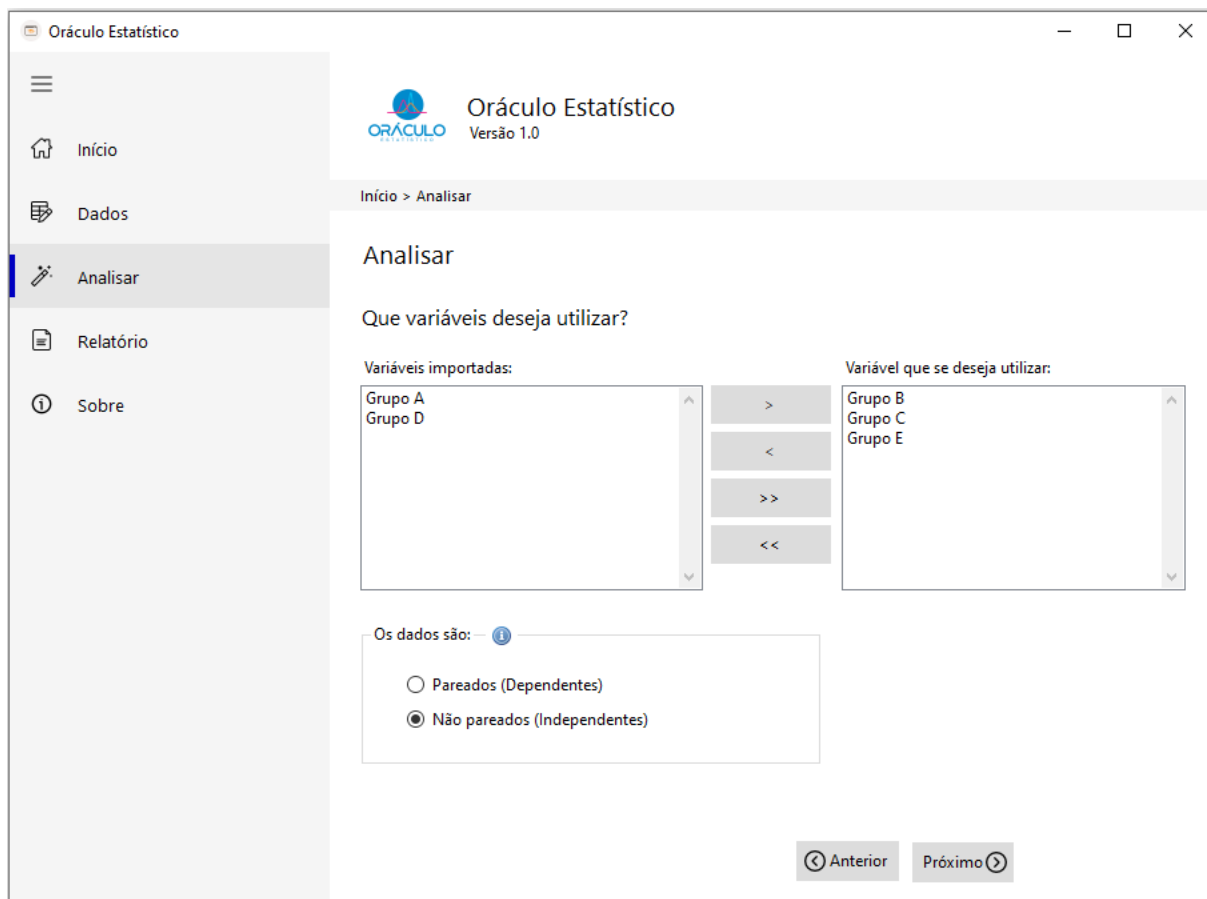


Figura 22 – Análise de mais de uma variável.
Fonte: O próprio autor.

Após fazer todos esses cálculos o Oráculo Estatístico retorna para o usuário uma tela de resultado, onde mostra uma tabela com uma estatística descritiva das variáveis (Figura 23).



Figura 23 – Resultado da Análise de três variáveis (Parte 1).

Fonte: O próprio autor.

Além da tabela com uma estatística descritiva das variáveis o Oráculo Estatístico retorna para o usuário os cálculos de alguns pressupostos necessários aos testes de hipóteses, assim como os testes de normalidade, de igualdade e de variância (Figura 24).

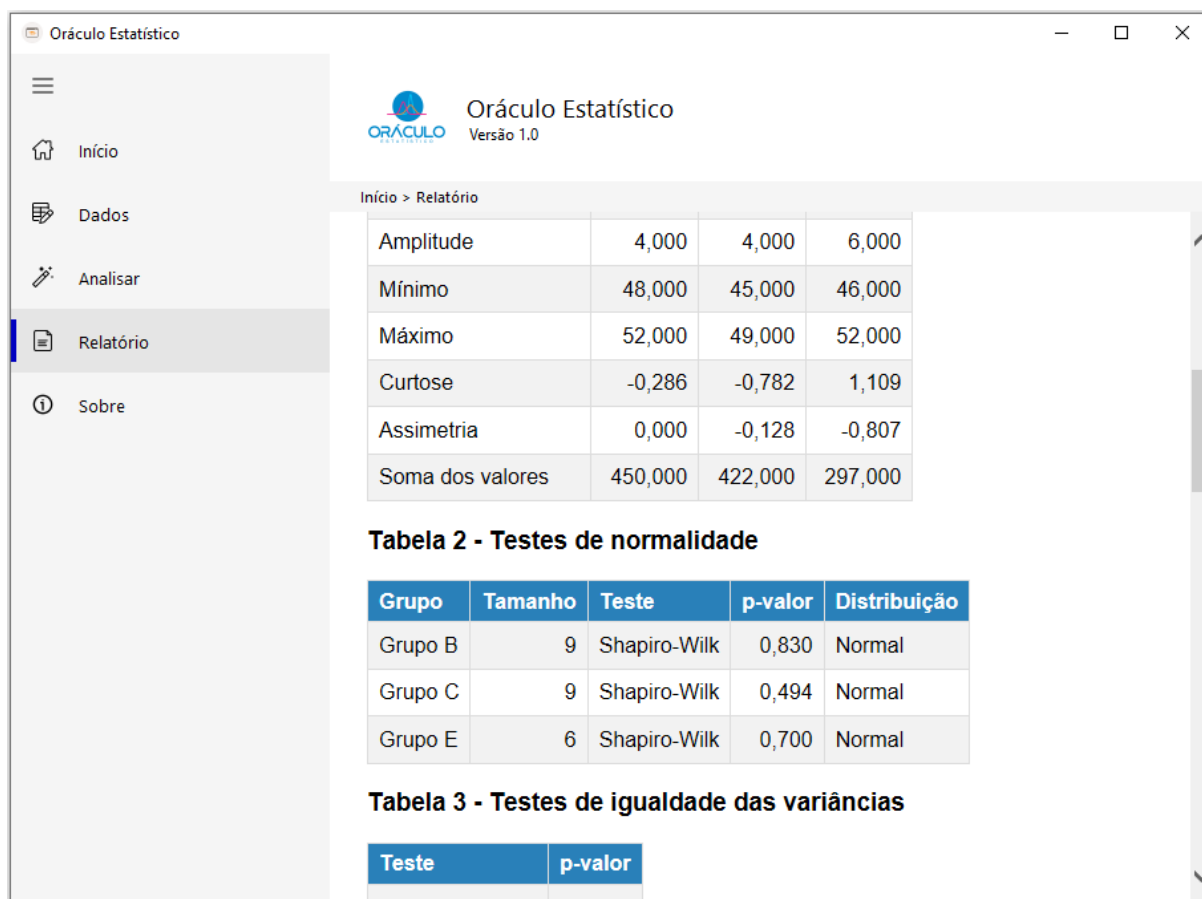


Figura 24 – Resultado da Análise de três variáveis (Parte 2).
Fonte: O próprio autor.

O Oráculo Estatístico verifica os pressupostos (normalidade e igualdade de variância) para um teste ANOVA, caso esses pressupostos tenham sido atendidos o *software* realiza o teste ANOVA. Se o resultado identificar a existência de diferença significativa, automaticamente será realizado o teste *post-hoc* de Bonferroni para detectar onde estão situadas as diferenças (Figura 25).

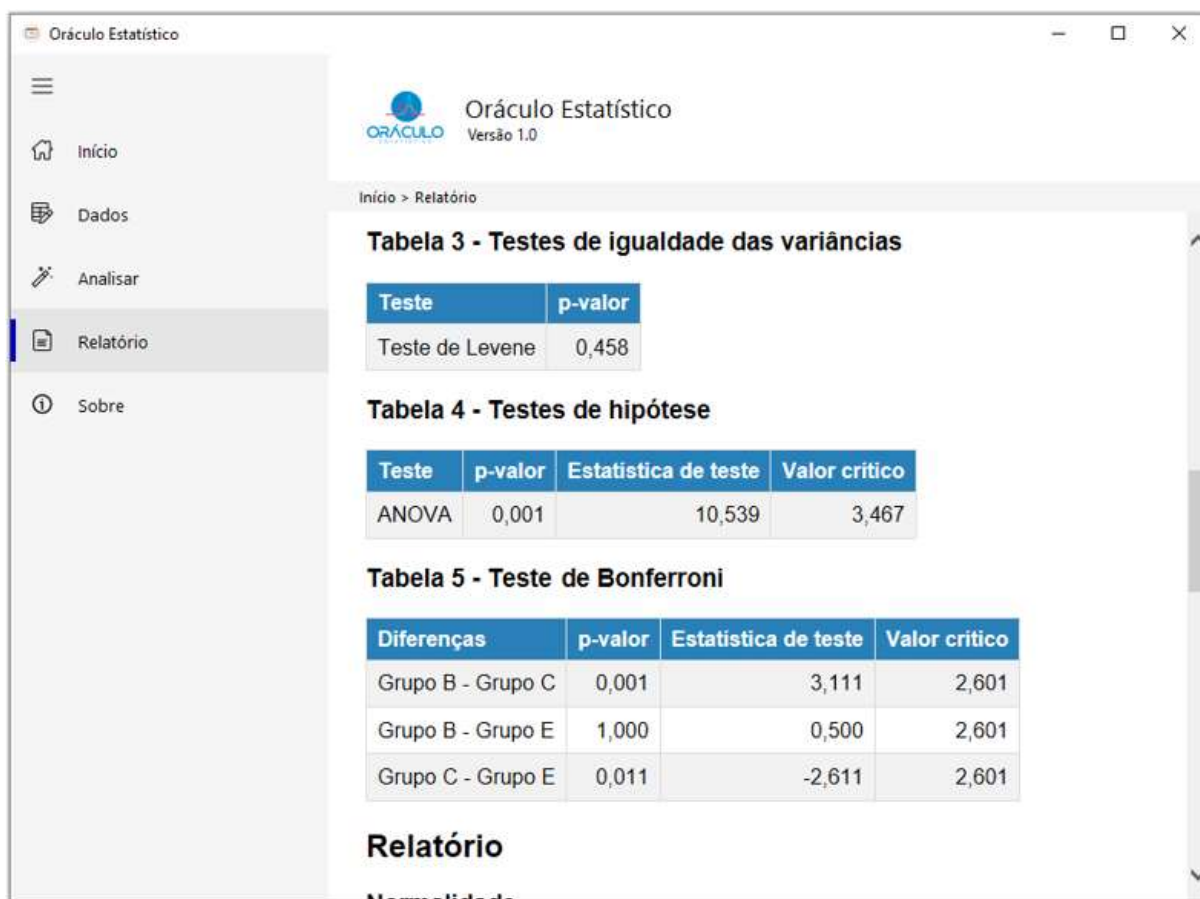


Figura 25 – Resultado da Análise de três variáveis (Parte 3).
 Fonte: O próprio autor.

Por fim, o Oráculo Estatístico apresenta o relatório contendo o nome do teste realizado e o resultado alcançado com interpretação. Na Figura 26 pode-se observar o relatório para o teste de normalidade.

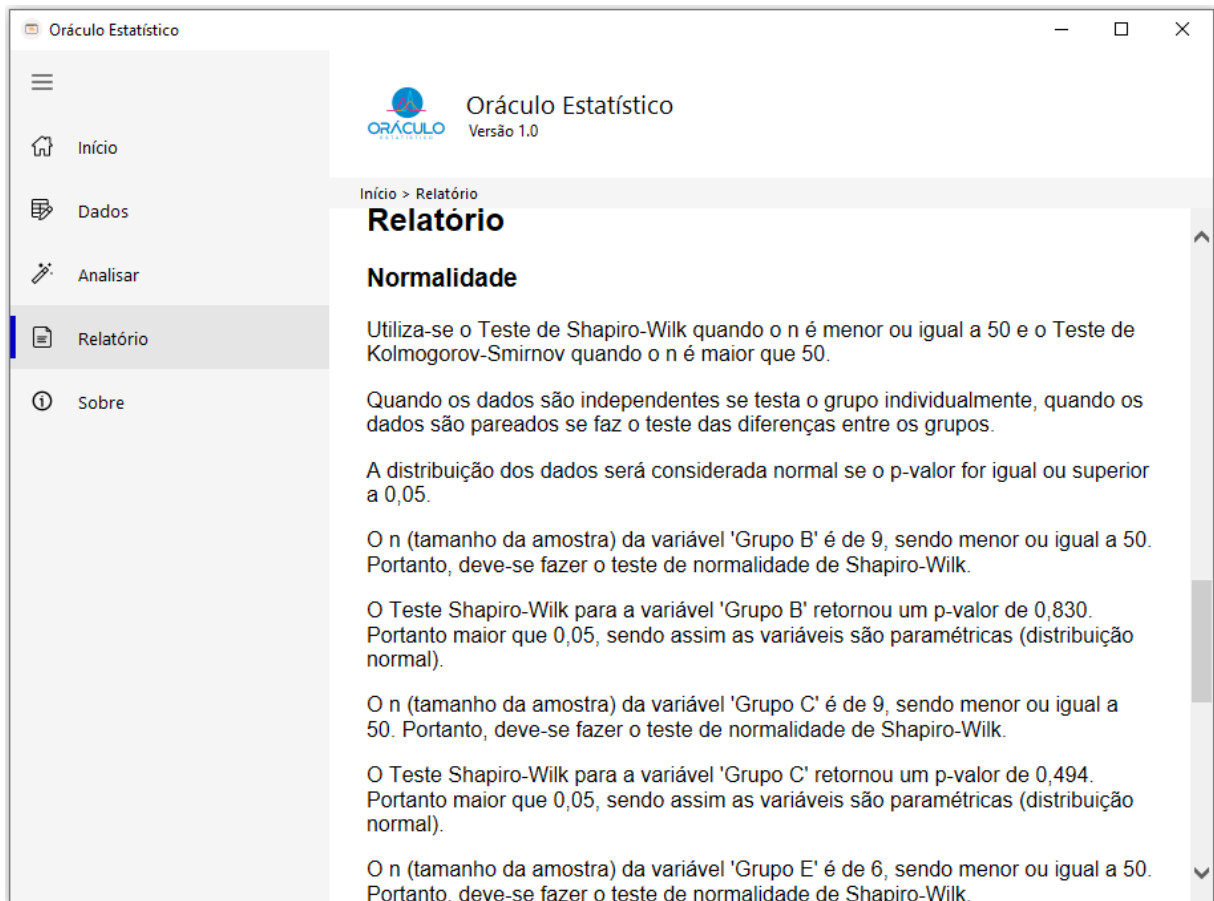


Figura 26 – Resultado da Análise de três variáveis (Parte 4).
Fonte: O próprio autor.

Na Figura 27 apresenta a continuação do relatório, contendo: os resultados e interpretação do teste de normalidade, do teste de igualdade de variância e do teste de hipótese propriamente dito.

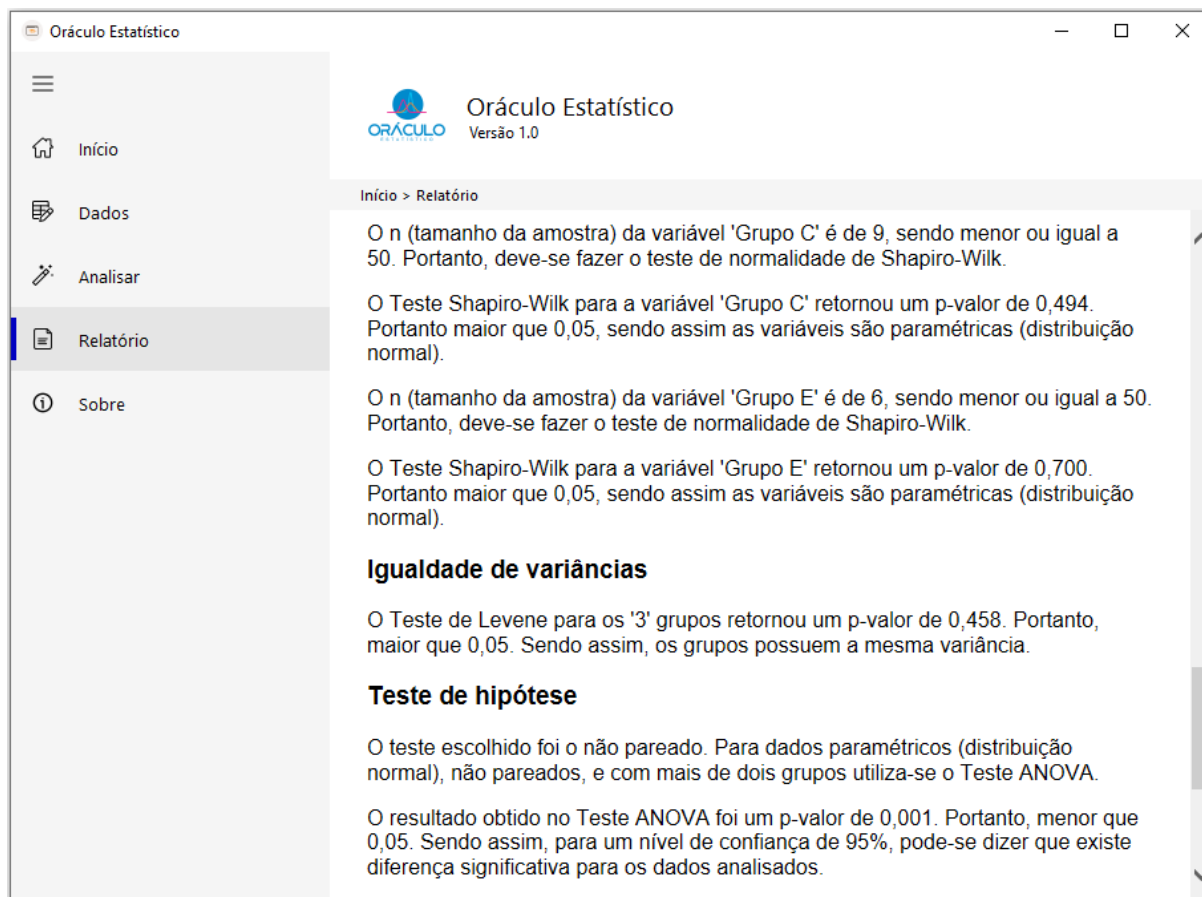


Figura 27 – Resultado da Análise de três variáveis (Parte 5).
Fonte: O próprio autor.

Na Figura 28 apresenta o relatório completo do teste de hipótese e do teste de Bonferroni.

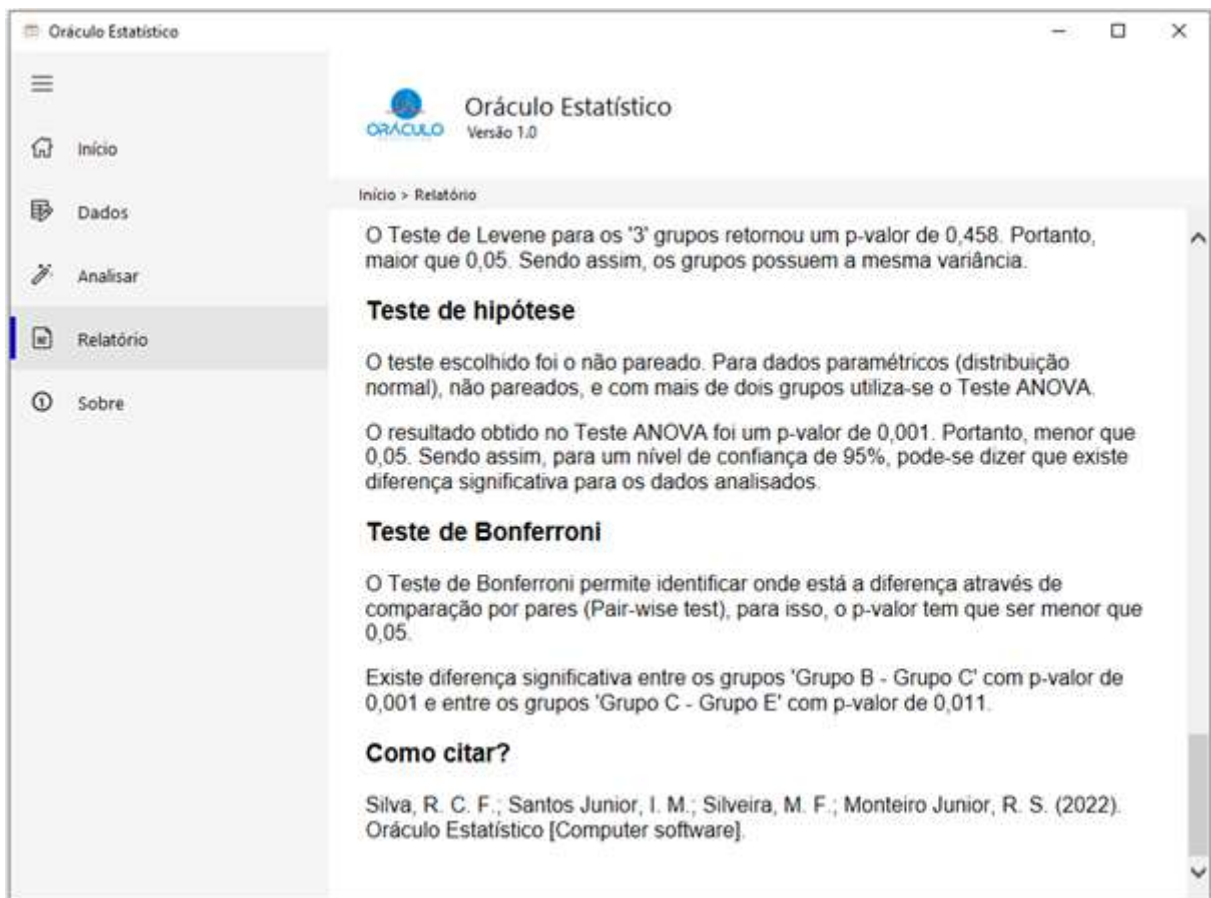


Figura 28 – Resultado da Análise de três variáveis (Parte 6).

Fonte: O próprio autor.

O usuário pode navegar também utilizando a aba lateral. Um dos itens dessas abas é o “Sobre”, que traz informações sobre o Oráculo Estatístico, a versão do *software*, a identificação da equipe que o desenvolveu, bem como o link do currículo lattes e ORCID de cada membro dessa equipe. Também consta nessa aba o item “Como Citar?”, para auxiliar o usuário na citação do *software* (Figura 29).

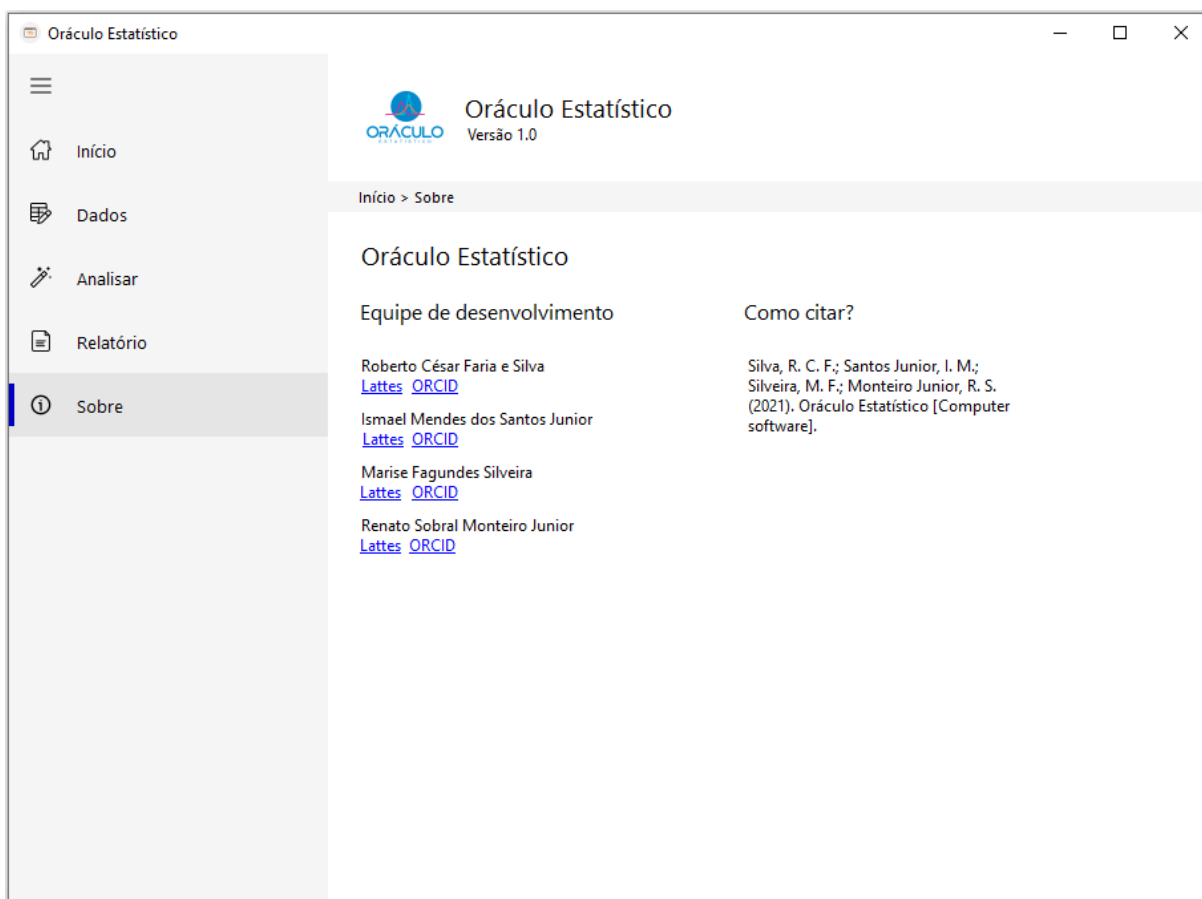


Figura 29 – Explorando a aba Sobre.
Fonte: O próprio autor.

Relatório do Oráculo Estatístico

O resultado da análise estatística pode ser copiado para o Word, bastando selecionar o que se deseja copiar na tela do *software*, pressionar as teclas Ctrl + C, e no documento do Word, pressionar as teclas Ctrl + v.

O resultado das análises apresenta a estatística descritiva e todos os testes realizados em forma de tabela, além disso, traz um relatório explicando cada teste, os resultados encontrados e sua interpretação.

Validação do oráculo estatístico

Para verificar a acurácia do *software* foram realizados testes de validação. Foram feitos testes com bases de dados diferentes. Nesses testes as análises foram feitas no PASW e R, além de alguns cálculos realizados no Excel. Em todos os testes o Oráculo Estatístico apresentou o mesmo resultado dos programas supracitados, o que valida o *software* e seus resultados (Tabela 2).

Foram usadas várias bases de dados dos mais diversos tamanhos, com destaque ao índice mineiro de responsabilidade social da fundação João Pinheiro (52), uma base extensa que permitiu constatar que o *software* consegue suportar bases grandes para a realização de seus cálculos.

Tabela 2 – Validação de testes de hipótese do software Oráculo Estatístico com o R e o PASW.

Variável	N	Média	Desvio padrão	Teste	Oráculo Estatístico (p-valor)	R (p-valor)	PASW (p-valor)
Gastos Per Capita 2016 [#]	853	646,08	277,24	Wilcoxon	<0,01	<0,01	<0,01
Gastos Per Capita 2017 [#]	853	687,43	303,41				
Mortalidade 2016 [#]	853	6,93	1,65	Teste T pareado	0,115	0,115	0,115
Mortalidade 2017 [#]	853	7,01	1,68				
Exergame DELTA pré repouso ^{*k}	10	0,23	0,14	Wilcoxon	0,131	0,131	0,131
Exergame DELTA pós repouso ^{*k}	10	0,18	0,11				
Exergame Theta pré repouso ^{*k}	10	0,27	0,05	Teste T pareado	0,399	0,399	0,399
Exergame Theta pos repouso ^{*k}	10	0,29	0,04				

Dados do Índice Mineiro de Responsabilidade Social, disponível em:

<<http://imrs.fjp.mg.gov.br/>>; ^φDados do Pesquisador T.B.X.R. (grupo GENESEs), ^{*k} Dados do Pesquisador O.T.S.(grupo GENESEs).

Para exemplificar um dos testes realizados, foram escolhidos três grupos de dados independentes (Tabela 3).

Tabela 3 – Dados utilizados para a análise de validação do Oráculo Estatístico.

Grupo B	Grupo C	Grupo E
48	45	49
49	45	46
49	46	49
50	47	50
50	47	52
50	47	51

51	48
51	48
52	49

Fonte: O próprio autor

De acordo com o passo a passo mostrado anteriormente (Fluxograma 1), o Oráculo Estatístico gerou o relatório demonstrado anteriormente (Figuras 16 a 21) com todos os cálculos e explicação. Para os outros *softwares* foram seguidas as etapas demonstradas no Fluxograma 1, de acordo com as particularidades de cada programa.

O primeiro passo foi calcular a estatística descritiva desses grupos, esse cálculo foi realizado com a ajuda do Excel (Tabela 4), e os resultados foram comparados com os resultados obtidos no Oráculo Estatístico (Tabela 5). Pode-se verificar que os resultados encontrados foram os mesmos.

Tabela 4 - Estatística descritiva realizada no Excel.

Grupo	Grupo B	Grupo C	Grupo E
Contagem	9,000	9,000	6,000
Média	50,000	46,889	49,500
Mediana	50,000	47,000	49,500
Moda	50,000	47,000	49,000
Desvio Padrão	1,225	1,364	2,074
Variância	1,500	1,861	4,300
Erro padrão	0,408	0,455	0,847
Amplitude	4,000	4,000	6,000
Mínimo	48,000	45,000	46,000
Máximo	52,000	49,000	52,000
Curtose	-0,286	-0,782	1,109
Assimetria	0,000	-0,128	-0,807
Soma	450,000	422,000	297,000

Fonte: O próprio autor.

Tabela 5 - Estatística descritiva realizada no *software* Oráculo Estatístico.

Variável	Grupo B	Grupo C	Grupo E
Contagem	9,000	9,000	6,000
Média	50,000	46,889	49,500
Mediana	50,000	47,000	49,500
Moda	50,000	47,000	49,000
Desvio padrão	1,225	1,364	2,074
Variância da amostra	1,500	1,861	4,300
Erro padrão	0,408	0,455	0,847
Amplitude	4,000	4,000	6,000
Mínimo	48,000	45,000	46,000
Máximo	52,000	49,000	52,000
Curtose	-0,286	-0,782	1,109
Assimetria	0,000	-0,128	-0,807
Soma dos valores	450,000	422,000	297,000

Fonte: O próprio autor.

Ao compararmos os resultados obtidos na Tabela 4 com a Tabela 5 pode-se constatar que os resultados encontrados foram os mesmos. Eles validam o Oráculo Estatístico em relação a estatística descritiva.

O teste de normalidade entre os três grupos foi feito no Oráculo Estatístico (Tabela 6) e no R (Figura 30). Pode-se observar que os resultados encontrados foram os mesmos, o que valida a acurácia do Oráculo Estatístico em relação ao teste de normalidade. Entretanto no R o leitor tem que interpretar o resultado, o Oráculo Estatístico já está inserido na tabela 5 a interpretação deste resultado.

Tabela 5 - Teste de normalidade realizado no *software* Oráculo Estatístico.

Grupo	Tamanho	Teste	p-valor	Distribuição
Grupo B	9	Shapiro-Wilk	0,830	Normal
Grupo C	9	Shapiro-Wilk	0,494	Normal
Grupo E	6	Shapiro-Wilk	0,700	Normal

Fonte: O próprio autor

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins
Console Terminal x Jobs x
~/
> shapiro.test(B)
      shapiro-wilk normality test
data:  B
w = 0.96307, p-value = 0.8299
> shapiro.test(C)
      shapiro-wilk normality test
data:  C
w = 0.93136, p-value = 0.4944
> shapiro.test(E)
      shapiro-wilk normality test
data:  E
w = 0.945, p-value = 0.6997
    
```

Figura 30 – Teste de normalidade no R.

Fonte: O próprio autor

O teste de igualdade de variância (homogeneidade) utilizado foi o Teste de Levene, o resultado encontrado no Oráculo Estatístico (Tabela 7) é o mesmo dos resultados encontrados no R (Figura 31) e PASW (Tabela 8), isso valida a precisão do Oráculo Estatístico em relação ao teste de igualdade de variância.

Tabela 7 – Teste de igualdade de variância realizado no *software* Oráculo Estatístico.

Teste	p-valor
Teste de Levene	0,458

Fonte: O próprio autor

```

> leveneTest(value~variable,dataset,center=mean)
Levene's Test for Homogeneity of Variance (center = mean)
      Df F value Pr(>F)
group  2  0.8102 0.4582
      21
    
```

Figura 31 – Teste de igualdade de variância no R.

Fonte: O próprio autor

Tabela 8 – Teste de igualdade de variância no PASW.

Estatística de Levene	gl1	gl2	Sig.
.810	2	21	.458

Fonte: O próprio autor

Após verificado que todos os três grupos independentes apresentavam normalidade e a variância apresentava homogeneidade, condições para se utilizar o Teste ANOVA, foi realizado o cálculo deste. O cálculo foi realizado no Oráculo Estatístico (Tabela 9), no Excel (Tabela 10), no R (Figura 32) e no PASW (Tabela 11). Os resultados obtidos em todos os *softwares* foram iguais, o que demonstra a exatidão do Oráculo Estatístico em relação ao teste ANOVA.

Tabela 9 - ANOVA no Oráculo Estatístico.

Teste	p-valor	Estatística de teste	Valor crítico
-------	---------	----------------------	---------------

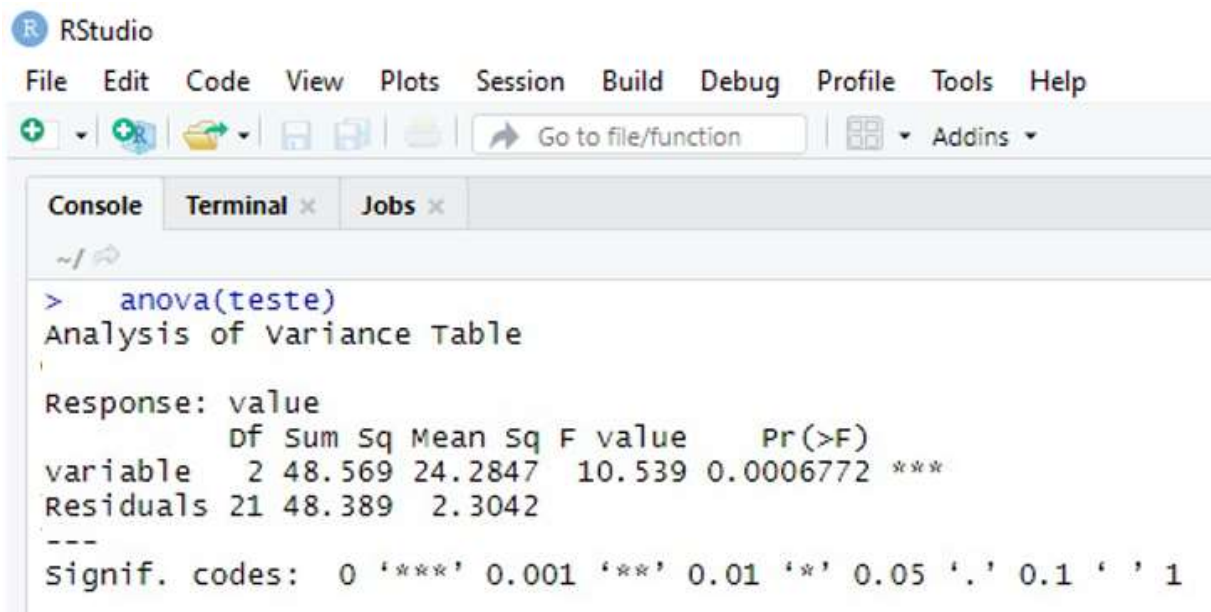
ANOVA	0,001	10,539	3,467
-------	-------	--------	-------

Fonte: O próprio autor.

Tabela 10 - ANOVA no Excel.

ANOVA (um Fator)						
fonte da variação	graus de liberdade	soma dos quadrados	quadrados médios	F calculado	p-valor	F Crítico (Tabelado)
Entre Grupos	2	48,569	24,285	10,539	0,001	3,467
Dentro dos Grupos	21	48,389	2,304			
Total	23	96,958				

Fonte: O próprio autor.



```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
~/
> anova(teste)
Analysis of Variance Table

Response: value
      Df Sum Sq Mean Sq F value    Pr(>F)
variable  2  48.569  24.2847  10.539 0.0006772 ***
Residuals 21  48.389   2.3042
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Figura 32 – ANOVA no R.

Fonte: O próprio autor.

Tabela 11 – ANOVA no PASW.

	Soma dos Quadrados	gl	Quadrados Médio	F	Sig.
Entre Grupos	48.569	2	24.285	10.539	.001
Nos Grupos	48.389	21	2.304		
Total	96.958	23			

Fonte: O próprio autor.

Como foi constatada uma diferença estatística significativa no teste ANOVA, deve-se fazer um teste post hoc para se saber em que grupos está(ão) essa(s) diferença(s). Nesse caso o teste utilizado foi o de Bonferroni. Esse teste foi realizado no Oráculo estatístico (Tabela 12), no R (Figura 33), no PASW (Tabela 13) e no Excel (Tabela 14). Pode-se observar o resultado do teste de Bonferroni no Oráculo Estatístico é igual àqueles encontrados nos outros *softwares* utilizados, o que permite validar a exatidão do Oráculo Estatístico nesse teste.

Tabela 12 - Teste de Bonferroni no Oráculo Estatístico.

Diferenças	p-valor	Estatística de teste	Valor crítico
Grupo B - Grupo C	0,001	3,111	2,601
Grupo B - Grupo E	1,000	0,500	2,601
Grupo C - Grupo E	0,011	-2,611	2,601

Fonte: O próprio autor

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Console Terminal x Jobs x
~/
> bonfCI(data2$value, data2$variable, conf.level = 0.95)
95% Bonferroni confidence intervals

      Diff      Lower      Upper Decision Adj. p-value
muB-muC 3.11111 1.24964 4.97258 Reject H0 0.000849
muB-muE 0.5 -1.58119 2.58119 FTR H0 1
muC-muE -2.61111 -4.6923 -0.52993 Reject H0 0.011133
> pairw.anova(data2$value, data2$variable, method = "bonf")

```

Figura 33 – ANOVA no R.
 Fonte: O próprio autor.

Tabela 13 - Teste de Bonferroni no PASW.

(I)var	(J)var	Diferença			Intervalo de Confiança 95%	
		média (I-J)	Erro Padrão	Sig.	Limite Inferior	Limite Superior
1.00	2.00	3.111*	.716	.001	1.25	4.97
	3.00	.500	.800	1.000	-1.58	2.58
2.00	1.00	-3.111*	.716	.001	-4.97	-1.25
	3.00	-2.611*	.800	.011	-4.69	-.53
3.00	1.00	-.500	.800	1.000	-2.58	1.58
	2.00	2.611*	.800	.011	.53	4.69

* A diferença média é significativa no nível 0.05.

Fonte: O próprio autor.

Tabela 14 - Teste de Bonferroni no Excel.

Diferenças	p-valor	Estatística de Teste	Valor Crítico
Grupo B - Grupo C	0,001	3,111	2,601
Grupo B - Grupo E	1,000	0,500	2,601
Grupo C - Grupo E	0,011	2,611	2,601

Fonte: O próprio autor

Pode-se verificar que o Oráculo estatístico se mostrou preciso em todos os testes. É importante mencionar que, além dos testes demonstrados, foram feitos vários outros com bases de dados diferentes, e em todos esses o Oráculo Estatístico demonstrou precisão nos seus resultados, o que permitiu sua validação.

Conclusão

O Oráculo Estatístico se mostrou uma ferramenta válida, e em consonância com os programas comercialmente encontrados, atendendo problemas que envolvem teste de hipótese. Ele pode ser usado para a análise inferencial em problemas de pesquisa que envolvem comparação de médias (testes paramétricos) e seus respectivos testes não paramétricos. O *software* poderá auxiliar pesquisadores pois traz como novidade e originalidade os resultados automáticos e com relatórios explicativos que contribuem para o aprendizado. Além disso, apresenta-se como vantajoso entre os demais por bloquear ações errôneas nas escolhas de testes. Como perspectivas futuras, estratégias estão sendo criadas para disponibilizar o *software* com o seu código para que demais usuários possam fazer melhorias contínuas de acordo com demandas, ou adaptações em métodos, isso paralelo ao incremento das versões por meio do proponente criador do *software*. Ainda, objetiva-se a adição de novos testes, bem como a adição de funções como a criação de gráficos em versões posteriores.

3.3 Produto 3 – Artigo:

Oráculo estatístico: uma nova ferramenta de análise

SILVA, Roberto César Faria e¹

SANTOS JUNIOR, Ismael Mendes dos²

RODRIGUES, Vinicius Dias³

SILVEIRA, Marise Fagundes⁴

MONTEIRO-JUNIOR, Renato Sobral⁵

RESUMO

O uso da inferência estatística em trabalhos científicos é bastante comum e necessário. Entretanto, existem dificuldades por parte de alguns pesquisadores quanto ao uso das técnicas propostas por este ramo da estatística. Apesar de existirem muitos *softwares* no mercado, os mais populares apresentam interfaces gráficas pouco intuitivas. Para resolver este problema, foi desenvolvido o Oráculo Estatístico. Esta pesquisa tem por objetivo verificar se o *software* Oráculo Estatístico é uma ferramenta eficiente de análise estatística. A metodologia utilizada contou com uma amostra de 11 pessoas do grupo GENESEs, com pouco conhecimento de estatística, que primeiramente assistiram à uma aula de estatística e depois foram divididos em dois grupos para fazer uma atividade (Questionário I). O Grupo I utilizou o Oráculo Estatístico e posteriormente o PSPP e o Grupo II utilizou os mesmos em ordem inversa. Ao final da atividade, foi aplicado o Questionário II para avaliar a interface do usuário com o sistema. Como resultado, percebeu-se que o Oráculo Estatístico apresentou os mesmos resultados do R e PSPP e quando comparado ao PSPP em relação a rapidez e facilidade de uso o Oráculo Estatístico se mostrou superior. Pode-se concluir que o Oráculo Estatístico é uma eficiente ferramenta estatística.

Palavras-Chave: Software; Estatística; Análise; Intuitividade; Usabilidade.

¹ Discente de doutorado do programa de pós graduação em ciências da saúde.

² Professor Mestre do Instituto Federal do Norte de Minas.

³ Professor Doutor do departamento de Educação Física - Unimontes

⁴ Professora Doutora do Programa de Pós Graduação Ciências da Saude

⁵ Professor Doutor do Programa de Pós Graduação Ciências da Saude

1 INTRODUÇÃO

Devido à dificuldade de acesso, custos e até mesmo riscos para se analisar uma população, muitos pesquisadores utilizam informações obtidas através de amostras para obter inferências para a população^{1,2}. Para Steverson³ qualquer característica populacional pode ser estimada a partir de uma amostra aleatória utilizando-se de procedimentos estatísticos.

Muitos trabalhos acadêmicos requerem testes estatísticos, entretanto a estatística não é de domínio de muitos pesquisadores. Tais testes, embora de essencial utilidade, podem desmotivar pesquisadores. Apesar de existirem muitos *softwares* que fazem análises estatísticas, suas interfaces nem sempre são simples e, além disso, esses programas exigem conhecimento prévio do pesquisador para utilizá-los e para a interpretação dos resultados alcançados. Nesse contexto, há necessidade do desenvolvimento de uma ferramenta mais atraente, prática e intuitiva que pode ser utilizada por pessoas que tenham pouco, ou até mesmo nenhum conhecimento estatístico, especialmente se considerarmos que vivemos em uma era de inteligência artificial, em que muitos problemas do dia a dia são resolvidos por interfaces digitais inteligentes⁶.

O software Oráculo Estatístico foi desenvolvido com o propósito de resolver estes problemas, sendo uma ferramenta de fácil utilização para a resolução de questões de investigação científica que necessitam da estatística. Acreditamos assim que a etapa de análise de dados de uma pesquisa pode deixar de ser um enigma para alguns pesquisadores. Portanto, o objetivo principal deste estudo foi verificar a efetividade do *software* Oráculo Estatístico como ferramenta de análise. Para tal, contou-se com os seguintes objetivos específicos: i) averiguar se o *software* analisa e fornece os resultados corretamente, quando comparado a outro *software* já conhecido (PSPP), ii) comparar o tempo de utilização para resolver uma análise em relação ao programa de estatística PSPP, e iii) analisar a facilidade de sua utilização.

2 MÉTODO

Participaram do estudo 11 estudantes e membros do Grupo de Estudos e Pesquisas em Neurociência, Exercício, Saúde e Esporte (GENESEs), vinculado à Universidade Estadual de Montes Claros (UNIMONTES). Os indivíduos foram escolhidos por terem experiência em pesquisa, porém, com pouco conhecimento em técnicas e *softwares* estatísticos. O trabalho foi aprovado pelo Comitê de Ética em Pesquisa da Universidade Estadual de Montes Claros sob o

parecer de nº 3.202.808, ano de 2019, CAAE:09252119.0.0000.5146.

2.1 Procedimentos

Os participantes assistiram a uma aula de estatística de 60 minutos, na qual foram apresentados conceitos básicos, sobretudo os relacionados aos testes paramétricos e não paramétricos para amostras dependentes e independentes. Em seguida, a amostra foi dividida aleatoriamente em dois grupos (Grupo I, n=5; e Grupo II, n=6) utilizando o site <http://randomization.com>. Foram aplicados dois questionários a cada indivíduo de cada grupo. Um dos questionários continha um problema fictício de pesquisa e um banco de dados quantitativos. Tal problema deveria ser resolvido utilizando os *softwares* estatísticos de forma individual. O outro questionário apresentava perguntas sobre a utilização de ambos os *softwares*.

Os indivíduos do Grupo I utilizaram num primeiro momento o Oráculo Estatístico e, num segundo momento, o software PSPP. Os participantes do Grupo II fizeram a ordem inversa. A escolha do PSPP se deve à sua semelhança com o amplamente conhecido e utilizado SPSS, além do fato de ser um *software* de acesso livre.

O *software* Oráculo Estatístico foi desenvolvido pelos autores deste trabalho, codificado em linguagem de programação C#⁷. O *software* tem como intuito buscar melhor interação com o usuário. Desta forma, permite realizar os testes estatísticos de forma fácil e rápida, além de fornecer relatórios explicativos para facilitar a interpretação dos resultados pelo usuário, principalmente daquele que tem pouco conhecimento estatístico.

2.2 Questionários

Para avaliar a eficácia do Oráculo Estatístico, foram aplicados dois questionários. O Questionário 1 tinha um problema a ser resolvido usando os *softwares* Oráculo Estatístico e PSPP, além de um campo para a mensuração do tempo utilizado para realizar o teste e um campo de resposta para verificar se o usuário conseguiu resolver o problema (Apêndice I).

O segundo questionário foi estruturado com perguntas a respeito da facilidade de uso dos *softwares* em questão. Nesse sentido, foram elaboradas doze questões derivadas de alguns fatores de usabilidade de um *software*, propostos por Nielsen⁸: a facilidade de aprendizado

(*learnability*), a eficiência (*efficiency*), a segurança no uso (*safety*) e a satisfação do usuário (*satisfaction*). Para cada questão os respondentes deram uma nota que variava de 1 (um) a 10 (dez) pontos, onde a nota 10 (dez) é o conceito máximo para a questão avaliada.

O primeiro fator, a facilidade de aprendizado, refere-se ao tempo e esforço necessários para que o usuário aprenda a utilizar o sistema. Quanto ao segundo fator, a eficiência, trata-se do tempo necessário para a conclusão de uma atividade por meio do *software*, também entendida como produtividade. A segurança no uso refere-se a como o *software* protege o usuário de condições desfavoráveis ou até mesmo perigosas. Por fim, a satisfação do usuário trata de uma avaliação subjetiva de cada pessoa em relação à sua interatividade com o *software*, resultado do uso do sistema sobre as suas emoções e sentimentos.

2.3 Análise dos dados

A normalidade foi verificada usando os testes de Shapiro-Wilk. De acordo com o contexto deste estudo, utilizaram-se os testes para amostras dependentes e independentes, considerando os diferentes grupos I e II (independentes) e as diferentes condições para um mesmo grupo (Oráculo Estatístico Vs. PSPP). No caso do problema aplicado através do Questionário I, observou-se que os dados eram não paramétricos e pareados, portanto, o teste utilizado foi o de Wilcoxon⁹.

3 RESULTADOS

3.1 Software Oráculo Estatístico

Com o objetivo de verificar se o *software* Oráculo Estatístico seria efetivo na resolução de problemas estatísticos, foram utilizados vários bancos de dados, os quais foram testados utilizando os *softwares* Oráculo Estatístico, PSPP e R.

Dentre os dados utilizou-se primeiramente o problema estatístico do Questionário I. Ele foi resolvido pelos pesquisadores antes da sua aplicação com os participantes do estudo, e teve seu resultado comparado com os resultados obtidos nos *softwares* PSPP e R. Em todos os *softwares* os resultados observados foram os mesmos (Quadro 1).

Também, buscou-se utilizar dados de duas pesquisas de membros do próprio grupo GENESEs, e dados do Instituto Mineiro de Responsabilidade Social (IMRS), para uma nova rodada de testes de validação, resultando em 32 no total. Esses dados também podem ser visualizados no Quadro 1.

Quadro 1. Comparação de Resultados entre os *Softwares*.

Variável	N	Média	Desvio padrão	Teste	Oráculo Estatístico (p-valor)	R (p-valor)	PSPP (p-valor)
Altura do Grupo Controle*	20	47,55	2,46	Teste T não pareado	0,342	0,342	0,342
Altura do Grupo de Comparação*	20	48,30	2,47				
Gastos Per Capita 2016 [#]	853	646,08	277,24	Wilcoxon	<0,01	<0,01	<0,01
Gastos Per Capita 2017 [#]	853	687,43	303,41				
Mortalidade 2016 [#]	853	6,93	1,65	Teste T pareado	0,115	0,115	0,115
Mortalidade 2017 [#]	853	7,01	1,68				
Exergame DELTA pré repouso ^{*k}	10	0,23	0,14	Wilcoxon	0,131	0,131	0,131
Exergame DELTA pós repouso ^{*k}	10	0,18	0,11				
Exergame Theta pré repouso ^{*k}	10	0,27	0,05	Teste T pareado	0,399	0,399	0,399
Exergame Theta pós repouso ^{*k}	10	0,29	0,04				

*Dados do Questionário 1; [#] Dados do Índice Mineiro de Responsabilidade Social, disponível em: <<http://imrs.fjp.mg.gov.br/>>; ^{*k} Dados do Pesquisador O.T.S.(grupo GENESEs).

3.2 Tempo utilizado na Análise

No questionário I foi mensurado o tempo utilizado pelo usuário para solucionar o problema estatístico. Os onze participantes da análise tinham que tentar resolver esse problema utilizando ambos os *softwares* (Oráculo Estatístico e PSPP). Ressalta-se que foi dado ao participante o direito de desistir a qualquer momento da pesquisa. Portanto, o tempo foi mensurado da hora que começou a atividade até a sua finalização ou desistência. A comparação entre os tempos gastos (em minutos) em ambos os *softwares* é demonstrada na Figura 1.

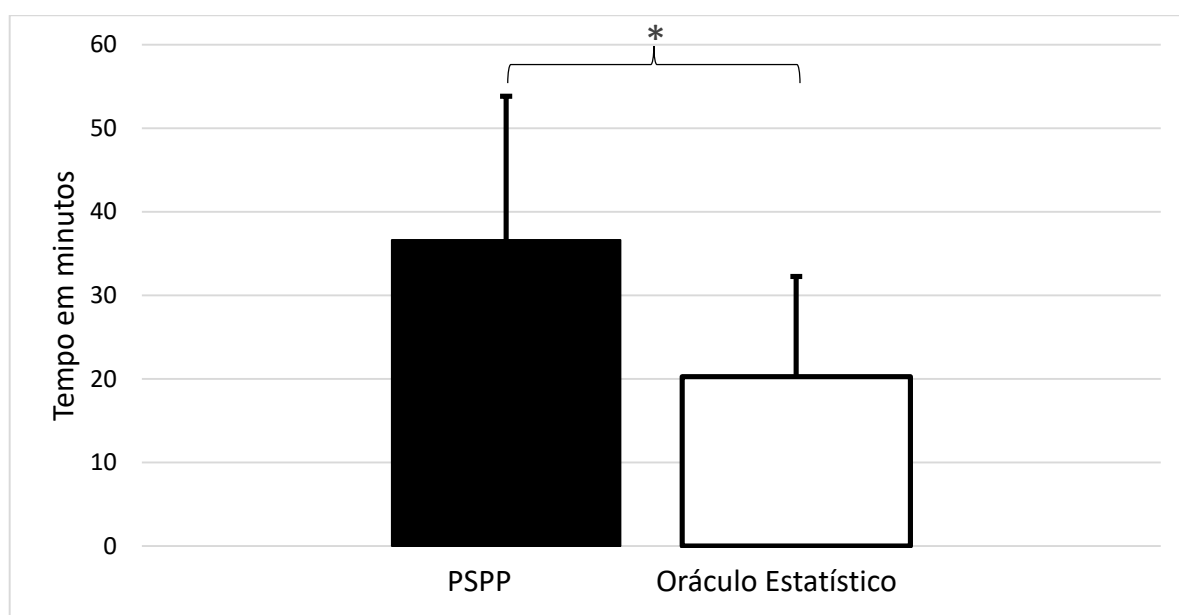


Figura 1. Comparação entre os tempos gastos. Análise teste t não pareado. Dados expressos em média \pm Desvio Padrão. Nível de confiança de 95%. * $p=0,013$.

Em relação à velocidade de análise, pode-se constatar que em média, o tempo utilizado no Oráculo Estatístico para resolver o problema proposto foi de $20,27 \pm 12$ minutos, enquanto que o tempo gasto no PSPP foi de $37,77 \pm 17,22$ minutos, de modo que o tempo do Oráculo foi 46% menor que do PSPP. O p-valor resultante da comparação foi de 0,013, com um Z de -2,491, o que demonstra existir diferença significativa entre os dois *softwares* onde o Oráculo Estatístico precisou de menos tempo para resolver o problema que o PSPP.

Como na mensuração do tempo levou-se em conta a desistência, deve-se avaliar o percentual de pessoas que conseguiram resolver o problema, conforme apresentado na Figura 2.

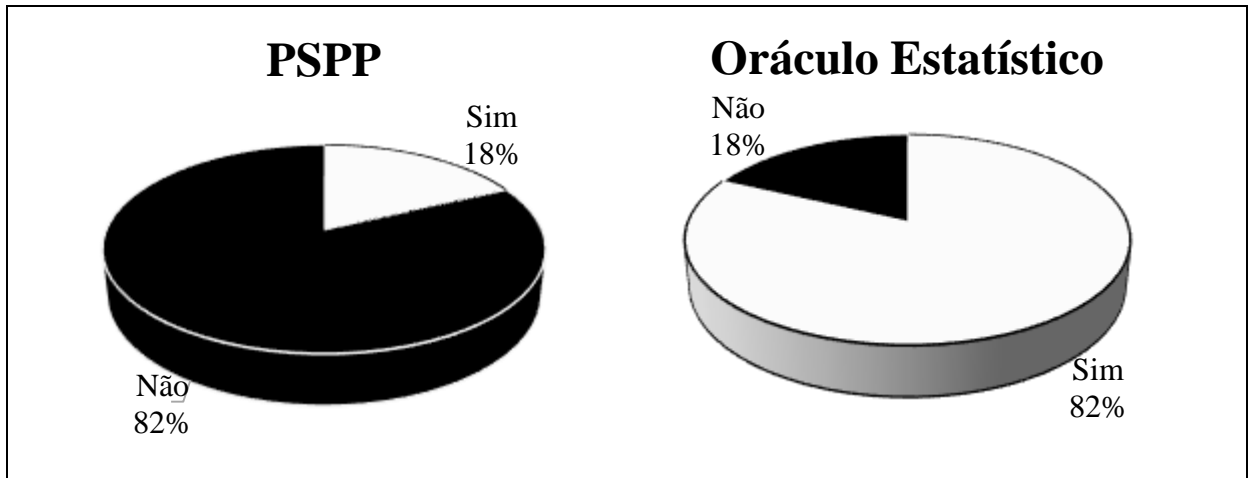


Figura 2. Percentual de Acertos entre os usuários do programas avaliados PSPP e Oráculo Estatístico.

Em relação à quantidade de acertos na resolução do problema proposto (Figura 2), no PSPP apenas 18% das pessoas conseguiram resolver o problema proposto corretamente, os outros 82% não conseguiram resolver e desistiram. Quanto ao Oráculo Estatístico o resultado foi o oposto: 82% conseguiram solucionar o problema corretamente e 18% não conseguiram resolver e acabaram desistindo.

3.3 Usabilidade dos *softwares*

Em relação à facilidade de uso foram avaliados o aprendizado, a eficiência, a segurança e a satisfação. Os resultados são apresentados no Tabela 2.

Tabela 2. Comparação dos Critérios de Usabilidade entre os softwares PSPP e Oráculo Estatístico.

Variáveis	PSPP	Oráculo	P-valor	Z	Teste
Aprendizado	3,323 (2,821)	8,121 (2,837)	<0,01	-4,507	Wilcoxon
Eficiência	4,000 (2,558)	8,036 (2,582)	<0,01	-5,369	Wilcoxon
Segurança	2,900 (2,558)	7,727 (3,495)	≤0,02	-2,325	Wilcoxon
Satisfação	4,727 (3,065)	7,727 (2,730)	<0,01	-4,112	Wilcoxon

O quesito aprendizado se refere ao tempo e ao esforço necessário para que o usuário aprenda a usar o sistema, nesse quesito o PSPP obteve uma média de 3,32 pontos e um desvio padrão de 2,82. O Oráculo Estatístico, por sua vez, obteve uma média de 8,12 e um desvio padrão de 2,83. Quando comparado através do teste de Wilcoxon pode-se perceber um p-valor menor que 0,01 e um Z de -4,507, portanto existe uma diferença significativa entre os dois *softwares* e o Oráculo Estatístico mostrou-se superior ao PSPP.

A eficiência está ligada ao tempo em que o usuário utiliza para realizar uma atividade com apoio computacional. Ela é fortemente ligada à interface com o usuário utilizada pelo sistema interativo. Quando avaliada a eficiência do PSPP, este obteve uma média de 4,000 e um desvio padrão de 2,558, já o Oráculo Estatístico obteve uma média de 8,036 e um desvio padrão de 2,582. O p-valor resultante da comparação foi <0,001, com um Z de -5,369, o que demonstra existir diferença significativa entre os dois *softwares* e novamente, o Oráculo Estatístico foi superior ao PSPP.

A segurança é ligada ao sentimento da pessoa ao usar o sistema, se ela tem confiança na hora de interagir, e utiliza o *software* sem medo de errar. Neste quesito, o PSPP obteve 2,90 de média com um desvio de 2,55 e o Oráculo Estatístico 7,72 de média com um desvio de 3,49. O p-valor menor igual a 0,02 e o Z de -2,32, o que mostra existir uma diferença significativa entre os dois *softwares*, sendo que o Oráculo Estatístico apresentou um melhor resultado.

Por fim, tem-se a satisfação, que é medida através das emoções e sentimentos das pessoas em relação ao uso do *software*, no PSPP a nota média da satisfação foi de 4,72 com um desvio de 3,06, já o Oráculo Estatístico obteve uma média de 7,72 com 2,73 de desvio padrão e o p-valor menor que 0,01 e o Z teste de -4,11. Constatou-se que o Oráculo Estatístico foi superior ao PSPP.

4 DISCUSSÃO

Este estudo comparou o *software* Oráculo Estatístico, desenvolvido para facilitar o uso da estatística por pesquisadores com pouco ou nenhum domínio do tema, com outros *softwares* já conhecidos e analisou a usabilidade do referido *software*. Verificou-se que tanto a reprodutibilidade dos resultados das análises, como a facilidade de utilização do Oráculo Estatístico foram confirmados.

A corretude funcional de um *software* é a sua capacidade em prover de forma precisa os resultados¹⁰. Nos testes estatísticos utiliza-se muito do cálculo do p-valor para análise dos dados¹¹. Devido ao intenso uso do p-valor este foi calculado em testes estatísticos para comparar dois grupos (Teste t pareado, não pareado ou independente, Mann-Whitney e Wilcoxon) utilizando-se dos *softwares* Oráculo Estatístico, R e PSPP (Quadro 1), pode-se perceber que o resultado encontrado foi o mesmo. Portanto o oráculo passou no teste de acurácia o que demonstra que o Oráculo Estatístico apresenta as respostas com a mesma exatidão dos demais *softwares* já consagrados.

A estatística, apesar de importante na área da saúde, ainda é negligenciada, mesmo nas revistas médicas mais prestigiadas são encontrados erros envolvendo a estatística¹², nas revistas de odontologia eles também são comuns¹³. Estudantes de saúde classificam a estatística como uma matéria difícil e a valorizam em uma situação um pouco melhor do que indiferença¹⁴. Os pacotes estatísticos embutidos em planilhas ou disponíveis no mercado são incapazes de mostrar que o pesquisador utiliza o teste estatístico errado¹⁵. As telas dos *softwares* devem ser organizadas de tal maneira que o usuário possa compreender o que está fazendo e capte tudo que for importante, encontre facilmente o que se precisa¹⁶. O Oráculo Estatístico foi projetado para ser um *software* autoexplicativo e que faz uso de inteligência artificial. Basta que o usuário clique no botão (analisar) que o próprio *software* faz o teste de normalidade (de acordo com o n dos dados), e de acordo com o resultado e a opção de pareado ou não pareado (que o usuário marca com um click), o software faz o teste estatístico e sua análise, o que melhora o percentual de acertos (Figura 2) e diminui o tempo necessário para solução de problemas (Figura 1).

Fala-se muito no termo usabilidade que é a capacidade do *software* ser compreendido, operado, aprendido, como exerce atração ao seu usuário⁹. Existe um consenso em que a usabilidade é a habilidade do software de permitir que seu usuário alcance seus objetivos de forma rápida e assertiva, promovendo o fácil aprendizado e altos níveis de eficiência, segurança

e satisfação^{8, 17, 18}. Em relação ao aprendizado, eficiência, segurança e satisfação o Oráculo Estatístico superou o PSPP em todos estes quesitos, demonstrando maior usabilidade.

5 CONCLUSÃO

Conclui-se que o *software* Oráculo estatístico é uma ferramenta de uso rápido, fácil e eficiente de análise estatística.

REFERÊNCIAS

1. BUSSAB, Wilton de O; MORETTIN, Pedro A. **Estatística Básica**. 5 ed. São Paulo: Saraiva, 2003.
2. VIEIRA, Sonia. **Introdução a Bioestatística**. 4 ed. Rio de Janeiro: Elsevier, 2011.
3. STEVENSON, William J. **Estatística Aplicada a Administração**. São Paulo: Harper & How do Brasil, 1981.
4. BRUNI, Adriano Leal. **Estatística Aplicada a Gestão Empresarial**. 4 ed. São Paulo: Atlas, 2017.
5. MONTEIRO JÚNIOR, Renato Sobral. **Estatística Aplicada a Educação Física**. Montes Claros: Editora Unimontes, 2017
6. RAINER JR, R. Kelly; CEGIELSKI, Casey G. **Introdução a Sistemas de Informação**. 3 ed. Rio de Janeiro, Elsevier, 2011.
7. TRIGO, Antônio; HENRIQUES, Jorge. **Aprenda a Programar com C#**. Lisboa: Silabo, 2017.
8. NIELSEN, Jakob. **Usability Engineering**. San Francisco: Morgan Kaufmann, Inc., 1993.
9. VIEIRA, Sonia. **Introdução a Bioestatística**. 4 ed. Rio de Janeiro: Elsevier, 2011
10. NBR ISO/IEC 9126-1, Engenharia de software - Qualidade de produto Parte 1: Modelo de qualidade. Disponível em: <https://aplicacoes.mds.gov.br/sagirms/simulacao/sum_executivo/pdf/fichatecnica_21.pdf>. Acessado em 05/05/2019.
11. Ronald L. Wasserstein & Nicole A. Lazar (2016): The ASA's statement on p-values: context, process, and purpose, The American Statistician, DOI:10.1080/00031305.2016.1154108. Disponível em: <<https://www.stat.ncsu.edu/people/bloomfield/courses/st370/The%20ASA%20s%20state>

- ment%20on%20p%20values%20context%20process%20and%20purpose.pdf>. Acesso em 06/05/2019.
12. MELO, Miguel. **Leitura Crítica de Artigos Médicos**. 2000, 16:471-6. Disponível em: <www.rpmgf.pt/ojs/index.php/rpmgf/article/download/9816/9554>. Acesso em 05/05/2019.
 13. Normando D, Almeida MAO, Quintão CCA. **Análise do emprego do cálculo amostral e do erro do método em pesquisas científicas publicadas na literatura ortodôntica nacional e internacional**. Dental Press J Orthod. 2011 Nov-Dec;16(6):33.e1-9. Disponível em: <https://www.researchgate.net/profile/David_Normando/publication/262758011_Analysis_of_the_use_of_sample_size_calculation_and_error_of_method_in_researches_published_in_Brazilian_and_international_orthodontic_journals/links/541bb7330cf25ebee98d9e28.pdf>. Acesso em 05/05/2019.
 14. PIMENTA, Rui; PEREIRA, Ilídio; COSTA, Elísio; MARGARIDA, Vieira. Atitudes Frente à estatística em diferentes grupos de profissionais de saúde em formação. Disponível em: <http://www.iiis.org/CDs2010/CD2010CSC/SIECI_2010/PapersPdf/XA843RM.pdf>. Acessado em 05/05/2019
 15. da CONCEIÇÃO, Mário José, Leitura crítica dos dados estatísticos em trabalhos científicos. Revista Brasileira de Cirurgia Cardiovascular/Brazilian Journal of Cardiovascular Surgery [en línea] 2008, 23 (Julio-Septiembre). Disponível em: <<http://www.redalyc.org/articulo.oa?id=398941868018>> ISSN 0102-7638. Acesso em 05/05/2019.
 16. KRUG, Steve. Não me faça pensar. Editora: Alta Books; Rio de Janeiro, 2006
 17. LOBATO, Vinícius. Qual a importância da usabilidade para o design digital? Boletim Informativo apDesign-Associação dos Profissionais em Design do Rio Grande do Sul. Ano X – nº 22 – Agosto 2004. Porto Alegre.
 18. BERGAMO, M. L; MELO, W. TUS- Tutorial de usabilidade de software. Departamento de Ciência da Computação. Universidade Católica de Brasília. DF. 2000.

Material Suplementar

Questionário 1

Hora de início: ____ : ____

Hora final: ____ : ____

Imagine a seguinte situação Hipotética: Em uma determinada cidade, um certo cientista, criou um medicamento para aumentar a altura dos recém nascidos. Para verificar a eficiência do medicamento esse cientista escolheu 40 gestantes e dividiu em 2 grupos: o grupo I (controle), que recebeu placebo, e o grupo II (comparação), que recebeu o medicamento. No dia do nascimento os bebês foram medidos e o resultado pode ser demonstrado na tabela abaixo:

Grupo I	Grupo II
43	42
43	51
44	50
48	49
50	44
51	46
51	48
46	49
49	47
50	47
47	48
49	50
45	51
46	49
49	48
47	47
48	49
50	53
47	50
48	48

De acordo com o exposto anteriormente, responda:

1) O tratamento foi eficaz? Justifique?

2) Qual foi o teste estatísticos utilizado para verificar se o tratamento foi eficaz?

Questionário 2

Esse questionário tem por objetivo avaliar sua experiência em relação ao uso do software estatístico utilizado para responder o questionário 1.

1. Software utilizado: () PSPP () Oráculo

Pergunta	Nota atribuída									
	1	2	3	4	5	6	7	8	9	10
2. Que nota você atribui à facilidade para inserir (digitar ou importar) dados no <i>software</i> ?										
3. Que nota você atribui à facilidade para analisar os dados no <i>software</i> ?										
4. Que nota você atribui à qualidade da interpretação dos resultados gerados pelo <i>software</i> ?										
5. Que nota você atribui à sequência de telas para inserir (digitar ou importar) dados no <i>software</i> ?										
6. Que nota você atribui à sequência de telas para analisar os dados no <i>software</i> ?										
7. Que nota você atribui ao tempo necessário para inserir (digitar ou importar) dados no <i>software</i> ?										
8. Que nota você atribui ao tempo necessário para analisar dados no <i>software</i> ?										

9. Que nota geral você atribui à produtividade com o uso do software?

10. Em relação às mensagens que o sistema retorna ao usuário como você as classifica?

11. Qual nota você atribui à aparência das telas?

12. Que nota você atribui à linguagem utilizada nas telas?

13. Qual seu grau de satisfação geral quanto ao uso do *software*?

14. Você gostaria de deixar alguma sugestão?

4 CONSIDERAÇÕES FINAIS

Neste trabalho foram desenvolvidas duas ferramentas para análise estatística na área de saúde. A primeira ferramenta foi o PPGCS *Sample Size*, foi verificado para atender a estimativa amostral de forma eficiente e prática atestada para validação. Dentre as limitações do PPGCS *Sample Size* está o fato deste precisar de internet para seu funcionamento e de fazer somente cálculos amostrais para estimar e comparar médias e proporções. Como sugestão para estudos futuros, sugere-se que sejam implementados novos cálculos ao *software*, tais como cálculo do intervalo de confiança. Ainda, da possibilidade de fazer o *download* para funcionamento *off-line*.

Em relação ao Oráculo Estatístico, que se mostrou válido na solução de problemas que envolvem teste de hipótese para média e seus respectivos testes não paramétricos. Ele pode ser utilizado em análises estatísticas e em problemas de pesquisa que envolvam comparação de médias (testes paramétricos) e seus respectivos testes não paramétricos. Ele se mostrou preciso, quando comparado com o R, SPSS e Excel, e apresentou como principal vantagem o relatório gerado que facilita o entendimento e a interpretação do pesquisador, o que indica contribuições pedagógicas. Ele também faz os cálculos de forma automática utilizando as respostas dos cálculos dos pré-requisitos para escolher o próximo passo do teste. Este fato permite a seleção do teste correto para solução do problema.

Em um teste de usabilidade em que foi utilizado o PSPP (Free Software Foundation) como software de comparação, pode-se constatar que quando foi utilizado o Oráculo Estatístico os usuários conseguiram resolver o problema proposto com um tempo 46% menor em relação ao PSPP e o índice de acerto com o Oráculo Estatístico foi de 82% contra 12% do PSPP, além de receber uma nota melhor nos quesitos: aprendizado, eficiência, segurança e satisfação. As principais novidades em relação aos demais softwares são a realização dos testes de forma automática. Eles ajudam a prever erros quando usados testes indevidos ou omissão de algum pré-requisito dos testes para comprovar a hipótese. Outro aspecto é o relatório explicativo que ajuda o usuário a entender e interpretar os testes utilizados.

Dentre as limitações do oráculo encontra-se na escolha da significância, que por sua vez, já está determinada ao nível de 5% ($p < 0,05$). Embora seja o mais amplamente difundido, não é

requerido em todas as áreas e delineamentos, podendo variar entre 20% a 1%. Como perspectivas futuras, sugere-se implementar outros testes/soluções ao Oráculo Estatístico, tais como testes para comparação de variáveis qualitativas, testes de regressão e correlação, entre outros. A ideia da concepção do Oráculo Estatístico é da criação um *software* colaborativo e de propriedade da Unimontes e do PPGCS, sendo assim, essa implementação pode ser feita por alunos, professores ou outros colaboradores desta instituição ou programa ou até mesmo por meio de convênio com outras instituições. Ainda, estuda-se a possibilidade de disponibilização de códigos ao se instalar o programa, para que outros usuários possam adaptar e implementar as análises demandadas pelo estudo em questão. Além disso, pretende-se realizar uma testagem do programa em uma amostra de usuários para verificação de seu funcionamento, bem como aceitação e necessidades, bem como sua usabilidade e intuitividade. Em perspectiva, a criação em outros idiomas para sua ampla divulgação.

Por fim, espera-se que as duas ferramentas para análise estatística sejam muito úteis na área de ciências da saúde, bem como em outras áreas, e que possam contribuir para as pesquisas que necessitem destes cálculos estatísticos, auxiliando os pesquisadores em seus trabalhos, sobretudo aqueles da iniciação científica, estudantes de graduação e de mestrado, resultando em trabalhos com maior qualidade e rigor.

REFERÊNCIAS

1. Memória JMP. Breve história da estatística. 2004;
2. Lopes AA. Medicina baseada em evidências: a arte de aplicar o conhecimento científico na prática clínica. *Revista da Associação Médica Brasileira*. 2000;46:285–8.
3. Paes ÂT. Itens essenciais em bioestatística. *Arquivos brasileiros de cardiologia*. 1998;71:575–80.
4. Celik Y. *The Science of Biostatistics in Decision Making within Health Sciences*.
5. Atallah AN. Medicina baseada em evidências. *Diagn Tratamento*. 2018;23(2):43–4.
6. Agarwal BL. *Basic statistics*. New Age International; 2006.
7. Ben-Shlomo Y, Brookes S, Hickman M. *Epidemiology, evidence-based medicine and public health*. John Wiley & Sons; 2013.
8. Bruni AL. *Estatística aplicada à gestão empresarial*. Editora Atlas SA; 2013.
9. Lee KJ, Moreno-Betancur M, Kasza J, Marschner IC, Barnett AG, Carlin JB. Biostatistics: a fundamental discipline at the core of modern health data science. *The Medical Journal of Australia*. 2019;211(10):444.
10. Oster RA, Enders FT. The importance of statistical competencies for medical research learners. *Journal of Statistics Education*. 2018;26(2):137–42.
11. Freitas EA de. *Estatística aplicada*. 2018;
12. Ignácio SA. Importância da estatística para o processo de conhecimento e tomada de decisão. *Revista Paranaense de Desenvolvimento-RPD*. 2010;(118):175–92.
13. Field A. *Descobrimo a estatística usando o SPSS-5*. Penso Editora; 2009.
14. Hair JF, Anderson RE, Tatham RL, Black WC. *Análise multivariada de dados*. Bookman editora; 2005.
15. Gupta NK. Advancements in cloud computing software testing research. Em: 2018 4th International Conference on Recent Advances in Information Technology (RAIT). IEEE; 2018. p. 1–6.
16. Bevan N, Claridge N, Petrie H. Tenuta: simplified guidance for usability and accessibility. Em: *Proceedings of HCI International*. sn; 2005.
17. Strasak AM, Zaman Q, Pfeiffer KP, Göbel G, Ulmer H. Statistical errors in medical research-a review of common pitfalls. *Swiss medical weekly*. 2007;137(0304).
18. Aycaguer LS. Frequent methodological errors in clinical research. *Medicina Intensiva (English Edition)*. 2018;42(9):541–6.

19. Slutsky DJ. Statistical errors in clinical studies. *Journal of wrist surgery*. 2013;2(04):285–7.
20. Yan F, Robert M, Li Y. Statistical methods and common problems in medical or biomedical science research. *International journal of physiology, pathophysiology and pharmacology*. 2017;9(5):157.
21. Thiese MS, Arnold ZC, Walker SD. The misuse and abuse of statistics in biomedical research. *Biochemia medica*. 2015;25(1):5–11.
22. Cassidy LD. Basic concepts of statistical analysis for surgical research. *Journal of Surgical Research*. 2005;128(2):199–206.
23. Greenland S, Senn SJ, Rothman KJ, Carlin JB, Poole C, Goodman SN, et al. Statistical tests, P values, confidence intervals, and power: a guide to misinterpretations. *European journal of epidemiology*. 2016;31(4):337–50.
24. Woolson RF, Clarke WR. *Statistical methods for the analysis of biomedical data*. John Wiley & Sons; 2011.
25. Hollander M, Wolfe DA, Chicken E. *Nonparametric statistical methods*. John Wiley & Sons; 2013.
26. Kim JS, Kim DK, Hong SJ. Assessment of errors and misused statistics in dental research. *International dental journal*. 2011;61(3):163–7.
27. Madadzadeh F, Asar ME, Hosseini M. Common statistical mistakes in descriptive statistics reports of normal and non-normal variables in biomedical sciences research. *Iranian journal of public health*. 2015;44(11):1557–8.
28. Hayat MJ. Statistics education in the health sciences. Vol. 26, *Journal of Statistics Education*. Taylor & Francis; 2018. p. 135–6.
29. Torgerson DJ, Miles JN. Simple sample size calculation. *Journal of evaluation in clinical practice*. 2007;13(6).
30. Marotti J, Galhardo APM, Furuyama RJ, Pigozzo MN, Campos T de, Laganá DC. Amostragem em pesquisa clínica: tamanho da amostra. *Revista de Odontologia da Universidade Cidade de São Paulo*. 2008;20(2):186–94.
31. Miot HA. Tamanho da amostra em estudos clínicos e experimentais. *Jornal Vascular Brasileiro*. 2011;10:275–8.
32. Brito CJ, da Silva Grigoletto ME, de Toledo Nóbrega O, Córdova C. Dimensionamento de amostras e o mito dos números mágicos: ponto de vista. *Revista Andaluza de Medicina del Deporte*. 2016;9(1):29–31.
33. WILSON FILHO DPP. *Engenharia de Software*. São Paulo SP, LTC. 2003;
34. Pressman RS, Maxim BR. *Engenharia de software-9*. McGraw Hill Brasil; 2021.

35. Melendez Filho R. Prototipação de sistemas de informações: fundamentos, técnicas e metodologia. LTC; 1990.
36. Wazlawick R. Engenharia de software: conceitos e práticas. Elsevier Editora Ltda.; 2019.
37. Microsoft Corporation. Microsoft Excel. Microsoft Corporation; 2013.
38. Silveira MF, Haikal DS. Amostragem. Slide apresentado em: Aula de Bioestatística I; 2020.
39. R Core Team. R: A language and environment for statistical computing [Internet]. Vienna, Austria: R Foundation for Statistical Computing; 2019. Disponível em: <https://www.R-project.org/>
40. IBM Corp. IBM SPSS Statistics for Windows. Armonk, NY; 2013.
41. Free Software Foundation. PSPP [Internet]. 2022 [citado 14 de julho de 2022]. Disponível em: (<https://www.gnu.org/software/pspp/>)
42. StataCorp. Stata Statistical Software [Internet]. 2021. Disponível em: <https://www.stata.com/>
43. The jamovi project [Internet]. 2021. Disponível em: <https://www.jamovi.org/>
44. Laponi JC. Estatística usando excel. Elsevier Brasil; 2004.
45. Wongthongtham P, Chang E, Sommerville I. Software engineering ontology for software engineering knowledge management in multi-site software development environment. Em: 10th International Protege Conference. Stanford Centre for Biomedical Informatics Research; 2007.
46. Fettke P. Unified modeling language. Em: Encyclopedia of Information Science and Technology, First Edition. IGI Global; 2005. p. 2921–8.
47. Hejlsberg A, Torgersen M, Wiltamuth S, Golde P. The C# programming language. Pearson Education; 2008.
48. Hejlsberg A, Wiltamuth S, Golde P. C# language specification. Addison-Wesley Longman Publishing Co., Inc.; 2003.
49. microsoft. Um tour pela linguagem C# [Internet]. 2022 [citado 14 de julho de 2022]. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/>
50. Accord.NET. Machine learning made in a minute [Internet]. 2022 [citado 14 de julho de 2022]. Disponível em: <http://accord-framework.net/>
51. Troelsen AW, Olsen A. Pro C# 5.0 and the .NET 4.5 Framework. Vol. 6. Springer; 2012.

52. de Responsabilidade Social IÍM. Fundação João Pinheiro. Recuperado em <http://imrs.fjp.mg.gov.br/Perfil/PerfilMunicipal>. 2017;
53. Heller E. A psicologia das cores: como as cores afetam a emoção e a razão. Editora Olhares; 2022.
54. Pinto JC, Schwaab M. Análise de Dados Experimentais: I. Fundamentos de Estatística e Estimação de Parâmetros. Editora E-papers; 2007.
55. Arango HG. Bioestatística: teórica e computacional: com banco de dados reais. Rio de Janeiro: Guanabara Koogan, ed. 2011;3:351.
56. Hanusz Z, Tarasińska J. Normalization of the Kolmogorov–Smirnov and Shapiro–Wilk tests of normality. *Biometrical Letters*. 2015;52(2):85–93.
57. Steel RGD, Torrie JH. Principles and procedures of statistics, a biometrical approach. McGraw-Hill Kogakusha, Ltd.; 1980.
58. Siqueira AL, Tibúrcio JD. Estatística na área de saúde: conceitos, metodologia, aplicações e prática computacional. Em: Estatística na área de saúde: conceitos, metodologia, aplicações e prática computacional. 2011. p. 520–520.
59. Zaiontz C. Real statistics using Excel. 2015.
60. Sommerville I. Engenharia de software, 9a. São Palo, SP, Brasil. 2011;63.
61. Krug S. Não me faça pensar: atualizado. Uma Abordagem de Bom Senso à. 2014;
62. Krug S. Não me faça pensar!: uma abordagem de bom senso à usabilidade na web. Alta Books; 2006.

ANEXOS

ANEXO A – Certificado do Registro do Oráculo Estatístico Junto ao INPI.



REPÚBLICA FEDERATIVA DO BRASIL
MINISTÉRIO DA ECONOMIA
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL
DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS INTEGRADOS

Certificado de Registro de Programa de Computador

Processo Nº: **BR512022001978-7**

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1º de janeiro subsequente à data de 27/06/2022, em conformidade com o §2º, art. 2º da Lei 9.609, de 19 de Fevereiro de 1998.

Título: Oráculo Estatístico

Data de publicação: 27/06/2022

Data de criação: 27/06/2022

Titular(es): UNIVERSIDADE ESTADUAL DE MONTES CLAROS - UNIMONTES

Autor(es): MARISE FAGUNDES SILVEIRA; ISMAEL MENDES DOS SANTOS JUNIOR; ROBERTO CÉSAR FARIA E SILVA; RENATO SOBRAL MONTEIRO JUNIOR

Linguagem: C#

Campo de aplicação: MT-06

Tipo de programa: TC-01

Algoritmo hash: SHA-512

Resumo digital hash:
c986805a2185435177b71dbdec7ae060bac1a4267d0a55e0a8865e22d421bcbaf0c5c1c6adb5251ed320624336b4498
87eb95cb497dcb00a47268ecb14733a8

Expedido em: 02/08/2022

Aprovado por:
Joelson Gomes Pequeno
Chefe Substituto da DIPTO - PORTARIA/INPI/DIRPA Nº 02, DE 10 DE FEVEREIRO DE 2021



ANEXO B – Certificado do Registro do PPGCS Sample Size Junto ao INPI.



REPÚBLICA FEDERATIVA DO BRASIL
MINISTÉRIO DA ECONOMIA
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL
DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS INTEGRADOS

Certificado de Registro de Programa de Computador

Processo Nº: **BR512022001979-5**

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1º de janeiro subsequente à data de 06/06/2022, em conformidade com o §2º, art. 2º da Lei 9.609, de 19 de Fevereiro de 1998.

Título: PPGCS Sample Size.

Data de publicação: 06/06/2022.

Data de criação: 06/06/2022.

Titular(es): UNIVERSIDADE ESTADUAL DE MONTES CLAROS - UNIMONTES

Autor(es): MARISE FAGUNDES SILVEIRA; ISMAEL MENDES DOS SANTOS JUNIOR; ROBERTO CÉSAR FARIA E SILVA; RENATO SOBRAL MONTEIRO JUNIOR

Linguagem: PHP

Campo de aplicação: MT-06

Tipo de programa: TC-01

Algoritmo hash: SHA-512

Resumo digital hash:

4f4b41aaa2c3319aadae5fbc85b4447fb7fa7b8bb31d364f7ea057a839891cadcc9f61e3ee586ac747adc5d65c4b54be8511eeac97a00969c752361d1d77d474

Expedido em: 02/08/2022

Aprovado por:

Joelson Gomes Pequeno

Chefe Substituto da DIPTO - PORTARIA/INPI/DIRPA Nº 02, DE 10 DE FEVEREIRO DE 2021

ANEXO C – Dados do Pesquisador O.T.S. (grupo GENESEs).

Exergame DELTA pré repouso	Exergame Theta pré repouso	Exergame DELTA pós repouso	Exergame Theta pós repouso
0,109579	0,236884	0,091385	0,335874
0,199214	0,35617	0,130528	0,269658
0,296973	0,273751	0,084218	0,236086
0,129882	0,283784	0,203005	0,33873
0,312528	0,300128	0,226131	0,26964
0,136533	0,252593	0,137673	0,300306
0,210000	0,208000	0,186000	0,303000
0,081213	0,321725	0,105707	0,231379
0,569723	0,192531	0,44835	0,322021
0,229678	0,254198	0,165635	0,293043

ANEXO D – Dados do Índice Mineiro de Responsabilidade Social.

Município	Mortalidade 2016	Mortalidade 2017	Gastos Per Capta 2016	Gastos Per Capta 2017
Abadia dos Dourados	4,6	7,44	652,89	731,63
Abaeté	6,39	7,02	543,86	603,07
Abre Campo	7,29	9,34	508,78	489,1
Acaiaca	5,23	8,45	712,08	787,04
Açucena	7,53	6,77	382,59	490,46
Água Boa	4,17	7,52	402,27	406,95
Água Comprida	8,33	9,82	1843,37	2009
Aguanil	7,57	7,28	802,45	894,71
Águas Formosas	7,98	7,75	615,77	643,76
Águas Vermelhas	4,94	4,69	438,27	460,64
Aimorés	8,69	8,64	330,8	974,48
Aiuruoca	9,74	12,04	532,91	597,48
Alagoa	7,7	5,87	1248,82	1473,56
Albertina	4,32	5,64	978,59	1082,71
Além Paraíba	8,95	8,9	412,63	475,79
Alfenas	7,05	7,93	939,36	1174,46
Alfredo Vasconcelos	5,57	6,71	425,47	556,22
Almenara	6,42	6,65	373,33	470,63
Alpercata	9,3	7,26	293,93	328,99
Alpinópolis	6,5	6,11	628,23	673,65
Alterosa	9,41	8,89	536,18	576,31
Alto Caparaó	6,15	3,84	331,95	669,74
Alto Jequitibá	8,65	8,89	546,23	550,51
Alto Rio Doce	10,45	10,37	524,3	751,86
Alvarenga	12,38	9,86	700,06	678,88
Alvinópolis	6,86	8,68	436,47	517,18
Alvorada de Minas	7,44	9,35	1255,46	972,21
Amparo da Serra	8,99	8,41	517,11	550,01
Andradas	8,3	8,24	500,03	572,74
Andrelândia	6,14	7,75	543,94	709,03
Angelândia	4,75	5,79	570,99	573,16
Antônio Carlos	6,89	7,92	383,05	412,24
Antônio Dias	7,32	6,18	769,19	719,83
Antônio Prado de Minas	6,65	12,75	1266,32	1551,69
Araçai	5,13	5,55	1078,61	1108,89
Aracitaba	8,61	10,05	788,12	1035,06

Araçuaí	6,77	6,73	575,81	580,29
Araguari	6,6	7,53	581,77	1429,37
Arantina	10,89	8,09	903,66	769,03
Araponga	6,52	5,44	640,45	684,22
Araporã	4,66	5,82	943,03	991,04
Arapuá	7,72	11,56	910,52	1104,2
Araújos	5,35	5,84	557,65	637,46
Araxá	6,97	7,32	631,32	714,23
Arceburgo	6,08	5,54	618,63	762,93
Arcos	6,7	6,19	563,76	580,72
Areado	7,01	7,63	346,48	390,17
Argirita	9,14	8,83	1142,39	1167,86
Aricanduva	5,88	4,28	578,16	762,44
Arinos	4,32	5,92	424,72	436,73
Astolfo Dutra	7,49	7,23	436,85	516,09
Ataléia	7,85	7,71	339,52	390,04
Augusto de Lima	6,43	8,45	532,37	594,97
Baependi	7,61	8	613,55	630,79
Baldim	8,65	8,41	507,81	507,98
Bambuí	8,36	8,87	552,87	619,72
Bandeira	5,25	6,28	543,88	683,3
Bandeira do Sul	7,61	7,39	607,79	758,52
Barão de Cocais	6,29	5,02	850,78	931,15
Barão do Monte Alto	6,59	8,94	790,33	817,97
Barbacena	7,34	7,71	761,54	772,86
Barra Longa	10,09	8,62	669,55	419,25
Barroso	7,24	7,06	482,49	436,28
Bela Vista de Minas	7,77	6,69	702,32	806,48
Belmiro Braga	10,39	9,52	1280,91	1188,18
Belo Horizonte	6,25	6,32	864,97	1320,82
Belo Oriente	4,88	5,87	356,25	297,38
Belo Vale	7,23	7,09	1375,16	1211,82
Berilo	4,73	5,23	399,91	477,11
Berizal	4,95	5,14	396,92	528,2
Bertópolis	7,14	7,13	464,42	
Betim	5,01	5,13	877,14	907,44
Bias Fortes	7,48	7,83	756,23	617,47
Bicas	10,05	10	425,99	468,35
Biquinhas	5,38	8,1	963,4	1019,41
Boa Esperança	8,2	7,25	488,34	567,61
Bocaina de Minas	6,63	7,01	555,85	686,7

Bocaiúva	6,38	6,14	425,74	426,22
Bom Despacho	6,33	6,74	678,28	737,15
Bom Jardim de Minas	10,03	12,62	774,64	832,01
Bom Jesus da Penha	7,01	4,81	1090,96	1060,77
Bom Jesus do Amparo	7,28	5,04	248,7	244,5
Bom Jesus do Galho	7,69	8,82	287,91	307,63
Bom Repouso	7,13	7,03	561,06	609,28
Bom Sucesso	8,87	8,24	395,42	441,92
Bonfim	7,34	8,78	579,19	628,67
Bonfinópolis de Minas	8,37	8,41	845,07	818,84
Bonito de Minas	3,93	4,8	436,8	537,98
Borda da Mata	6,53	7,58	496,59	533,25
Botelhos	7,65	8,17	475,78	520,23
Botumirim	5,09	5,87	503,86	496,31
Brasilândia de Minas	3,68	6,52	554,53	543,29
Brasília de Minas	7,12	7,53	981,59	1155,18
Brás Pires	5,3	9,32	629,18	728,11
Braúnas	7,04	6,06	708,68	768,79
Brazópolis	8,61	7,33	440,31	468,01
Brumadinho	6,32	5,64	1471,62	1417,31
Bueno Brandão	6,57	10,16	241,49	302,55
Buenópolis	5,91	8,3	486,9	559,74
Bugre	7,57	5,85	341,13	487,43
Buritis	4,57	5,44	493,08	510,57
Buritizinho	6,26	5,56	404,56	462,84
Cabeceira Grande	5,42	5,82	617	676,53
Cabo Verde	7,84	6,77	500,81	550,49
Cachoeira da Prata	8,7	8,44	847,04	889,19
Cachoeira de Minas	7,05	5,38	547,2	588,92
Cachoeira de Pajeú	8,88	6,93	359,27	381,56
Cachoeira Dourada	10,2	5,63	1679,37	1877,81
Caetanópolis	6,54	8,06	559,97	558,72
Caeté	6,74	6,53	392,12	420,03
Caiana	6,55	6,31	578,62	771,66
Cajuri	9,33	8,61	849,7	935,32
Caldas	6,85	8,34	475,54	532,14
Camacho	7,2	6,26	991,9	1110,91
Camanducaia	7,3	7,05	570,96	622,9
CambuÍ	6,81	7,79	248,72	293,3
Cambuquira	8,21	7,51	416,56	449,45
Campanário	8,09	7,53	884,38	857,39

Campanha	9,78	6,74	479,8	567,17
Campestre	8,35	7,35	360,64	420,69
Campina Verde	7,56	5,43	691,1	643,23
Campo Azul	6,56	4,71	655,19	756,87
Campo Belo	8,32	8,55	623,77	795,89
Campo do Meio	7,75	8,08	681,37	715,85
Campo Florido	6,49	6,02	556,5	511,61
Campos Altos	7,01	6,7	606,25	613,24
Campos Gerais	7,5	6,54	339,79	392,73
Canaã	7,09	5,37	878,95	1135,29
Canápolis	6,87	5,17	417,22	433,53
Cana Verde	7,75	9,51	661,4	717,83
Candeias	6,81	6,6	560,57	626,63
Cantagalo	6,75	4,47	394,62	522,44
Caparaó	5,36	6,81	694,9	732,61
Capela Nova	9,65	6,73	660,22	726,4
Capelinha	4,73	6	339,77	395,36
Capetinga	7,05	5,93	666,29	757,3
Capim Branco	6,94	6,37	509,23	477,92
Capinópolis	7,49	6,09	626,19	664,99
Capitão Andrade	6,97	6,54	536,06	719,24
Capitão Enéas	5,53	5,5	482,79	499,47
Capitólio	7,13	6,4	871,19	997,62
Caputira	6,66	6,64	484,38	441,9
Carai	6,57	7,31	426,41	301,81
Caranaíba	8,24	7,65	951,87	1102,95
Carandaí	7,65	7,8	685,12	676,49
Carangola	10,34	9,18	614,09	706,78
Caratinga	6,95	7,9	552,77	571,57
Carbonita	7,53	5,29	455,44	419,15
Careaçu	10,22	6,13	433,15	409,1
Carlos Chagas	9,26	8,85	360,04	467,4
Carmésia	6,18	9,22	834,96	844,81
Carmo da Cachoeira	6,8	6,87	598,66	621,13
Carmo da Mata	5,7	7,52	588,41	576,3
Carmo de Minas	6,17	5,86	478	557,54
Carmo do Cajuru	5,48	5,7	366,68	458,4
Carmo do Paranaíba	6,95	6,94	674,41	712,35
Carmo do Rio Claro	7,22	7,2	527,96	575,62
Carmópolis de Minas	6,28	5,96	559,25	607
Carneirinho	4,43	5,82	1117,49	981,5

Carrancas	7,88	9,58	985,98	446,94
Carvalhópolis	9,64	8,18	687,37	744,22
Carvalhos	9,83	10,5	689,18	768,49
Casa Grande	6,13	8,75	915,79	927,87
Cascalho Rico	5,62	6,25	1187,19	1160,97
Cássia	7,96	7,72	713,48	769,59
Cataguases	7,91	7,86	539,32	554,5
Catas Altas	6,51	5,32	1122,73	1123,53
Catas Altas da Noruega	5,53	5,51	656,19	694,46
Catuji	9,88	9,62	561,11	697,32
Catuti	6,26	8,82	597,35	508,06
Caxambu	9,1	9,32	410,89	465,82
Cedro do Abaeté	5	2,51	1353,94	1249,57
Central de Minas	7,69	11,65	553,67	478,18
Centralina	5,62	7,99	677,44	648,38
Chácara	10,2	10,1	865,13	1147,74
Chalé	8,51	10,06	781,26	691,26
Chapada do Norte	4,06	3,03	430,43	415,64
Chapada Gaúcha	2,54	3,74	557,98	646,32
Chiador	8,31	9,78	1067,1	1163,11
Cipotânea	7,52	7,35	448,38	510,38
Claraval	1,47	3,13	576,1	654,62
Claro dos Poções	9,8	6,33	552,98	576,35
Cláudio	6,81	5,64	544,23	546,14
Coimbra	10,62	9,09	535,53	606,22
Coluna	6,18	5,85	546,67	575,6
Comendador Gomes	3,55	4,83	1224,85	1449,37
Comercinho	7,86	6,51	492,69	514,15
Conceição da Aparecida	4,89	8,08	719,19	708,23
Conceição da Barra de Minas	6,23	9,23	673,17	847,08
Conceição das Alagoas	6,23	6,06	659,88	701,3
Conceição das Pedras	8,14	4,95	914,2	1212,13
Conceição de Ipanema	5,23	9,14	554,43	692,68
Conceição do Mato Dentro	6,51	7,19	860,52	1014,67
Conceição do Pará	5,89	6,04	1001,31	1051,91
Conceição do Rio Verde	7,32	7,36	436,07	469,26
Conceição dos Ouros	6,76	7,84	365,22	387,61
Cônego Marinho	6,24	5,7	308,1	376
Confins	6,18	5,96	1499,71	1362,62
Congonhal	6,54	7,51	639,9	690,07
Congonhas	6,48	6,51	648,48	1200,74

Congonhas do Norte	5,71	7,87	632,92	604,57
Conquista	6,86	8,56	744,75	889,27
Conselheiro Lafaiete	6,55	7,12	556,08	621,59
Conselheiro Pena	6,45	7,21	392,17	412,51
Consolação	6,15	7,81	1234,59	1349,02
Contagem	5,34	5,33	243,97	289,77
Coqueiral	7,27	7,18	321,16	440,42
Coração de Jesus	5,24	6,38	488,29	490,48
Cordisburgo	7,85	5,93	347,3	439,39
Cordislândia	11,01	8,73	827,06	256,25
Corinto	8,36	7,99	453,25	463,16
Coroaci	8,58	7,82	468,01	479,56
Coromandel	6	7,72	463,23	540,11
Coronel Fabriciano	6,48	6,17	265,94	379,58
Coronel Murta	6,45	5,69	396,15	441,97
Coronel Pacheco	10,05	6,79	867,63	1107,57
Coronel Xavier Chaves	5,26	4,66	608,73	676,55
Córrego Danta	8,39	7,22	785,71	911,67
Córrego do Bom Jesus	12,47	10,62	758,95	727,73
Córrego Fundo	5,82	5,3	479,74	659,61
Córrego Novo	5,05	7,48	359,84	422,13
Couto de Magalhães de Minas	9,13	11,36	607,33	727,37
Crisólita	5,07	5,94	364,55	302,29
Cristais	5,52	5,95	655,65	539,39
Cristália	4,53	3,34	559,88	583,51
Cristiano Ottoni	6,78	5,41	664,83	697,82
Cristina	8,1	8	606,46	694,46
Crucilândia	11,44	11,19	572,21	674,09
Cruzeiro da Fortaleza	8,02	7,99	930,44	787,22
Cruzília	8,84	8,8	623,65	611,33
Cuparaque	8,33	9,1	583,95	691,24
Curral de Dentro	5,48	5,96	434,29	504,07
Curvelo	7,08	7,56	703,63	782,16
Datas	7,04	8,12	452,22	511,14
Delfim Moreira	5,42	7,88	531,38	578,34
Delfinópolis	7,62	7,45	1294,13	1215,24
Delta	4,48	3,37	638,63	520,61
Descoberto	7,43	9,41	505,21	524,64
Desterro de Entre Rios	9,53	8,68	637,42	716,44
Desterro do Melo	16,69	12,06	881,77	1110,43
Diamantina	6,01	5,95	392,31	601,03

Diogo de Vasconcelos	5,67	9,29	1118,58	878,11
Dionísio	7,48	6,95	608,5	457,14
Divinésia	8,21	9,65	900,34	967,26
Divino	8,96	9,03	490,84	581,83
Divino das Laranjeiras	9,14	8,94	704	611,62
Divinolândia de Minas	6,32	5,35	433,2	510,47
Divinópolis	6,38	6,17	687,93	807,21
Divisa Alegre	7,4	4,88	455,73	530,4
Divisa Nova	6,51	7,49	601,74	660,32
Divisópolis	4,18	2,68	376,77	400,93
Dom Bosco	7,92	6,35	724,04	653,02
Dom Cavati	9,6	8,09	568,53	485,2
Dom Joaquim	7,67	7,68	587,96	772,96
Dom Silvério	4,91	8,49	700,4	776,61
Dom Viçoso	7,23	7,56	901,13	924,73
Dona Euzébia	7,17	8,36	486,66	477,98
Dores de Campos	6,15	8,01	419,68	440,75
Dores de Guanhães	7,59	5,89	392,05	546,03
Dores do Indaiá	8,04	9,58	488,97	574,32
Dores do Turvo	10,89	8,42	682,7	701,29
Doresópolis	3,31	3,95	1297,15	1618,35
Douradoquara	6,82	4,71	1369,1	1421,47
Durandé	4,89	5,51	387,49	397,64
Elói Mendes	6,25	7,9	569,51	581,35
Engenheiro Caldas	6,14	7,74	459,72	506,08
Engenheiro Navarro	5,76	7,53	580,46	648,15
Entre Folhas	8,05	7,79	638,05	713,96
Entre Rios de Minas	7,31	8,92	383,18	429,26
Ervália	5,76	7,12	493,4	589,67
Esmeraldas	5,87	5,33	417,14	365,35
Espera Feliz	6,48	7,66	503,34	566,01
Espinosa	7,6	6,33	327,93	388,15
Espírito Santo do Dourado	8,62	6,43	809,98	972,2
Estiva	7,44	6,44	384,11	448,31
Estrela D'Alva	13,15	9,08	957,97	1254,37
Estrela do Indaiá	8,72	8,16	801,4	985,99
Estrela do Sul	6,24	6,83	591,85	653,46
Eugenópolis	7,92	8,15	560,08	541,45
Ewbank da Câmara	11,03	8,69	819,62	922,74
Extrema	6,56	7,12	1080,9	1189,88
Fama	5,42	9,16	977,67	992,58

Faria Lemos	7,17	7,79	657,41	592,99
Felício dos Santos	7,81	9,67	576,81	696,43
Felisburgo	7,66	7,34	516,15	444,49
Felixlândia	6,59	6,15	336,19	354,37
Fernandes Tourinho	6,36	8,39	656,28	768,18
Ferros	7,11	7,75	557,44	611,02
Fervedouro	5,61	6,68	613,06	678,3
Florestal	7,08	6,05	711,17	838,99
Formiga	8,13	8,56	383	448,03
Formoso	2,31	4,46	403,94	525
Fortaleza de Minas	8,07	4,35	791,33	819,24
Fortuna de Minas	9,71	8,61	944,75	1065,83
Francisco Badaró	7,09	4,98	401,2	407,38
Francisco Dumont	5,26	4,84	739,86	661,73
Franciscópolis	5,31	4,99	582,48	598,32
Francisco Sá	5,72	5,47	528,83	447,66
Frei Gaspar	8,05	8,89	419,74	285,9
Frei Inocência	10,27	8,42	487,81	550,53
Frei Lagonegro	4,62	4,6	773,8	777,14
Fronteira	4,77	4,32	826,2	938,04
Fronteira dos Vales	6,39	7,68	610,71	525,65
Fruta de Leite	7,72	8,85	457,02	585,52
Frutal	7,61	7,87	656,22	711,35
Funilândia	7,15	4,25	770,44	774,64
Galiléia	8,75	8,33	416,17	410,48
Gameleiras	5,77	7,32	620,34	622,61
Glaucilândia	5,14	4,48	845,02	917,14
Goiabeira	7,34	7,89	637,1	517,56
Goianá	9	8,43	806,68	907,16
Gonçalves	6,66	4,81	592,68	694,89
Gonzaga	4,23	6,01	717,14	776,52
Gouveia	6,71	6,7	296,68	403,78
Governador Valadares	6,93	7,94	551,2	599,19
Grão Mogol	6,3	4,82	550,71	587,44
Grupiara	2,14	8,55	1760,61	2187,76
Guanhães	6,51	6,32	602,11	605,9
Guapé	6,51	8,1	552,2	622,36
Guaraciaba	8,82	7,47	238,45	412,41
Guaraciama	7,1	8,08	566,94	573,25
Guaranésia	7,1	8,97	490,04	663,92
Guarani	8,28	8,15	458,91	472,32

Guarará	8,95	8,46	882,87	867,95
Guarda-Mor	4,5	4,95	1053,17	1018,34
Guaxupé	8,03	7,78	319,35	408,02
Guidoval	7,05	7,89	596,64	725,77
Guimarânia	5,89	8,76	500,5	481,8
Guiricema	8,32	7,65	404,35	452,12
Gurinhata	4,88	6,61	781,76	626,74
Heliodora	8,05	7,85	371,09	360,26
Iapu	5,65	6,82	515,24	307,84
Ibertioga	8,82	10,79	850,5	990,83
Ibiá	5,95	7	555,5	594,99
Ibiaí	5,92	4,81	581,03	578,32
Ibiracatu	6,86	6,39	572,14	636,46
Ibiraci	3,61	4,39	862,49	852,99
Ibirité	4,89	4,28	469,95	503,06
Ibitiúra de Minas	7,45	8	1043,71	1179,93
Ibituruna	6,72	11,38	354,42	331,26
Icaraí de Minas	5,42	5,04	431,89	500,49
Igarapé	5,52	5,67	437,86	463,57
Igaratinga	5,24	5,46	718,29	806,88
Iguatama	9,88	6,8	750,03	761,56
Ijaci	4,1	5	886,16	848,51
Ilicínea	6,99	6,87	574,34	615,04
Imbé de Minas	6,92	6,73	475,33	565,19
Inconfidentes	8,83	7,28	523,98	593,09
Indaiabira	4,3	6,44	497,62	477,98
Indianópolis	4,34	4,6	903,37	952,13
Ingaí	6,19	7,98	1711,41	1532,59
Inhapim	7,17	7,29	567,45	510,57
Inhaúma	7,17	7,28	797,36	823,37
Inimutaba	6,11	7,28	350,25	519,14
Ipaba	5,82	5,28	412,96	430,37
Ipanema	7,32	9,52	525,62	516,11
Ipatinga	4,94	5,19	681,14	682,98
Ipiaçu	8,03	7,54	280,9	474,88
Ipuiúna	7,12	6,29	551,05	609,58
Iraí de Minas	4,38	5,65	389,31	619,1
Itabira	5,62	5,69	1027,94	886,46
Itabirinha	7,42	7,64	451,91	555,11
Itabirito	5,81	6,34	826,75	838,56
Itacambira	3,4	5,45	487,64	576,16

Itacarambi	5,87	5,97	605,83	676,65
Itaguara	8,39	7,73	605,29	703,42
Itaipé	7,59	7,22	426,26	511,65
Itajubá	6,56	6,72	785,16	864,44
Itamarandiba	5,1	5,71	443,98	460,14
Itamarati de Minas	4,89	8,1	317,12	410,84
Itambacuri	9,29	9,79	574,99	584,98
Itambé do Mato Dentro	9,54	3,66	1399,92	1594,6
Itamogi	9,32	9,23	736,28	753,27
Itamonte	8,27	6,3	398,42	345,32
Itanhandu	6,98	7	1065,65	1222,78
Itanhomi	8,82	7,74	442,09	556,08
Itaobim	8,34	8,53	491,18	534,74
Itapagipe	5,22	4,77	657,96	772,47
Itapecerica	8,95	8,84	379,43	486,44
Itapeva	7,74	7,56	641,3	750,13
Itatiaiuçu	8,36	8,19	1764,33	1496,23
Itaú de Minas	6,7	6,91	933,47	971,26
Itaúna	7,46	7,08	628,46	678,35
Itaverava	6,89	8,17	569,03	541,2
Itinga	5,22	6,14	368,44	354,7
Itueta	4,8	7,76	519,94	604,66
Ituiutaba	7,49	8,05	533,38	566,1
Itumirim	8,28	7,64	605,46	709,87
Iturama	6,74	5,98	776,39	
Itutinga	6,67	9,52	1034,87	1066,79
Jaboticatubas	6,49	6,62	466,3	500,33
Jacinto	7,65	6,51	566,55	607,54
Jacuí	6,74	9,57	605,69	675
Jacutinga	5,45	6,35	602	724,39
Jaguaraçu	7,06	6,72	1097,31	1126,84
Jaíba	5,49	5,09	350,09	391,33
Jampruca	8,25	8,02	535,64	489,06
Janaúba	6,34	6,3	1086,55	1129,23
Januária	5,89	6,05	370,08	442,28
Japaraíba	5,2	7,27	683,49	800,7
Japonvar	7,58	5,12	566,02	545,9
Jeceaba	10,2	10,28	1050,36	1329,97
Jenipapo de Minas	3,84	6,58	415,88	453,63
Jequeri	6,81	8,33	571,14	566,16
Jequitaiá	7,13	8,07	498,32	486,32

Jequitibá	7,6	6,46	698,84	668,05
Jequitinhonha	6,43	7,67	541,44	648,61
Jesuânia	7,84	7,63	539,95	701,81
Joáma	7,32	5,88	494,11	517,01
Joanésia	7,98	7,08	817,13	713
João Monlevade	6,09	6,33	720,04	734,43
João Pinheiro	6,92	6,51	691,2	697,45
Joaquim Felício	7,62	9,3	712,92	604,7
Jordânia	5,86	7,23	393,46	422,6
José Gonçalves de Minas	5,23	4,36	651,87	653,22
Josenópolis	6,45	6,21	594,61	615,19
José Raydan	5,84	6,81	557,07	565,4
Juatuba	6,1	5,35	695,6	762,69
Juiz de Fora	7,97	7,66	984,6	1014,36
Juramento	5,12	7,19	747,65	718,76
Juruaia	6,42	4,79	631,04	641,65
Juvenília	3,79	3,28	474,46	466,65
Ladainha	7,66	7,62	410,21	482,11
Lagamar	6,87	7,52	684,79	670,97
Lagoa da Prata	5,76	6,19	469,85	482,37
Lagoa dos Patos	8,3	8,08	638,31	761,89
Lagoa Dourada	5,99	7,35	410,08	462,7
Lagoa Formosa	6,58	8,11	606,7	639,27
Lagoa Grande	6,9	4,28	476,32	532,86
Lagoa Santa	5,29	5,5	642,56	625,51
Lajinha	6,97	7,91	429,11	442,75
Lambari	8,76	8,91	458,14	438,32
Lamim	6,63	9,24	926,3	972,58
Laranjal	7,55	7,37	568,61	602,97
Lassance	6,52	5,61	490,15	375,97
Lavras	6,67	6,89	778,13	843,53
Leandro ferreira	9,49	7,65	1014,15	932,46
Leme do Prado	3,85	3,44	607,35	788,23
Leopoldina	8,44	8,42	458,55	492,58
Liberdade	10,07	10,11	554,82	659,38
Lima Duarte	8,21	8,96	546,48	623,5
Limeira do Oeste	8,7	6,21	837,2	967,66
Lontra	6,52	7,59	448,67	425,82
Luisburgo	6,94	6,15	637,43	818,17
Luislândia	7,65	6,58	424,68	499,88
Luminárias	9,8	7,8	603,3	661,48

Luz	7,88	7,8	656,92	701,62
Machado	6,6	6,53	559,83	615,57
Madre de Deus de Minas	7,67	8,04	858,36	858,1
Malacacheta	8,01	7,7	648,42	626,81
Mamonas	5,05	5,49	404,81	386,3
Manga	8,03	7,98	314,37	386,86
Manhuaçu	5,8	5,91	651,11	725,96
Manhumirim	8,73	8,69	449,04	510,01
Mantena	7,34	7,04	270,59	361,63
Maravilhas	5,17	4,87	626,09	677,22
Mar de Espanha	7,98	8,4	431,67	419,19
Maria da Fé	7,87	7,11	644,04	738,48
Mariana	5,35	5,67	990,15	984,51
Marilac	9,24	7,36	484,33	437,4
Mário Campos	4,78	4,31	339,7	402,13
Maripá de Minas	10,22	6,78	1055,29	966,97
Marliéria	7,1	8,56	430,17	507,04
Marmelópolis	7,62	6,97	794,9	1032,07
Martinho Campos	7,25	6,47	566,02	527,56
Martins Soares	6,26	7,29	533,9	552,49
Mata Verde	5,72	5,91	372,86	376,14
Materlândia	7,2	9,18	249,63	211,92
Mateus Leme	6,44	6,82	509,44	534,05
Mathias Lobato	8,43	6,65	544,51	564,54
Matias Barbosa	7,1	7,13	1004,39	1055,43
Matias Cardoso	5,47	4,13	568,6	594,67
Matipó	5,37	8,87	499,93	615,01
Mato Verde	7,62	8,02	448,86	465,1
Matozinhos	5,81	6,82	515,56	503,31
Matutina	6,83	9,19	743,7	767,18
Maxacalis	7,69	7,96	423,15	483,87
Medeiros	7,03	6,17	698,87	437,5
Medina	8,2	9,72	383,61	475,99
Mendes Pimentel	5,24	6,77	550,58	566,39
Mercês	7,74	8,55	477,71	402,48
Mesquita	7,13	8,03	468,17	436,83
Minas Novas	5,63	5,21	351,21	311,71
Minduri	5,09	7,63	853,58	979,2
Mirabela	7,09	8,09	525,43	549,53
Miradouro	6,27	9,32	402	430,22
Miraí	7,01	7,64	361,37	416,01

Miravânia	6,24	4,96	442,2	391,9
Moeda	7,36	11,21	749,33	823,3
Moema	7,15	5,24	552,28	581,33
Monjolos	6,48	6,51	1538,75	1442,75
Monsenhor Paulo	9,32	6,96	656,91	602,29
Montalvânia	6,5	6,09	365,63	427,12
Monte Alegre de Minas	6,21	6,03	511,02	569,46
Monte Azul	6,88	7,28	432,37	460,07
Monte Belo	6,24	9,31	429,4	476,32
Monte Carmelo	6,53	6,68	533,18	622,25
Monte Formoso	7,6	7,78	601,29	492,08
Monte Santo de Minas	7,97	8,38	547,04	657,12
Montes Claros	5,35	5,15	371,64	342
Monte Sião	5,7	5,43	596,81	566,32
Montezuma	4,24	5,07	403,55	486,81
Morada Nova de Minas	7,8	6,5	617,6	653,68
Morro da Garça	8,9	3,11	876,7	883,09
Morro do Pilar	7,52	11,8	514,28	674,99
Munhoz	6,59	6,94	585,3	706,59
Muriaé	8,16	7,6	347,53	506,41
Mutum	6,5	6,31	463,77	364,53
Muzambinho	9,33	8,5	502,72	555,88
Nacip Raydan	6,18	6,17	836,73	707,72
Nanuque	6,96	8,82	490,71	501,96
Naque	4,26	5,25	527,6	535,75
Natalândia	4,48	4,48	879,34	919,09
Natércia	10,49	7,13	863,12	993,12
Nazareno	6,87	7,18	516,02	589,38
Nepomuceno	8,02	7,73	322,52	392,92
Ninheira	4,98	5,74	558,04	651,76
Nova Belém	4,6	6,12	1058,88	1258,49
Nova Era	7,19	6,46	514,71	562,86
Nova Lima	5,84	5,77	1283,48	1285,52
Nova Mógica	5,89	5,91	546,82	963,58
Nova Ponte	5,01	4,26	854,9	978,52
Nova Porteirinha	4,89	7,26	510,36	506,88
Nova Resende	5,44	5,83	623,63	712,06
Nova Serrana	3,45	3,74	401,48	511,61
Nova União	7,68	6,79	248,27	239,72
Novo Cruzeiro	6,03	7,19	356,86	351,97
Novo Oriente de Minas	8,2	9,39	426,79	444,8

Novorizonte	4,21	8,75	496,26	519,34
Olaria	9,61	5,93	1335,35	1302,29
Olhos D' Água	6,02	4,42	537,23	605,46
Olímpio Noronha	7,73	9,14	1152,73	1167,69
Oliveira	7,19	7,98	620,44	651,08
Oliveira Fortes	7,87	5,55	866,11	1085,5
Onça do Pitangui	7,93	5,7	1050,92	1149,86
Oratórios	7,74	6	696,95	708,49
Orizânia	5,25	6,35	612,1	561,92
Ouro Branco	5,31	5,37	733,92	769,58
Ouro Fino	8,49	7,67	387,17	398,86
Ouro Preto	5,99	6,16	1046,9	1009,22
Ouro Verde de Minas	10,9	10,41	305,14	369,62
Padre Carvalho	6,9	4,62	442,71	421,04
Padre Paraíso	7,75	7,71	445,84	445,36
Paineiras	9,1	6,95	730,18	736,24
Pains	7,12	7,34	922,82	988,54
Pai Pedro	6,22	5,72	562,47	557,17
Paiva	12,79	12,17	1482,32	1349,69
Palma	10,63	9,57	503,72	450,86
Palmópolis	6,6	9,67	666,78	534,81
Papagaios	7,09	7,55	575,53	593,14
Paracatu	5,36	5,57	792,71	868,19
Pará de Minas	6,25	6,4	738,53	700,69
Paraguaçu	7,67	8,1	550,76	660,79
Paraisópolis	6,54	6,74	511	511,57
Paraopeba	6,2	6,33	485,97	490,66
Passabém	8,7	6,42	1229,12	1508,73
Passa Quatro	9,23	7,73	628,79	603,07
Passa Tempo	10,06	8,13	609,83	610,89
Passa Vinte	3,84	7,68	1223,62	1298,6
Passos	6,8	6,81	519,72	508,7
Patis	4,42	4,73	492,52	507,07
Patos de Minas	6,61	6,41	832,08	835,22
Patrocínio	6,16	6,38	719,8	819,42
Patrocínio do Muriaé	8,59	7,47	576,5	762,81
Paula Cândido	8,46	6,35	702,79	705,84
Paulistas	5,67	5,88	473,06	474,3
Pavão	7,41	8,23	375,31	381,15
Peçanha	6,34	6,22	243,88	291,13
Pedra Azul	6,83	7,31	391,02	395,84

Pedra Bonita	3,85	4,26	619,57	551,67
Pedra do Anta	7,34	9,27	1150,35	1333,35
Pedra do Indaiá	8,53	7,01	838,05	871,27
Pedra Dourada	5,41	7,41	1238,91	1065,61
Pedralva	6,8	8,73	497,43	595,82
Pedras de Maria da Cruz	5,19	5,76	500,05	479,53
Pedrinópolis	6,9	9,08	1271,88	1396,04
Pedro Leopoldo	6,09	6,91	627,22	644,64
Pedro Teixeira	3,84	6,58	1396,38	1514,43
Pequeri	4,24	5,73	967,14	1011,88
Pequi	6,71	7,12	934,24	867,69
Perdigão	6,36	5,49	440,17	477,69
Perdizes	4,99	4,31	773,33	816,98
Perdões	5,87	6,41	468,08	546,4
Periquito	9,28	8,73	373,57	411,59
Pescador	7,98	7,5	653,62	
Piau	8,48	8,86	1099,76	1091,97
Piedade de Caratinga	6,84	6,74	714,82	765,86
Piedade de Ponte Nova	8,65	9,11	657,91	624,28
Piedade do Rio Grande	7,09	8,84	955,52	932,5
Piedade dos Gerais	3,88	4,26	721,42	878,23
Pimenta	5,23	8,22	517,06	551,46
Pingo D' Água	8,99	6,63	651,01	466,45
Pintópolis	3,47	4,93	369,13	422,63
Piracema	9,08	10,46	536,21	598,13
Pirajuba	6,24	4,88	842,17	964,77
Piranga	7,06	6,43	529,63	588,65
Piranguçu	7,72	6,59	573,64	615,33
Piranguinho	6,73	7,28	314,69	370,44
Pirapetinga	8,29	8,17	810,71	931,37
Pirapora	6,66	6,2	749,46	825,15
Piraúba	7,2	7,93	426,15	562,56
Pitangui	6,73	6,82	431,18	480,6
Piumhi	6,54	7,52	591,85	741,68
Planura	5,72	5,31	682,39	785,2
Poço Fundo	6,24	6,75	549,22	607,23
Poços de Caldas	6,61	7,49	828,6	838,9
Pocrane	7,5	9,48	540,76	485,42
Pompéu	6,73	6,74	592,6	645,55
Ponte Nova	7,61	7,88	1039,72	1072,85
Ponto Chique	3,82	5,45	624,39	853,66

Ponto dos Volantes	6,78	7,49	456,19	393,08
Porteirinha	6,18	5,92	443,36	447,74
Porto Firme	7,86	7,46	516,07	588,06
Poté	7,68	9,04	434,01	460,44
Pouso Alegre	6,3	6,4	718,14	638,6
Pouso Alto	9,27	10,78	805,66	885,38
Prados	8,45	7,95	409,45	433,23
Prata	6,47	6,4	677,3	689,14
Pratápolis	9,53	9,32	690,16	740,89
Pratinha	5,42	7,08	972,85	1025,25
Presidente Bernardes	9,78	9,63	795,92	885,62
Presidente Juscelino	5,25	7,65	757,87	877,26
Presidente Kubitschek	6,61	6,93	688,31	781,88
Presidente Olegário	6,67	6,24	630,84	689,89
Prudente de Moraes	7,61	7,07	504,62	473,56
Quartel Geral	4,86	7,38	861,16	799,39
Queluzito	7,24	11,35	436,23	462,86
Raposos	5,64	6,29	350,33	390,6
Raul Soares	7,63	8,5	385,21	420,45
Recreio	9,74	11,43	542,6	514,94
Reduto	4,86	6,95	610,54	584,67
Resende Costa	9,03	8,03	365,1	404,26
Resplendor	9,31	8,78	597,6	627,83
Ressaquinha	9,8	5,4	593,16	733,59
Riachinho	4,51	5,12	653,86	743,56
Riacho dos Machados	5,44	5,33	415,22	453,35
Ribeirão das Neves	4,87	4,9	420,67	432,46
Ribeirão Vermelho	8	9,46	932,2	944,66
Rio Acima	6,05	5,69	1081,42	954,05
Rio Casca	8,26	8,22	471,47	473,06
Rio Doce	5,8	7,32	1042,7	1409,24
Rio do Prado	5,33	8	554,69	582,8
Rio Espera	10,66	11,27	687,77	727,8
Rio Manso	4,76	5,6	496,36	508,52
Rio Novo	8,69	7,56	536,37	606,81
Rio Paranaíba	5,2	6,48	747,67	725,06
Rio Pardo de Minas	5,3	5,54	374,88	385,47
Rio Piracicaba	6,5	7,46	556,46	564,57
Rio Pomba	8,14	8,33	432,72	424,77
Rio Preto	7,86	6,92	591,26	498,84
Rio Vermelho	7,17	7,28	356,05	450,19

Ritópolis	11,01	9,82	375,86	395,04
Rochedo de Minas	5,32	5,73	1141,67	1135,52
Rodeiro	8,08	7,58	622,66	712,76
Romaria	6,92	9,15	826,16	843,83
Rosário da Limeira	6,66	7,94	780,53	900,27
Rubelita	8,01	9,08	426,57	487,5
Rubim	7,03	7,88	408,42	412,62
Sabará	6,05	5,79	167,69	193,99
Sabinópolis	6,58	7,29	496,93	490,94
Sacramento	6,3	7,58	697,18	891,3
Salinas	6,92	6,83	485,4	504,9
Salto da Divisa	6,53	5,53	684,17	818,33
Santa Bárbara	5,54	5,86	766,89	771,01
Santa Bárbara do Leste	5,83	6,54	382	415,17
Santa Bárbara do Monte Verde	6,26	4,57	936,6	1011,58
Santa Bárbara do Tugúrio	7,46	8,14	561,92	1076,56
Santa Cruz de Minas	6,67	8,04	338,84	363,45
Santa Cruz de Salinas	9,27	6,52	591,37	588,64
Santa Cruz do Escalvado	8,12	8,55	889,07	974,98
Santa Efigênia de Minas	5,93	5,73	564,85	465,6
Santa Fé de Minas	4,55	4,82	751,3	822,5
Santa Helena de Minas	6,8	5,67	346,86	418,4
Santa Juliana	5,3	5,13	727,12	871,46
Santa Luzia	5,62	5,56	382,38	452,29
Santa Margarida	5,66	5,37	601,29	584,65
Santa Maria de Itabira	6,44	5,32	379,09	450,88
Santa Maria do Salto	6	5,07	398,01	451,5
Santa Maria do Suaçuí	6,38	8,14	341,5	338
Santana da Vargem	6,75	5,38	745,71	805,44
Santana de Cataguases	8,65	7,3	725,18	889,93
Santana de Pirapama	7,46	6,34	568,1	565,62
Santana do Deserto	5,27	7,77	872,78	1014,8
Santana do Garambéu	10,44	3,32	1240,17	1371,96
Santana do Jacaré	9,59	8,73	691,22	744,12
Santana do Manhuaçu	6,17	5,13	474,83	450,96
Santana do paraíso	4,55	5,2	454,63	269,15
Santana do Riacho	5,43	6,34	708,58	743,84
Santana dos Montes	6,75	7,54	833,05	775,85
Santa Rita de Caldas	9	8,24	632,94	594,04
Santa Rita de Jacutinga	10,4	7,41	660,89	660,33
Santa Rita de Minas	8,54	6,64	315,56	353,67

Santa Rita do Ibitipoca	7,06	5,95	1307,75	1123,26
Santa Rita do Itueto	7,07	7,8	429,95	530,9
Santa Rita do Sapucaí	5,89	6,49	469,96	551,6
Santa Rosa da Serra	6,58	7,76	858,85	990,54
Santa Vitória	7,04	7,15	1076,66	1185,31
Santo Antônio do Amparo	7,33	8,11	729,29	784,89
Santo Antônio do Aventureiro	7,45	6,33	719,56	926,38
Santo Antônio do Grama	7,91	12,41	1002,39	1012,94
Santo Antônio do Itambé	7,21	4,75	575,96	759,48
Santo Antônio do Jacinto	4,13	3,96	380,17	393,74
Santo Antônio do Monte	5,57	5,93	713,69	701,98
Santo Antônio do Retiro	4,7	5,37	478,25	490,68
Santo Antônio do Rio Abaixo	3,34	8,36	1237,11	1365,2
Santo Hipólito	7,19	6,59	1046,43	1004,26
Santos Dumont	8,54	8,05	541,99	594,87
São Bento Abade	4,75	5,87	658,36	776,52
São Brás do Suaçuí	6,23	8,63	1045,42	1012,07
São Domingos das Dores	4,09	5,14	838,43	836,35
São Domingos do Prata	6,93	7,04	499,87	538,93
São Félix de Minas	6,43	6,72	1105,52	1265,57
São Francisco	5,91	6,76	400,4	423,03
São Francisco de Paula	8,48	6,96	582,86	561,88
São Francisco de Sales	5,72	4,38	932	924,03
São Francisco do Glória	8,69	9,14	641,6	667,75
São Geraldo	6,97	6,19	493,1	525,87
São Geraldo da Piedade	6,42	8,63	764,32	905,65
São Geraldo do Baixio	6,25	5,15	691,64	715,81
São Gonçalo do Abaeté	6,2	4,54	563,68	553,06
São Gonçalo do Pará	5,56	6,91	438,63	508,72
São Gonçalo do Rio Abaixo	5,2	5,63	2751,19	2391,32
São Gonçalo do Rio Preto	4,42	3,47	904,33	859,65
São Gonçalo do Sapucaí	7,48	7,64	477,28	488,96
São Gotardo	5,67	6,63	487,11	667,61
São João Batista do Glória	6,16	6,52	820,53	944,8
São João da Lagoa	5,54	5,72	510,79	570,97
São João da Mata	7,55	7,55	976,1	1129,75
São João da Ponte	5,51	5,63	530,83	504,6
São João das Missões	6,02	6,05	289,91	354,36
São João Del Rei	8,77	8,33	637,67	679,76
São João do Manhuaçu	6,1	6,58	562,32	556,88
São João do Manteninha	9,74	9,65	469,2	586,8

São João do Oriente	9,14	9,43	534,32	527,5
São João do Pacuí	7,86	5,74	541,61	511,46
São João do Paraíso	6,11	6,04	373,16	346,41
São João Evangelista	7,61	7,29	569,6	520,34
São João Nepomuceno	7,41	8,68	600,52	636,3
São Joaquim de Bicas	5,38	5,36	466,35	571,58
São José da Barra	4,69	5,48	1112,32	1200,59
São José da Lapa	5,95	6,26	348,26	361,06
São José da Safira	4,48	5,63	608,58	551,95
São José da Varginha	5,93	3,97	201,51	221,34
São José do Alegre	6,24	6,69	431,67	589,38
São José do Divino	5,89	4,1	659,78	788,46
São José do Goiabal	5,18	8,97	616,78	679,4
São José do Jacuri	6,68	7,59	313,71	346,61
São José do Mantimento	6,93	8,7	977,29	1107,49
São Lourenço	8,37	9,07	997,68	1086,14
São Miguel do Anta	7,62	9,03	614,69	682,8
São Pedro da União	6,74	8,64	937,92	1015,66
São Pedro dos Ferros	8,35	6,79	452,9	407,39
São Pedro do Suaçuí	8,42	6,07	380,1	411,32
São Romão	3,36	3,4	413,46	352,99
São Roque de Minas	6,15	6,12	660,67	820,53
São Sebastião da Bela Vista	6,74	6,32	551,37	720,81
São Sebastião da Vargem Alegre	6,76	7,06	1081,17	1192
São Sebastião do Anta	5,09	6,29	643,18	658,69
São Sebastião do Maranhão	5,36	7,02	394,34	394,33
São Sebastião do Oeste	6,36	5,06	1002,53	1070,6
São Sebastião do Paraíso	7,14	6,82	1146,84	1114,17
São Sebastião do Rio Preto	10,79	4,47	1250,41	1311,31
São Sebastião do Rio Verde	6,77	9,43	1344,04	1596,8
São Tiago	6,86	6,75	466,93	542,01
São Tomás de Aquino	7,27	6,99	797,61	875,79
São Tomé das Letras	5,57	5,54	545,66	643,78
São Vicente de Minas	7,17	7,64	682,16	692,89
Sapucai Mirim	5,65	5,6	583,21	591,91
Sardoá	5,26	6,19	577,99	476,6
Sarzedo	4,58	4,56	876,19	885,18
Sem-Peixe	9,04	6,19	1019,69	1022,38
Senador Amaral	5,58	6,69	678,24	811,51
Senador Cortês	5,43	4,44	1218,89	1294,53
Senador Firmino	7,56	7,78	464,34	582,46

Senador José Bento	9,5	9,06	1191,76	1386,18
Senador Modestino Gonçalves	7,27	4,81	757,04	628,99
Senhora de Oliveira	6,87	8,57	622,57	710,09
Senhora do Porto	8,14	8,97	891,07	857,2
Senhora dos Remédios	7,72	6,94	499,02	483,32
Sericita	5,99	6,79	582,35	592,61
Seritinga	7,02	4,31	1320,66	1453,19
Serra Azul de Minas	4,63	6,93	537,73	612,73
Serra da Saudade	4,96	8,71	2968,98	3766,4
Serra dos Aimorés	7,82	6,31	223,02	236,19
Serra do Salitre	5,4	6,15	734,89	842,68
Serrania	8,55	9,44	652,26	780,42
Serranópolis de Minas	5,54	8,9	709,27	682,53
Serranos	10,47	2,5	1296,54	1384,82
Serro	6,7	7,02	353,43	377,89
Sete Lagoas	6,16	5,89	795,54	779,65
Setubinha	3,64	6,62	427,56	423,25
Silveirânia	5,3	7,05	1180,79	1223,33
Silvianópolis	6,58	8,64	492,73	498,28
Simão Pereira	9,16	8	1245,25	1378,22
Simonésia	5,23	5,71	551,23	538,09
Sobralia	7,82	8,2	631,72	800,94
Soledade de Minas	8,95	8,57	584,92	646,91
Tabuleiro	8,86	12,23	701,56	810,85
Taiobeiras	5,81	5,91	639,04	783,67
Taparuba	9,48	11,38	832,85	946,68
Tapira	6,59	4,78	1390,28	1762,3
Tapiraí	6,84	7,36	1264,16	1355,7
Taquaraçu de Minas	7,23	5,45	249,23	562,08
Tarumirim	8,47	9,57	463,36	517,8
Teixeiras	7,78	8,19	473,41	612,36
Teófilo Otoni	8,01	7,34	848,07	966,54
Timóteo	5,56	5,67	387,08	417,06
Tiradentes	6,02	8,15	470,88	553,03
Tiros	8,13	9,22	735,23	720,03
Tocantins	6,9	9,4	400,45	435,1
Tocos do Moji	6,11	6,82	705,11	759,08
Toledo	6,53	10,37	615,37	717,6
Tombos	10,9	10,48	646,72	697,06
Três Corações	6,97	7,42	313,34	381,12
Três Marias	6,44	5,51	581,51	695,59

Três Pontas	6,84	7,52	706,56	783,98
Tumiritinga	7,38	7,19	435,74	522,21
Tupaciguara	7,94	8,27	746,84	772,69
Turmalina	5,41	3,88	547,25	638,23
Turvolândia	6,88	7,44	634,03	753,92
Ubá	6,33	6,16	307,2	367,73
Ubaí	6,4	7,34	506,14	503,33
Ubaporanga	6,58	7,28	409,54	347,37
Uberaba	7,22	7,33	727,09	711,25
Uberlândia	5,82	5,78	603,98	710,06
Umburatiba	7,81	9,69	747,47	729,27
Unaí	5,76	5,64	598,16	625,45
União de Minas	7,47	4,31	944,1	872,12
Uruana de Minas	6,97	8,47	1016,73	1087,93
Urucânia	7,73	7,25	609,18	641,07
Urucuaia	2,49	2,26	401,51	481,42
Vargem Alegre	5,94	6,55	306,81	391,28
Vargem Bonita	10,51	8,23	1087,94	1274,33
Vargem Grande do Rio Pardo	4,44	3,41	572,67	607,21
Varginha	6,88	6,96	865,18	967,06
Varjão de Minas	5,75	4,65	686,41	648,56
Várzea da Palma	5,85	5,24	326,14	361,3
Varzelândia	6,15	5,63	485,21	407,66
Vazante	5,32	5,93	719,21	686,03
Verdelândia	6,52	5,37	460,49	449,29
Veredinha	4,02	6,62	459,45	619,6
Veríssimo	7,57	5,42	955,17	978
Vermelho Novo	7,02	6,18	781,84	807,37
Vespasiano	4,71	4,55	552,23	576,56
Viçosa	6,1	6,66	561,21	570,41
Vieiras	9,69	8,64	787,9	727,71
Virgem da Lapa	6,62	6,26	450,16	481,95
Virgínia	6,84	9,45	496,74	495,01
Virginópolis	9,36	7,11	420,47	419,94
Virgolândia	7,71	8,46	615,44	714,03
Visconde do Rio Branco	7,51	7,56	292,96	439,07
Volta Grande	7,62	7,6	726,11	839,03
Wenceslau Braz	7,34	8,5	871,5	834,96

Anexo E: Declaração de revisão linguística

26/12/2022

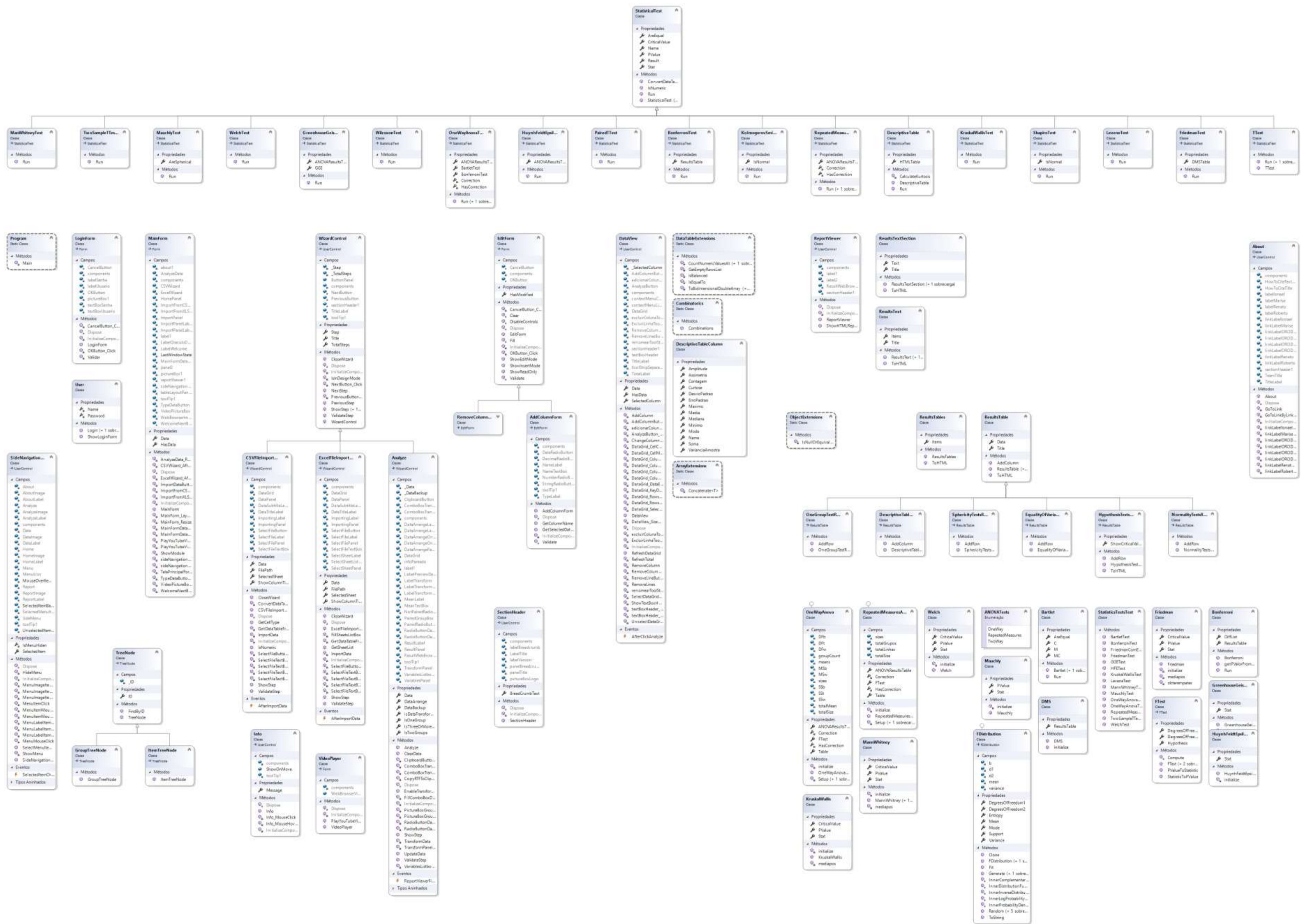
CERTIFICADO DE EDIÇÃO

A quem possa dizer respeito,

Este documento certifica que a tese com o título - Desenvolvimento e Validação de Softwares para Análise Estatística em Ciências da Saúde – teve a língua portuguesa avaliada e corrigida por Gustavo Leal Teixeira, PhD em Linguística e língua portuguesa. Nem o conteúdo da pesquisa nem as intenções do autor foram alterados.



Gustavo Leal Teixeira
Professor Adjunto
UFMG



Apendice B : Artigo

Bibliometrics of CNS & Neurological Disorders – Drug Targets: An International Evolution Along Time

Mirna Rossi Barbosa Medeiros⁶, Ana Carolina de Mello Alves Rodrigues¹, Mariana Rocha, Alves⁷, Roberto Cesar Faria e Silva¹, Lais Francielle Francisca Felício¹, Lara S.F. Carneiro^{8,9}, Daniel Ferreira Fagundes¹, Sérgio Machado¹⁰ and Renato Sobral Monteiro-Junior^{1, 2, 11}

Abstract: Objective and Method: To investigate trends in the scientific evolution of the journal CNS & Neurological Disorders - Drug Targets in the neuroscience scope, we compared the contribution of publications between this journal and others from different geographical regions of the world. To track research output we conducted a bibliometric analysis of neuroscience research based on the SCImago Journal and Country Rank® from 2003 to 2017. Journal rankings were verified according to the following inclusion criteria: journals publishing the neuroscience scope and sub-areas; geographical location and journal trajectory. Additionally, the total number of original, peer-reviewed and conference articles was analyzed using bibliometric tools.

Results: Results showed that Europe, North America and the Middle East have been the greatest contributors of neuroscience publications. Nevertheless, there is a huge discrepancy in the number of magazines per region. Until 2017, Europe was on top with 85 magazines in the neuroscience field. Moreover, research on neuroscience displayed a swift expanding trend, with significant growth in recente years.

⁶ Post-Graduate Programo of Health Sciences, State University of Montes Claros, Montes Claros, Brazil

⁷ Departament of Physical Education, State University of Montes Claros, Montes Claros, Brazil

⁸Research Centre in Sports Sciences, Health Sciences and Human Development, CIDESD, GERON Research Community, Portugal;

⁹University Institute of Maia, ISMAI, Maia, Portugal

¹⁰Universidade Salgado de Oliveira, Niterói, Rio de Janeiro, Brazil;

¹¹Federal Fluminense University, Rio de Janeiro, Brazil;

Conclusion: In spite of CNS & Neurological Disorders - Drug Targets being a recent journal, it is an international journal emphasizing quality and innovations, and it is a hallmark on the scientific production in neuroscience. Research articles on the scope of the potential role of endocannabinoid systems in central appetite control and in obesity management and the potential of minocycline use in schizophrenia are paramount examples of innovation. Final results will help scientific researchers to know the current interests in neuroscience and provide useful information for further investigation and publication strategies.

Keywords: Neuroscience, bibliometric analysis, citation index, journals, trends, geographical regions.

1. INTRODUCTION

Science represents the knowledge obtained by observation, research and phenomena explanation. Scientific research contributes to the advance of a nation on social, economic and political progress. Globalization and the diffusion of Science through internet databases have enabled access to research. Databases made the search for information easier, especially since the 1980s [1-3]. Considered as "Memories of Science", databases organize and provide scientific discoveries to different regions, which enhance scientific knowledge and development worldwide [2-4].

To quantify scientific development, indicators have been created to provide rankings and information about journals and countries [5, 6]. The SCImago Journal & Country Rank is a portal that presents classification parameters of indexed journals through information contained in the Scopus database [5-7]. Different areas of knowledge are sought in SciMago. Over the last years, around the world, Neuroscience research that covers different areas such as biology, humanities, social and math, aiming to understand the central nervous system, the mechanisms involved in neurological diseases and the relationships between brain and mental functions [8, 9, 11]. The dissemination of research in Neuroscience through database results in the propagation of findings about the brain and presents the discoveries, showing the importance of data dissemination and organization [1].

In this scenario, the journal CNS & Neurological Disorders - Drug Targets in the United Arab Emirates discusses neuroscience through topics that include central nervous system

performance and drug use. Despite being a new journal, started in 2003, CNS & Neurological Disorders - Drug Targets needs to be highlighted due to its bibliometric classification, verified in SciMago over the last 23 years, emphasizing the quality of the journal in the neuroscience field.

To improve quality scientific development, researchers must seek knowledge about neuroscience in prestigious journals that are engaged in specific research in their area of expertise. Thus, the present study aims to investigate the scientific evolution of the journal CNS & Neurological Disorders - Drug Targets in the neuroscience field, comparing this journal to others from different areas of the world and the Middle East. has grown [8-10]. Neuroscience is a multidisciplinary Science.

2. MATERIALS AND METHOD

This is a bibliometric study, with retrospective analysis of secondary data. For data collection, a survey of the scientific production of the United Arab Emirates, the Middle East, and other world areas (Europe, Latin and North America, Asia and the Pacific) was conducted in the neuroscience field between 2003 and 2017, which corresponds to the available period through the search portal used since the creation of the journal CNS & Neurological Disorders Drug Targets.

The search process was performed by the SCimago Journal and Country Rank®, a free access portal, developed in 2008, which is configured as a method for the evaluation of scientific journals indexed in the Scopus® database, providing indicators based on received citation numbers.

To verify journals ranking, we used the following inclusion criteria: journals publishing the major area of neurosciences and sub-area of neurosciences (miscellaneous); journals between the years of 2003 and 2017. Regarding the geographical location, journal trajectories were analyzed in the United Arab Emirates, the Middle East region and other regions of the world.

After providing information, the indicators extracted for bibliometric analysis were:

- Total number of citable documents produced, that is, the total number of original, peer-reviewed and conference articles, published by the journal between 2003 and 2017.

- SJR, which corresponds to the impact index of a given journal in a specific year. SJR calculation is based on the average number of citations received in the selected year by the documents published in the journal in the previous three years.

SJR of each region was presented through descriptive values (mean, median, minimum, maximum) in a Boxplot. We used the mean and standard deviation of production of the 7 journals of the Middle East (Central Nervous System Agents in Medicinal Chemistry, CNS and Neurological Disorders - Drug Targets, Computational Intelligence and Neuroscience, Noropsikiyatri Arsivi, Open Neuroscience Journal, Open Pain Journal and Parkinson's Disease) in the analyzed period. Annual production (total docs) and biennial citation percentage of 7 Middle East periodicals from 2003 to 2017 were tabulated. We conducted a linear regression of the evolution of CNS & Neurological Disorders Drug Targets from 2003 to 2017, with the objective of verifying the evolution of the journal according to the annual production. All the variables were analyzed and processed in a Matlab® environment, at a significance level of 5%.

3. RESULTS

Figure (1) presents the publications in Neurosciences from 2003 to 2017 regarding the geographical location. We could observe that Europe, North America and the Middle East have excelled in areas with the highest rates of the scientific development index compared to, Asia, Latin America and Pacific.

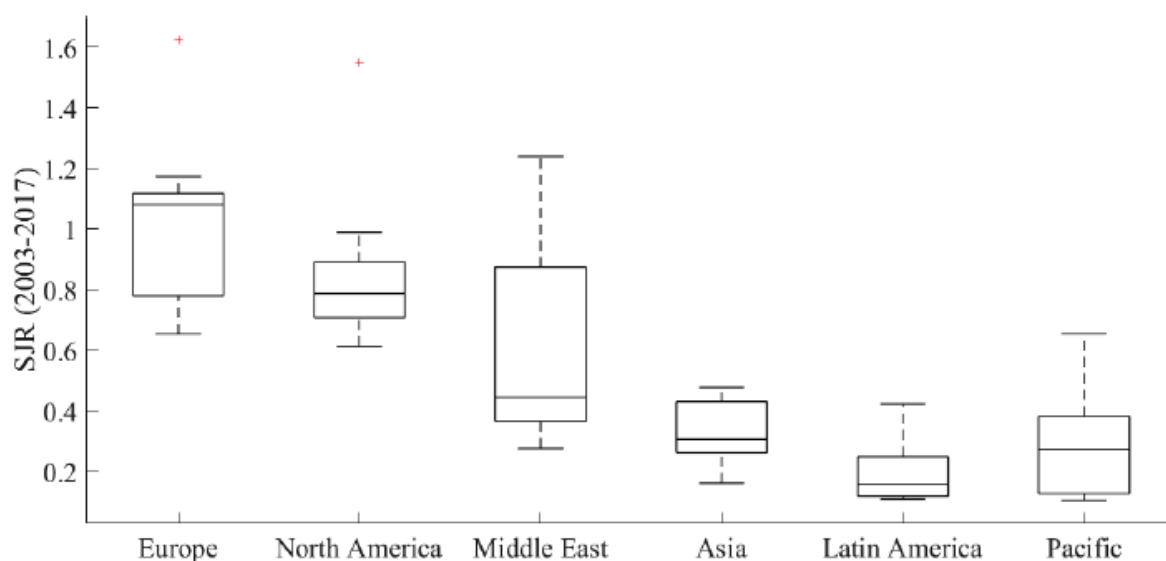


Fig. (1). Boxplot of the scientific development index SciMago (SJR) of Europe, North America, Middle East, Asia, Latin America and Pacific, from 2003 to 2017.

Figure (2) shows the publication mean and standard deviation of each journal from the Middle East from 2003 to 2017. It is possible to observe that the major contributions to the scientific development of the Middle East region in the Neuroscience field have been made by the journal CNS & Neurological Disorders Drug Targets (94.2 ± 49.30), followed by the journal Neuropsikiyatri (77.75 ± 10.93), Computational (70.50 ± 66.03) and Parkinson Disease (67.57 ± 29.23).

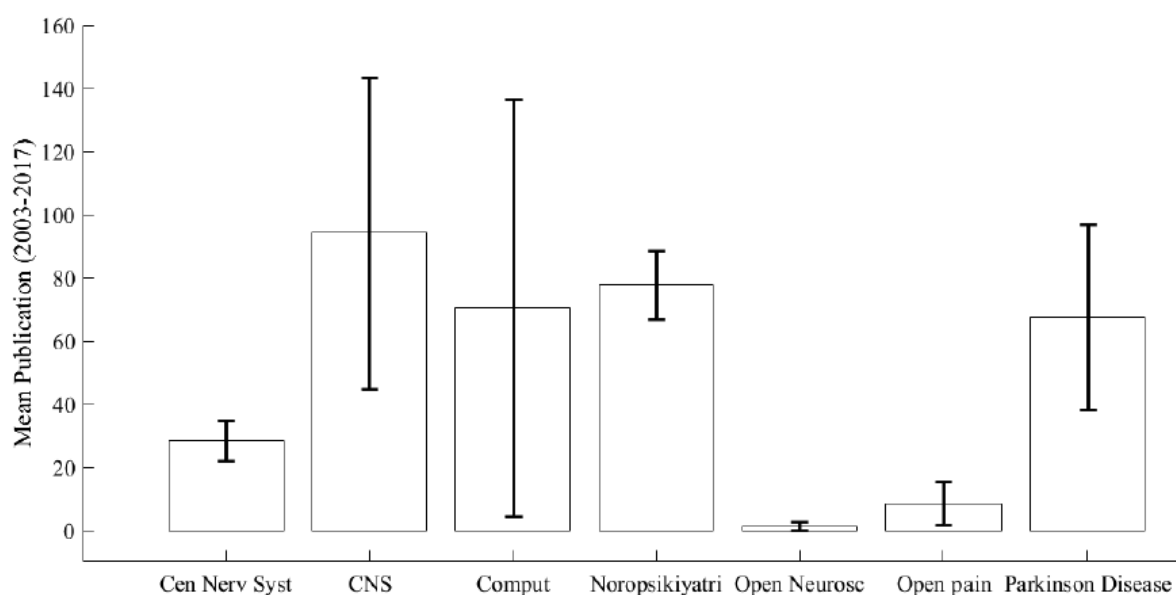


Fig. (2). Mean and standard deviation of production of the Middle East journals from 2003 to 2017. Cen Nerv Syst: Central Nervous System Agents in Medicinal Chemistry; CNS: CNS and Neurological Disorders - Drug Targets; Comput: Computational Intelligence and Neuroscience; Noropsikiyatri: Noropsikiyatri Arsiyi; Open Neurosc: Open Neuroscience Journal; Open Pain: Open Pain Journal; Parkinson Disease: Parkinson's Disease.

Table 1 indicates the annual production and biennial percentage of 7 journals from the Middle East (Central Nervous System Agents in Medicinal Chemistry, CNS and Neurological Disorders - Drug Targets, Computational Intelligence and Neuroscience, Noropsikiyatri Arsivi, Open Neuroscience Journal, Open Pain Journal and Parkinson's Disease), from 2003 to 2017 in the neuroscience area. It is possible to observe that the number of documents produced by CNS Neurol Disord Drug Targets until 2014 was greater compared to the other newspapers. Already in 2016, there was a decrease in the publication number of CNS Neurol Disord Drug Targets when compared to Comput Intell Neurosci. It was also noticed that in 2016 the CNS magazine had a production of 134 and 2.57% of biennial citations against 212 productions and 1.08% of biennial citations of Comput Intell Neurosci, which demonstrates a higher quality in the production of CNS even with less production during that year.

Table 1. Annual production and biennial citation percentage of journals of the Middle East from 2003 to 2017 in the area of Neuroscience.

	Cent Nerv Syst Agents Med Chem		CNS Neurol Disord Drug Targets		Comput Intell Neurosci		Noropsikiyatri Arsivi		Open Neuroscience Journal		Open Pain Journal		Parkinson's Disease	
	Total Docs	% Biennial	Total Docs	% Biennial	Total Docs	% Biennial	Total Docs	% Biennial	Total Docs	% Biennial	Total Docs	% Biennial	Total Docs	% Biennial
2003	30	1.06	39	1.88										
2004	26	1.16	45	4.44										
2005	29	1.21	63	5.1										
2006	20	1.38	55	5.09										
2007	28	0.64	47	4.84										
2008	30	0.5	52	5.36	14	0.86								
2009	35	0.6	53	4.23	20	2.39								
2010	32	1	86	4.01	39	2.31	71	0.18						
2011	35	2.83	100	4.54	37	2.04	58	0.19			3	0.22	88	0.48
2012	32	2.29	124	4.07	33	3.87	71	0.24	2	0.25	4	0.67	84	1.96
2013	23	1.98	140	2.95	20	5.42	88	0.15	3	0.17	22	2.14	26	2.88
2014	15	2.58	196	2.82	68	1.37	82	0.19	3	0.2	11	1.79	30	3.08
2015	25	1.7	164	2.43	137	0.7	92	0.18	0	0.17	6	0.83	103	2.24
2016	41	2.2	134	2.57	212	1.08	83	0.21	0	0	3	0.07	74	1.78
2017	25	1.08	115	2.05	125	1.48	77	0.36	0	0	11	0.25	68	2.30

Abbreviations: Cen Nerv Syst = Central Nervous System Agents in Medicinal Chemistry; CNS Neurol Disord Drug Targets = CNS and Neurological Disorders - Drug Targets; Comput Intell Neurosci = Computational Intelligence and Neuroscience.

Figure (3) shows the correlation between the annual productions of seven journals related to Neuroscience in the Middle East and the period evaluated. When observing the evolution of

the CNS, we can notice that the adjustment line is positive and increasing with $R^2 = 0.712$ ($p < 0.001$).

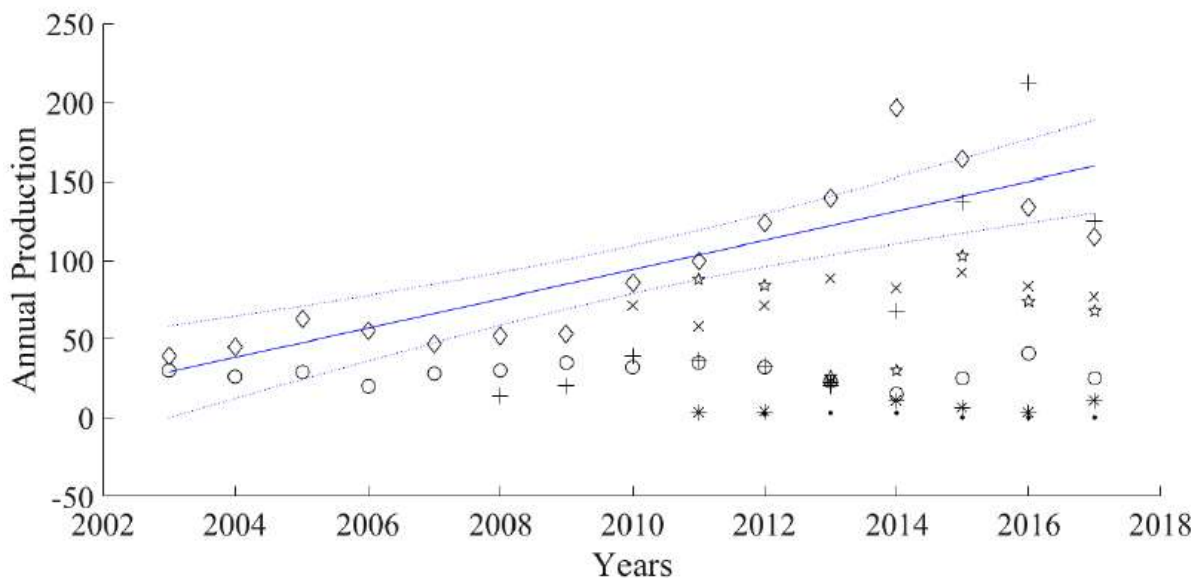


Fig. (3). Linear regression model of the evolution of the CNS in the period 2003 to 2017. \diamond represents the annual production values of the CNS and Neurological Disorders - Drug Targets. The solid blue line represents the linear regression line and the dotted blue line represents the confidence interval of the CNS and Neurological Disorders - Drug Targets. The other symbols represent the annual production of other Middle East journals: \circ represents the annual production values of the Central Nervous System Agents in Medicinal Chemistry; $+$ represents the annual production values of the Computational Intelligence and Neuroscience; \times represents the annual production values of the Noropsikiyatri Arsivi; \cdot represents the annual production values of the Open Neuroscience Journal; $*$ represents the annual production values of the Open Pain Journal and represents the annual production values of the Parkinson's Disease.

4. DISCUSSION

The constant construction of scientific reality demands an ever-widening field of study. With the advancement of technology, nowadays it is possible to obtain new data and new results in the scientific field [3, 4, 12]. Neuroscience emerged as a field to understand biology and human behavior, revealing several previously unknown integrated processes and modifying the perception of certain human processes [12, 13].

The great importance of scientific journals is recognized in collaborating to make decisions about the future and in the dissemination of scientific information [4, 14]. When analyzing the SJR of Europe area, North America, the Middle East, Asia, Latin America and the Pacific, the first three regions are highlighted as obtaining the most citations in neuroscience field between 2003 and 2017. The contribution of Middle East area to the development in

neuroscience is evident since this region occupies the third position in this study field. This worldwide placement has a strong relevance, despite the Human Development Index (degree of economic development and quality of life offered to the population) in some countries of the Middle East is considered very high (e.g. Israel 0.894), but there is great socioeconomic inequality in the region (see HDI of Yemen 0.498) [15]. Thus, according to data provided by the UNDP (2015), the HDI of Middle East in all countries is considerably smaller than in North America and Europe. It is understood that the scientific advance can contribute to the social, economic and political development of Middle East.

Furthermore, the Gross Domestic Product (sum in monetary values of all the final goods and services produced) of a given region can predict the capital investment capacity in the scientific research. Thus, the GDP disparity of the three main regions analyzed in its SJR index is emphasized: Europe,

North America and the Middle East. The World Bank Group (2016) has provided data of the major economies of the three regions referred above, respectively: the United Kingdom, the United States of America and Turkey. Turkey had a GDP three times smaller than the United Kingdom and 21 times smaller than the United States of America, which is a relevant factor when comparing the SciMago Journal Ranks (SJR) rates of the three regions [16].

Another relevant topic in the SJR analysis by world areas is the number of journals that each region has, considering the first three regions with the most citations, Europe, North America and the Middle East, there is a great disparity in the number of circulating magazines. According to the SciMago Journal and Country Rank, Europe had, until 2017, 85 magazines in Neuroscience field, North America 41 magazines and the Middle East only 07. Thereby, it is important to highlight the probable correlation between the high number of journals in Europe and North America areas and their placements in the SJR index. Although Middle East has only 07 journals, it remains the third region with more citations in Neuroscience area.

Over the past 15 years (2003 to 2017), CNS & Neurological Disorders - Drug Targets has been at the top of the biennial percentage citation of Middle East journals for 12 years. In the Neuroscience area, CNS & Neurological Disorders - Drug Targets received the largest average number of publications in the Middle East between 2003 and 2014, increasing its annual production of documents by more than three times. Although in 2013 there has been a decrease in citation number, this could be the result of the emergence of new periodicals circulating in the United Arab Emirates. This information offers the possibility of action to

improve the magazine ranking and CNS & Neurological Disorders - Drug Targets is an influent magazine.

It is noteworthy that in its publications, CNS & Neurological Disorders - Drug Targets brought innovative studies worldwide, such as the endocannabinoids for appetite control and obesity treatment, published in 2006 (edition 3, volume 5, pages 275-292), and the clinical potential of minocycline use in schizophrenia, dated 2008 (edition 4, volume 7, pages 376-381) [17]. Such studies prove the journal has been searching for the scientific advance and its commitment to the unveiling of information that can contribute to the improvement of future reality.

Thus, the present investigation shows the significant growth in recent years and the importance of CNS & Neurological Disorders - Drug Targets journal to the scientific production in Neuroscience.

CONCLUSION

Over the last two decades, the CNS & Neurological Disorders - Drug Targets journal presented linear growth and has been most of the times ahead of the other journals in scientific production. Therefore, it is one of the leading Middle East journals and one of the major contributors to the growth of Neuroscience knowledge worldwide.

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

Not applicable.

HUMAN AND ANIMAL RIGHTS

No Animals/Humans were used for studies that are the basis of this research.

CONSENT FOR PUBLICATION

Not applicable.

CONFLICT OF INTEREST

The authors declare no conflict of interest, financial or otherwise.

ACKNOWLEDGEMENTS

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Fundação de Amparo à

Pesquisa do Estado do Rio de Janeiro (FAPERJ) e do Estado de Minas Gerais (FAPEMG) supported this work.

REFERENCES

- [1] Kötter R. Neuroscience databases: tools for exploring brain structure–function relationships. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*. 2001; 356(1412): 1111-20.
- [2] Sayão LF. Bases de dados: a metáfora da memória científica. *Ci Inf, Brasília*. 2009; 26(3): 314-1001.
- [3] Younies H, Berham B, Smith PC. Perceptions of continuing medical education, professional development, and organizational support in the United Arab Emirates. *J Contin Educ Health Prof*. 2010 Fall; 30(4): 251-6.
- [4] The Arabs' scientific vision. *Nature Materials*. [Editorial]. 2014; 13: 317.
- [5] Colledge L, de Moya-Anegón F, Guerrero-Bote V, et al. SJR and SNIP: two new journal metrics in Elsevier's Scopus. *Serials*. 2010; 23(3): 215-21.
- [6] Grupo S. SCImago journal & country rank: un nuevo portal, dos nuevos rankings. 2007.
- [7] Moed HF. A new journal citation impact measure that compensates for disparities in citation potential among research areas. 2010.
- [8] Abi-Rached JM. From brain to neuro: The Brain Research Association and the making of British neuroscience, 1965–1996. *Journal of the History of the Neurosciences*. 2012; 21(2): 189-213.
- [9] Guimarães TT, Monteiro-Junior RS, Deslandes AC. A evolução da neurociência no Brasil: uma comparação com os países da América Latina nos últimos 16 anos. *Revista Neurociências*. 2014; 22: 359- 64.
- [10] RIEDER CRM. Como anda a pesquisa em neurociência no Brasil? *Revista de Neurociência [serial on the Internet]*. 2014; 22.
- [11] Lent R. *Neurociência da mente e do comportamento*: Grupo Gen-Guanabara Koogan; 2000.
- [12] Gallego Gómez JC. Evolución y neurociencias. *Neurobiología, Filogenia Y Teoría Del Caos. Revista Colombiana de Psiquiatría*. 2004; 33(1).

- [13] Makhoul J, Nakkash RT, El Hajj T, et al. Development and validation of the Arab Youth Mental Health Scale. *Community Ment Health J.* 2011 Jun; 47(3): 331-40.
- [14] Tenopir C, King DW. A importância dos periódicos para o trabalho científico iii. *Revista de Biblioteconomia de Brasília.* 2001; 25(1): 15-26.
- [15] RDH. Human Development Report. 2015 [cited 2017 April]; Program of United Nations for Development (PNUD). Available from: http://hdr.undp.org/sites/default/files/2015_human_development_report.pdf.
- [16] MUNDIAL GB. PIB a precios actuales. <https://datos.bancomundial.org/indicador/NY.GDP.MKTP.CD?start=20032018> [cited 2018 22 of May].
- [17] Benthamscience. CNS & Neurological Disorders – Drug Targets. <https://benthamscience.com/journals/cns-and-neurological-disordersdrug-targets/volume/5/issue/4/2018> [cited 2018 19 of May].

Apendice C : Código Fonte PPGCS Sample Size

Application\autoload.php

```
<?php
```

```
spl_autoload_register(function ($filename) {
```

```
    $file = '..' . DIRECTORY_SEPARATOR . $filename . '.php';
```

```
    if ( DIRECTORY_SEPARATOR === '/' ):
```

```
        $file = str_replace('\\', '/', $file);
```

```
    endif;
```

```
    if ( file_exists($file) ):
```

```
        require $file;
```

```
    else:
```

```
        echo 'Erro ao importar o arquivo!';
```

```
    endif;
```

```
});
```

Application\controllers\Calcular.php

```
<?php
```

```

use Application\core\Controller;

use Application\helpers\Formulas;

use Application\helpers\Form;

class Calcular extends Controller

{

private $form;

private $formulas;

/**

* chama a view index.php da seguinte forma /user/index ou somente /user

* e retorna para a view todos os usuários no banco de dados.

*/

function __construct()

{

$this->form = new Form();

$this->formulas = new Formulas();

}

```

```
public function index()
```

```
{
```

```
    $this->view('home/index');
```

```
}
```

```
public function estimativaparamediaspopulacaoinfinita()
```

```
{
```

```
    //preenche os valores padrão
```

```
    $data['a'] = '95';
```

```
    $data['sd'] = '0.50';
```

```
    $data['e'] = '0.05';
```

```
    //verifica primeiro se houve post
```

```
    if (!empty($_POST)) {
```

```
        //se houve post, valida a entrada
```

```
        $data['a'] = $this->form->validar_campo_numerico($_POST["a"]);
```

```
        $data['sd'] = $this->form->validar_campo_numerico($_POST["sd"]);
```

```
        $data['e'] = $this->form->validar_campo_numerico($_POST["e"]);
```

```
        //calcula z
```

```

$data['z'] = $this->formulas->za($data['a']);

//agora com os valores validados calcula n

$data['n'] = $this->formulas->empi($data['z'], $data['sd'], ($data['e']));

}

//exibe a view com o resultado

$this->view('calcular/empi', $data);

}

public function estimativaparamediaspopulacaofinita()

{

//preenche os valores padrão

$data['a'] = '95';

$data['sd'] = '0.50';

$data['pop'] = '100';

$data['e'] = '0.05';

//verifica primeiro se houve post

if (!empty($_POST)) {

```

```

//se houve post, valida a entrada

$data['a'] = $this->form->validar_campo_numerico($_POST["a"]);

$data['sd'] = $this->form->validar_campo_numerico($_POST["sd"]);

$data['pop'] = $this->form->validar_campo_numerico($_POST["pop"]);

$data['e'] = $this->form->validar_campo_numerico($_POST["e"]);

//calcula z

$data['z'] = $this->formulas->za($data['a']);

//agora com os valores validados calcula n

$data['n'] = $this->formulas->empf($data['z'], $data['sd'], $data['pop'], $data['e']);

}

//exibe a view com o resultado

$this->view('calcular/empf', $data);

}

public function mediaexperimentalcontraumvalorreferenciatesteunilateral()

{

//preenche os valores padrão

```

```
$data['a'] = '95';
```

```
$data['b'] = '84';
```

```
$data['sd'] = '10';
```

```
$data['x0'] = '92.5';
```

```
$data['x1'] = '95';
```

```
//verifica primeiro se houve post
```

```
if (!empty($_POST)) {
```

```
    //se houve post, valida a entrada
```

```
    $data['a'] = $this->form->validar_campo_numerico($_POST["a"]);
```

```
    $data['b'] = $this->form->validar_campo_numerico($_POST["b"]);
```

```
    $data['sd'] = $this->form->validar_campo_numerico($_POST["sd"]);
```

```
    $data['x0'] = $this->form->validar_campo_numerico($_POST["x0"]);
```

```
    $data['x1'] = $this->form->validar_campo_numerico($_POST["x1"]);
```

```
    //calcula za
```

```
    $data['za'] = $this->formulas->za($data['a']);
```

```
    //calcula zb
```

```
    $data['zb'] = $this->formulas->zb($data['b']);
```

```

//agora com os valores validados calcula n

$data['n'] = $this->formulas->mecvr($data['za'], $data['zb'], $data['sd'], $data['x0'],
$data['x1']);

}

//exibe a view com o resultado

$this->view('calcular/mecvr', $data);

}

public function mediaexperimentalcontraumvalorreferenciatestebilateral()

{

//preenche os valores padrão

$data['a'] = '95';

$data['b'] = '84';

$data['sd'] = '10';

$data['x0'] = '92.5';

$data['x1'] = '95';

//verifica primeiro se houve post

```



```

if (!empty($_POST)) {

    //se houve post, valida a entrada

    $data['a'] = $this->form->validar_campo_numerico($_POST["a"]);

    $data['b'] = $this->form->validar_campo_numerico($_POST["b"]);

    $data['sd'] = $this->form->validar_campo_numerico($_POST["sd"]);

    $data['x0'] = $this->form->validar_campo_numerico($_POST["x0"]);

    $data['x1'] = $this->form->validar_campo_numerico($_POST["x1"]);

    //calcula za

    $data['za'] = $this->formulas->zb($data['a']);

    //calcula zb

    $data['zb'] = $this->formulas->zb($data['b']);

    //agora com os valores validados calcula n

    $data['n'] = $this->formulas->mecvr($data['za'], $data['zb'], $data['sd'], $data['x0'],
    $data['x1']);

}

//exibe a view com o resultado

```

```

$this->view('calcular/mecvrb', $data);

}

public function duasm medias amostras pareadas teste unilateral()

{

//preenche os valores padrão

$data['a'] = '95';

$data['b'] = '10';

$data['sd'] = '4';

$data['dif'] = '2';

//verifica primeiro se houve post

if (!empty($_POST)) {

//se houve post, valida a entrada

$data['a'] = $this->form->validar_campo_numerico($_POST["a"]);

$data['b'] = $this->form->validar_campo_numerico($_POST["b"]);

$data['sd'] = $this->form->validar_campo_numerico($_POST["sd"]);

$data['dif'] = $this->form->validar_campo_numerico($_POST["dif"]);

//calcula za

```

```

$data['za'] = $this->formulas->zb($data['a']);

//calcula zb

$data['zb'] = $this->formulas->zb($data['b']);

//agora com os valores validados calcula n

$data['n'] = $this->formulas->dmap($data['za'], $data['zb'], $data['sd'], $data['dif']);
}

//exibe a view com o resultado

$this->view('calcular/dmap', $data);
}

public function duasm medias amostrais pareadas teste bilateral()
{
//preenche os valores padrão

$data['a'] = '95';

$data['b'] = '10';

$data['sd'] = '4';

$data['dif'] = '2';

```

```

//verifica primeiro se houve post

if (!empty($_POST)) {

    //se houve post, valida a entrada

    $data['a'] = $this->form->validar_campo_numerico($_POST["a"]);

    $data['b'] = $this->form->validar_campo_numerico($_POST["b"]);

    $data['sd'] = $this->form->validar_campo_numerico($_POST["sd"]);

    $data['dif'] = $this->form->validar_campo_numerico($_POST["dif"]);

    //calcula za

    $data['za'] = $this->formulas->za($data['a']);

    //calcula zb

    $data['zb'] = $this->formulas->zb($data['b']);

    //agora com os valores validados calcula n

    $data['n'] = $this->formulas->dmap($data['za'], $data['zb'], $data['sd'], $data['dif']);

}

//exibe a view com o resultado

```

```

$this->view('calcular/dmapb', $data);

}

public function duasm medias amostras independentes mesmo tamanhoteste unilateral()
{
    //preenche os valores padrão

    $data['a'] = '95';

    $data['b'] = '80';

    $data['sd1'] = '10';

    $data['sd2'] = '10';

    $data['x1'] = '90';

    $data['x2'] = '95';

    //verifica primeiro se houve post

    if (!empty($_POST)) {

        //se houve post, valida a entrada

        $data['a'] = $this->form->validar_campo_numerico($_POST["a"]);

        $data['b'] = $this->form->validar_campo_numerico($_POST["b"]);

        $data['sd1'] = $this->form->validar_campo_numerico($_POST["sd1"]);

        $data['sd2'] = $this->form->validar_campo_numerico($_POST["sd2"]);
    }
}

```

```

$data['x1'] = $this->form->validar_campo_numerico($_POST["x1"]);

$data['x2'] = $this->form->validar_campo_numerico($_POST["x2"]);

//calcula za

$data['za'] = $this->formulas->zb($data['a']);

//calcula zb

$data['zb'] = $this->formulas->zb($data['b']);

//agora com os valores validados calcula n

$data['n'] = $this->formulas->dmai($data['za'], $data['zb'], $data['sd1'], $data['sd2'],
$data['x1'], $data['x2']);

}

//exibe a view com o resultado

$this->view('calcular/dmai', $data);

}

public function duasm medias amostras independentes mesmo tamanho teste bilateral()

{

```

```

//preenche os valores padrão

$data['a'] = '95';

$data['b'] = '80';

$data['sd1'] = '10';

$data['sd2'] = '10';

$data['x1'] = '90';

$data['x2'] = '95';

//verifica primeiro se houve post

if (!empty($_POST)) {

    //se houve post, valida a entrada

    $data['a'] = $this->form->validar_campo_numerico($_POST["a"]);

    $data['b'] = $this->form->validar_campo_numerico($_POST["b"]);

    $data['sd1'] = $this->form->validar_campo_numerico($_POST["sd1"]);

    $data['sd2'] = $this->form->validar_campo_numerico($_POST["sd2"]);

    $data['x1'] = $this->form->validar_campo_numerico($_POST["x1"]);

    $data['x2'] = $this->form->validar_campo_numerico($_POST["x2"]);

    //calcula za

    $data['za'] = $this->formulas->za($data['a']);

```

```

//calcula zb

$data['zb'] = $this->formulas->zb($data['b']);

//agora com os valores validados calcula n

$data['n'] = $this->formulas->dmai($data['za'], $data['zb'], $data['sd1'], $data['sd2'],
$data['x1'], $data['x2']);

}

//exibe a view com o resultado

$this->view('calcular/dmaib', $data);

}

public function duasm medias amostrais independentes tamanho diferente teste unilateral()
{

//preenche os valores padrão

$data['a'] = '95';

$data['b'] = '80';

$data['sd1'] = '10';

$data['sd2'] = '10';

```



```
$data['x1'] = '90';

$data['x2'] = '95';

$data['k'] = '2';

//verifica primeiro se houve post

if (!empty($_POST)) {

    //se houve post, valida a entrada

    $data['a'] = $this->form->validar_campo_numerico($_POST["a"]);

    $data['b'] = $this->form->validar_campo_numerico($_POST["b"]);

    $data['sd1'] = $this->form->validar_campo_numerico($_POST["sd1"]);

    $data['sd2'] = $this->form->validar_campo_numerico($_POST["sd2"]);

    $data['x1'] = $this->form->validar_campo_numerico($_POST["x1"]);

    $data['x2'] = $this->form->validar_campo_numerico($_POST["x2"]);

    $data['k'] = $this->form->validar_campo_numerico($_POST["k"]);

    //calcula za

    $data['za'] = $this->formulas->zb($data['a']);

    //calcula zb

    $data['zb'] = $this->formulas->zb($data['b']);
```

```

//agora com os valores validados calcula n

$ret = array();

$ret = $this->formulas->dmaitd($data['za'], $data['zb'], $data['sd1'], $data['sd2'],
$data['x1'], $data['x2'], $data['k']);

$data['n1'] = $ret['n1'];

$data['n2'] = $ret['n2'];

}

//exibe a view com o resultado

$this->view('calcular/dmaitd', $data);

}

public function duasm medias amostras independentes tamanho diferente teste bilateral()
{

//preenche os valores padrão

$data['a'] = '95';

$data['b'] = '80';

$data['sd1'] = '10';

```

```

$data['sd2'] = '10';

$data['x1'] = '90';

$data['x2'] = '95';

$data['k'] = '5';

//verifica primeiro se houve post

if (!empty($_POST)) {

    //se houve post, valida a entrada

    $data['a'] = $this->form->validar_campo_numerico($_POST["a"]);

    $data['b'] = $this->form->validar_campo_numerico($_POST["b"]);

    $data['sd1'] = $this->form->validar_campo_numerico($_POST["sd1"]);

    $data['sd2'] = $this->form->validar_campo_numerico($_POST["sd2"]);

    $data['x1'] = $this->form->validar_campo_numerico($_POST["x1"]);

    $data['x2'] = $this->form->validar_campo_numerico($_POST["x2"]);

    $data['k'] = $this->form->validar_campo_numerico($_POST["k"]);

    //calcula za

    $data['za'] = $this->formulas->za($data['a']);

    //calcula zb

```

```

$data['zb'] = $this->formulas->zb($data['b']);

//agora com os valores validados calcula n

$ret = array();

$ret = $this->formulas->dmaitd($data['za'], $data['zb'], $data['sd1'], $data['sd2'],
$data['x1'], $data['x2'], $data['k']);

$data['n1'] = $ret['n1'];

$data['n2'] = $ret['n2'];

}

//exibe a view com o resultado

$this->view('calcular/dmaitdb', $data);

}

public function estimativasparaproporcoespopulacaoinfinita()

{

//preenche os valores padrão

$data['a'] = '95';

$data['p'] = '33';

```

```

$data['e'] = '0.05';

//verifica primeiro se houve post

if (!empty($_POST)) {

    //se houve post, valida a entrada

    $data['a'] = $this->form->validar_campo_numerico($_POST["a"]);

    $data['p'] = $this->form->validar_campo_numerico($_POST["p"]);

    $data['e'] = $this->form->validar_campo_numerico($_POST["e"]);

    //calcula za

    $data['z'] = $this->formulas->za($data['a']);

    //agora com os valores validados calcula n

    $data['n'] = $this->formulas->eppi($data['z'], $data['p']/100, $data['e']);

}

//exibe a view com o resultado

$this->view('calcular/eppi', $data);

}

```

```

public function estimativasparaproporcoespopulacaofinita()

{

//preenche os valores padrão

$data['a'] = '95';

$data['p'] = '50';

$data['e'] = '0.05';

$data['pop'] = '2300';

//verifica primeiro se houve post

if (!empty($_POST)) {

//se houve post, valida a entrada

$data['a'] = $this->form->validar_campo_numerico($_POST["a"]);

$data['p'] = $this->form->validar_campo_numerico($_POST["p"]);

$data['e'] = $this->form->validar_campo_numerico($_POST["e"]);

$data['pop'] = $this->form->validar_campo_numerico($_POST["pop"]);

//calcula za

$data['z'] = $this->formulas->za($data['a']);

//agora com os valores validados calcula n

```

```

    $data['n'] = $this->formulas->epppf($data['z'], $data['p']/100, $data['e'], $data['pop']);

}

//exibe a view com o resultado

$this->view('calcular/epppf', $data);

}

public function estimardiferencaentreduasproporcoes()

{

    //preenche os valores padrão

    $data['a'] = '95';

    $data['p1'] = '40';

    $data['p2'] = '28';

    $data['d'] = '0.1';

    //verifica primeiro se houve post

    if (!empty($_POST)) {

        //se houve post, valida a entrada

        $data['a'] = $this->form->validar_campo_numerico($_POST["a"]);

        $data['p1'] = $this->form->validar_campo_numerico($_POST["p1"]);
    }
}

```

```

$data['p2'] = $this->form->validar_campo_numerico($_POST["p2"]);

$data['d'] = $this->form->validar_campo_numerico($_POST["d"]);

//calcula za

$data['z'] = $this->formulas->za($data['a']);

//agora com os valores validados calcula n

$data['n'] = $this->formulas->ededp($data['z'], $data['p1']/100, $data['p2']/100, $data['d']);

}

//exibe a view com o resultado

$this->view('calcular/ededp', $data);

}

public function compararduasproporcoes()

{

//preenche os valores padrão

$data['a'] = '95';

$data['b'] = '80';

$data['p1'] = '40';

```



```
$data['p2'] = '28';

//verifica primeiro se houve post

if (!empty($_POST)) {

    //se houve post, valida a entrada

    $data['a'] = $this->form->validar_campo_numerico($_POST["a"]);

    $data['b'] = $this->form->validar_campo_numerico($_POST["b"]);

    $data['p1'] = $this->form->validar_campo_numerico($_POST["p1"]);

    $data['p2'] = $this->form->validar_campo_numerico($_POST["p2"]);

    //calcula za

    $data['za'] = $this->formulas->za($data['a']);

    //calcula zb

    $data['zb'] = $this->formulas->zb($data['b']);

    //agora com os valores validados calcula n

    $data['n'] = $this->formulas->cdp($data['za'], $data['zb'], $data['p1']/100, $data['p2']/100);

}
```

```

//exibe a view com o resultado

$this->view('calcular/cdp', $data);

}

public function calculoz()

{

//preenche os valores padrão

$data['a'] = '0.05';

$data['b'] = '0.8';

//verifica primeiro se houve post

if (!empty($_POST)) {

//se houve post, valida a entrada

$data['a'] = $this->form->validar_campo_numerico($_POST["a"]);

$data['b'] = $this->form->validar_campo_numerico($_POST["b"]);

//agora com os valores validados calcula za/2 e zb

$data['za'] = abs($this->formulas->invnormp($data['a']/2));

$data['zb'] = abs($this->formulas->invnormp(1 - $data['b']));

}

```

```
//exibe a view com o resultado

$this->view('calcular/cz', $data);

}

}
```

Application\controllers\Home.php

```
<?php
```

```
use Application\core\Controller;
```

```
class Home extends Controller
```

```
{
```

```
/*
```

```
* chama a view index.php do /home ou somente /
```

```
*/
```

```
public function index()
```

```
{
```

```
$this->view('home/index');
```

```
}
```

```
}
```

Application\controllers\Sobre.php

```
<?php
```

```
use Application\core\Controller;
```

```
class Sobre extends Controller
```

```
{
```

```
    public function index()
```

```
    {
```

```
        $this->view('sobre/index');
```

```
    }
```

```
}
```

Application\core\App.php

```
<?php
```

```
namespace Application\core;
```

```
/**
```

```
* Esta classe é responsável por obter da URL o controller, método (ação) e os parâmetros
```

```
* e verificar a existência dos mesmo.
```

```
*/
```

```
class App
```

```
{
```

```
    protected $controller = 'Home';
```

```
    protected $method = 'index';
```

```
    protected $page404 = false;
```

```
    protected $params = [];
```

```
    // Método construtor
```

```
    public function __construct()
```

```
    {
```

```
        $URL_ARRAY = $this->parseUrl();
```

```
        $this->getControllerFromUrl($URL_ARRAY);
```

```

$this->getMethodFromUrl($URL_ARRAY);

$this->getParamsFromUrl($URL_ARRAY);

// chama um método de uma classe passando os parâmetros

call_user_func_array([$this->controller, $this->method], $this->params);

}

/**
 * Este método pega as informações da URL (após o domínio do site) e retorna esses dados
 *
 * @return array
 */

private function parseUrl()

{

    $REQUEST_URI = explode('/', substr(filter_input(INPUT_SERVER, 'REQUEST_URI'),
1));

    unset($REQUEST_URI[current(array_keys($REQUEST_URI))]); //TEMPORÒIO:
remove o ppgcssamplesize.com.br OU paineiseletricos.com.br

    unset($REQUEST_URI[current(array_keys($REQUEST_URI))]); //TEMPORÒIO:
remove o public

```

```

$REQUEST_URI = array_values($REQUEST_URI); //TEMPORÒIO: reseta o array

return $REQUEST_URI;

}

/**

* Este método verifica se o array informado possui dados na posição 0 (controlador)

* caso exista, verifica se existe um arquivo com aquele nome no diretório
Application/controllers

* e instancia um objeto contido no arquivo, caso contrário a variável $page404 recebe true.

*

* @param array $url Array contendo informações ou não do controlador, método e
parâmetros

*/

private function getControllerFromUrl($url)

{

if ( !empty($url[0]) && isset($url[0]) ) {

if ( file_exists('../Application/controllers/' . ucfirst($url[0]) . '.php') ) {

$this->controller = ucfirst($url[0]);

} else {

$this->page404 = true;

}

}

```

```

}

require '../Application/controllers/' . $this->controller . '.php';

$this->controller = new $this->controller();

}

/**
 * Este método verifica se o array informado possui dados na posição 1 (método)
 * caso exista, verifica se o método existe naquele determinado controlador
 * e atribui a variável $method da classe.
 *
 * @param array $url Array contendo informações ou não do controlador, método e
parâmetros
 */

private function getMethodFromUrl($url)

{

if ( !empty($url[1]) && isset($url[1]) ) {

$url[1] = str_replace('-', '', $url[1]); //retira os traços da url

if ( method_exists($this->controller, $url[1]) && !$this->page404 ) {

$this->method = $url[1];

```



```

    } else {

        // caso a classe ou o método informado não exista, o método pageNotFound

        // do Controller é chamado.

        $this->method = 'pageNotFound';

    }

}

}

}

/**

* Este método verifica se o array informador possui a quantidade de elementos maior que 2

* ($url[0] é o controller e $url[1] o método/ação a executar), caso seja, é atribuído

* a variável $params da classe um novo array a partir da posição 2 do $url

*

* @param array $url Array contendo informações ou não do controlador, método e
parâmetros

*/

private function getParamsFromUrl($url)

{

    if (count($url) > 2) {

        $this->params = array_slice($url, 2);

```

```
}  
  
}  
  
}
```

Application\core\Controller.php

```
<?php
```

```
namespace Application\core;
```

```
/**
```

```
* Esta classe é responsável por instanciar um model e chamar a view correta
```

```
* passando os dados que serão usados.
```

```
*/
```

```
class Controller
```

```
{
```

```
/**
```

```
* Este método é responsável por chamar uuma determinada view (página).
```

```
*
```

```
* @param string $view A view que será chamada (ou requerida)
```

```
* @param array $data São os dados que serão exibido na view
```

```
*/
```

```
public function view(string $view, $data = [])
```

```
{
```

```
    require '../Application/views/' . $view . '.php';
```

```
}
```

```
/**
```

```
* Este método é herdado para todas as classes filhas que o chamaram quando
```

```
* o método ou classe informada pelo usuário não forem encontrados.
```

```
*/
```

```
public function pageNotFound()
```

```
{
```

```
    $this->view('erro404');
```

```
}
```

```
}
```

Application\helpers\Form.php

```
<?php
```

```
namespace Application\helpers;

class Form
{
    public function validar_campo_numerico($campo)
    {
        if (isset($campo)) {
            if (is_numeric(htmlspecialchars($campo))) {
                return $campo;
            }
        }
        return 0;
    }
}
?>
```

Application\helpers\Formulas.php

```
<?php
```

```
namespace Application\helpers;
```

```

class Formulas

{

//estimativa para medias - população infinita

function empi ($z, $s, $e)

{

return ceil((( $Z^2$ )*( $S^2$ ))/(( $e^2$ )));

}

//estimativa para médias - população finita

function empf ($z, $s, $n, $e)

{

return ceil(( $Z^2 * S^2 * n$ )/((( $n-1$ )* $e^2$ )+( $Z^2 * S^2$ )));

}

//Média experimental contra um valor referência

function mecvr($za, $zb, $sd, $x0, $x1)

{

return ceil(((( $Za+Zb$ )*( $Sd$ ))/( $x1-x0$ ))**2);

}

```

```
//Duas Médias amostrais pareadas
```

```
function dmap($za, $zb, $sd, $dif)
```

```
{
```

```
    return ceil((2*($sd**2)*($za+$zb)**2)/$dif**2);
```

```
}
```

```
//Duas Médias amostrais independentes
```

```
function dmai($za, $zb, $sd1, $sd2, $x1, $x2)
```

```
{
```

```
    return ceil(((($sd1**2+$sd2**2)*($za+$zb)**2)/($x2-$x1)**2);
```

```
}
```

```
//Duas médias amostrais independentes com tamanho diferente
```

```
function dmaitd($za, $zb, $sd1, $sd2, $x1, $x2, $k)
```

```
{
```

```
    $ret = array();
```

```
    $ret['n1'] = ceil(((($sd1**2+($sd2**2/$k))*($za+$zb)**2)/($x2-$x1)**2);
```

```
    $ret['n2'] = ceil(((($k*$sd1**2+$sd2**2)*($za+$zb)**2)/($x2-$x1)**2);
```

```
    return $ret;
```

```
}
```

```
//Estimativas para Proporções - População Infinita
```

```
function eppi($z, $p, $e)
```

```
{
```

```
    return ceil((((z**2)*p*(1-p))/(e**2));
```

```
}
```

```
//Estimativas para Proporções - População Finita
```

```
function epppf($z, $p, $e, $n)
```

```
{
```

```
    return ceil((((z**2)*n*p*(1-p))/(e**2*(n-1)+z**2*p*(1-p)));
```

```
}
```

```
//Estimar diferença entre duas proporções (intervalo de confiança da diferença)
```

```
function ededp($z, $p1, $p2, $d)
```

```
{
```

```
    return ceil((((z**2)*((p1*(1-p1)+((p2*(1-p2)))))/(d**2))));
```

```
}
```

```
//Comparar duas proporções (uma experimental com uma referência)
```

```
function cdp($za, $zb, $p1, $p2)
```

```
{
```

```
    return ceil((((($za*sqrt($p1*(1-$p1))+($zb*sqrt($p2*(1-$p2))))))**2)/($p1-$p2)**2);
```

```
}
```

```
//calcula za/2 a partir de a
```

```
function za($a)
```

```
{
```

```
    return abs($this->invnormp((1 - $a/100)/2));
```

```
}
```

```
//calcula zb a partir de b
```

```
function zb($b)
```

```
{
```

```
    return abs($this->invnormp(1 - $b/100));
```

```
}
```

```
//retorna z para uma dada probabilidade (o inverso da distribuição normal cumulativa  
padrão)
```


//a distribuição tem a média = 0 e sd = 1

```
function invnormp($probability) {
```

```
    $a1 = -39.6968302866538;
```

```
    $a2 = 220.946098424521;
```

```
    $a3 = -275.928510446969;
```

```
    $a4 = 138.357751867269;
```

```
    $a5 = -30.6647980661472;
```

```
    $a6 = 2.50662827745924;
```

```
    $b1 = -54.4760987982241;
```

```
    $b2 = 161.585836858041;
```

```
    $b3 = -155.698979859887;
```

```
    $b4 = 66.8013118877197;
```

```
    $b5 = -13.2806815528857;
```

```
    $c1 = -7.78489400243029E-03;
```

```
    $c2 = -0.322396458041136;
```

```
    $c3 = -2.40075827716184;
```

```
    $c4 = -2.54973253934373;
```

```
    $c5 = 4.37466414146497;
```

`$c6 = 2.93816398269878;`

`$d1 = 7.78469570904146E-03;`

`$d2 = 0.32246712907004;`

`$d3 = 2.445134137143;`

`$d4 = 3.75440866190742;`

`$p_low = 0.02425;`

`$p_high = 1 - $p_low;`

`$q = 0;`

`$r = 0;`

`$normSInv = 0;`

`if ($probability < 0 ||`

`$probability > 1)`

`{`

`throw new \Exception("normSInv: Argument out of range.");`

`} else if ($probability < $p_low) {`

`$q = sqrt(-2 * log($probability));`

```

$normSInv = (((($c1 * $q + $c2) * $q + $c3) * $q + $c4) * $q + $c5) * $q + $c6) /
(((($d1 * $q + $d2) * $q + $d3) * $q + $d4) * $q + 1);

```

```

} else if ($probability <= $p_high) {

```

```

    $q = $probability - 0.5;

```

```

    $r = $q * $q;

```

```

    $normSInv = (((($a1 * $r + $a2) * $r + $a3) * $r + $a4) * $r + $a5) * $r + $a6) *
    $q / (((($b1 * $r + $b2) * $r + $b3) * $r + $b4) * $r + $b5) * $r + 1);

```

```

} else {

```

```

    $q = sqrt(-2 * log(1 - $probability));

```

```

    $normSInv = -((((($c1 * $q + $c2) * $q + $c3) * $q + $c4) * $q + $c5) * $q + $c6)
/(((($d1 * $q + $d2) * $q + $d3) * $q + $d4) * $q + 1);

```

```

}

```

```

return round($normSInv,2);

```

```

}

```

```

}

```

?>

Application\views\erro404.php

```
<div class="container">
```

```
<div class="row">
```

```
<div class="col-12 text-center" style="margin-top:200px">
```

```

```

```
</div>
```

```
</div>
```

```
</div>
```

Application\views\calcular\cdp.php

```
<div class="content">
```

```
<h2>Comparar duas proporções (uma experimental com uma referência)</h2>
```

```
<form class="pure-form pure-form-stacked" action="comparar-duas-proporcoes"
method="post" >
```

```
<fieldset>
```

```
<legend>Informe os parâmetros</legend>
```

```
<img src= "../assets/img/cdp.png" alt="Fórmula para o cálculo de n." class="pure-img">
```

```
<label for="a">Nível de Confiança (1-a):</label>
```

```
<div class="pure-g">
```

```
    <input type="number" id="a" name="a" class="pure-u-sm-1-6 showTooltip"
    title="Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo o
    mais comum 95%. Quanto maior o nível de confiança, maior a amostra."
    placeholder="Informe o nível de confiança." step="0.01" min="0" max="9999999"
    value=<?php echo $data['a']; ?> required />
```

```
        <span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['za']))
```

```
{
```

```
    echo '<label for="za">Valor Z (Nível de Confiança):</label>';
```

```
    echo '<input type="number" id="za" name="za" value="" . $data['za'] . "" readonly />';
```

```
}
```

```
?>
```

```
<label for="b">Poder do teste ( $\beta$ ):</label>
```

```
<div class="pure-g">
```

```
  <input type="number" id="b" name="b" class="pure-u-sm-1-6 showTooltip" title="É a
  probabilidade de se identificar a diferença que realmente existe, geralmente utiliza-se o poder
  do teste de 80%." placeholder="Informe o poder do teste." step="0.01" min="0"
  max="9999999" value=<?php echo $data['b']; ?> required />
```

```
  <span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['zb']))
```

```
{
```

```
  echo '<label for="zb">Valor Z (Poder do Teste):</label>';
```

```
  echo '<input type="number" id="zb" name="zb" value="" . $data['zb'] . "" readonly />';
```

```
}
```

```
?>
```

```
<label for="p1">Prevalência Referência (Po):</label>
```

```
<div class="pure-g">
```

```
<input type="number" id="p1" name="p1" class="showTooltip" title="Percentual do
evento que se pretende estudar na população de referência." placeholder="Informe a
prevalência referência." step="0.01" min="0" max="9999999" value=<?php echo $data['p1'];
?> required />
```

```
<span class="percent">%</span>
```

```
</div>
```

```
<label for="p2">Prevalência Experimental (Pa):</label>
```

```
<div class="pure-g">
```

```
<input type="number" id="p2" name="p2" class="showTooltip" title="Percentual
estimado do evento que se pretende estudar (experimento), pode-se fazer um estudo piloto ou
consultar na literatura." placeholder="Informe a prevalência experimental." step="0.01"
min="0" max="9999999" value=<?php echo $data['p2']; ?> required />
```

```
<span class="percent">%</span>
```

```
</div>
```

```
<button type="submit" class="pure-button pure-button-primary">Calcular!</button>
```

```
</fieldset>
```

```
<?php
```

```
if (isset($data['n']))
```

```
{
```

```

        echo '<fieldset><legend>Resultado:</legend>';

        echo '<label for="n">Tamanho da amostra (n):</label>';

        echo '<input type="number" id="n" name="n" value="" . $data['n'] . "" readonly />';

        echo '</fieldset>';

    }

?>

</form>

</div>

Application\views\calcular\cz.php

<div class="content">

<h2>Cálculo do  $Z_{\alpha/2}$  e  $Z_{1-\beta}$ </h2>

<form class="pure-form pure-form-stacked" action="calculo-z" method="post" >

<fieldset>

<legend>Informe os parâmetros</legend>

<label for="a">Nível Confiança (a):</label>

```



```
<input type="text" id="a" name="a" placeholder="Informe o nível de confiança."
value=<?php echo $data['a']; ?> />
```

```
<label for="b">Poder do Teste ( $\beta$ ):</label>
```

```
<input type="text" id="b" name="b" placeholder="Informe o poder do teste."
value=<?php echo $data['b']; ?> />
```

```
<button type="submit" class="pure-button pure-button-primary">Calcular!</button>
```

```
</fieldset>
```

```
<?php
```

```
if (isset($data['za']))
```

```
{
```

```
    echo '<fieldset><legend>Resultado:</legend>';
```

```
    echo '<label for="za">Valor Z (nível Confiança):</label>';
```

```
    echo '<input type="text" id="za" name="za" value="" . $data['za'] . ""/>';
```

```
    echo '<label for="zb">Valor Z (Poder do teste):</label>';
```

```
    echo '<input type="text" id="zb" name="zb" value="" . $data['zb'] . ""/>';
```

```
    echo '</fieldset>';
```

```
}
```

```
?>
```

```
</form>
```

```
</div>
```

```
Application\views\calcular\dmai.php
```

```
<div class="content">
```

```
<h2>Duas Médias amostrais independentes de mesmo tamanho (Teste Unilateral)</h2>
```

```
<form class="pure-form pure-form-stacked" action="duas-medias-amostrais-independentes-  
mesmo-tamanho-teste-unilateral" method="post" >
```

```
<fieldset>
```

```
<legend>Informe os parâmetros</legend>
```

```
<img src= "../assets/img/dmai.png" alt="Fórmula para o cálculo de n." class="pure-  
img">
```

```
<label for="a">Nível de Confiança (1-a):</label>
```

```
<div class="pure-g">
```

```
<input type="number" id="a" name="a" class="pure-u-sm-1-6 showTooltip"  
title="Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo o  
mais comum 95%. Quanto maior o nível de confiança, maior a amostra."
```

```
placeholder="Informe o nível de confiança." step="0.01" min="0" max="9999999"
value=<?php echo $data['a']; ?> required />
```

```
<span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['za']))
```

```
{
```

```
echo '<label for="za">Valor Z (Nível de Confiança):</label>';
```

```
echo '<input type="number" id="za" name="za" value="" . $data['za'] . "" readonly />';
```

```
}
```

```
?>
```

```
<label for="b">Poder do teste ( $\beta$ ):</label>
```

```
<div class="pure-g">
```

```
<input type="number" id="b" name="b" class="pure-u-sm-1-6 showTooltip" title="É a
probabilidade de se identificar a diferença que realmente existe, geralmente utiliza-se o poder
do teste de 80%." placeholder="Informe o poder do teste." step="0.01" min="0"
max="9999999" value=<?php echo $data['b']; ?> required />
```

```
<span class="percent">%</span>
```

```
</div>
```

```

<?php

if (isset($data['zb']))

{

    echo '<label for="zb">Valor Z (Poder do Teste):</label>';

    echo '<input type="number" id="zb" name="zb" value="" . $data['zb'] . "" readonly />';

}

?>

```

<label for="sd1">Desvio padrão pop. 1 (o1):</label>

<input type="number" id="sd1" name="sd1" class="showTooltip" title="Corresponde ao desvio padrão da população 1, pode ser obtido em uma amostra piloto ou na literatura." placeholder="Informe o desvio padrão da pop. 1." step="0.01" min="0" max="9999999" value=<?php echo \$data['sd1']; ?> />

<label for="sd2">Desvio padrão pop. 2 (o2):</label>

<input type="number" id="sd2" name="sd2" class="showTooltip" title="Corresponde ao desvio padrão da população 2, pode ser obtido em uma amostra piloto ou na literatura." placeholder="Informe o desvio padrão da pop. 2." step="0.01" min="0" max="9999999" value=<?php echo \$data['sd2']; ?> />

<label for="x1">Média pop. 1 (μ_1):</label>

```
<input type="number" id="x1" name="x1" class="showTooltip" title="Corresponde a
média da população 1, pode ser obtido em uma amostra piloto ou na literatura."
placeholder="Informe a média da pop. 1." step="0.01" min="0" max="9999999"
value=<?php echo $data['x1']; ?> />
```

```
<label for="x2">Média pop. 2 ( $\mu_0$ ):</label>
```

```
<input type="number" id="x2" name="x2" class="showTooltip" title="Corresponde a
média da população 2, pode ser obtido em uma amostra piloto ou na literatura."
placeholder="Informe a média da pop. 2." step="0.01" min="0" max="9999999"
value=<?php echo $data['x2']; ?> />
```

```
<button type="submit" class="pure-button pure-button-primary">Calcular!</button>
```

```
</fieldset>
```

```
<?php
```

```
if (isset($data['n']))
```

```
{
```

```
    echo '<fieldset><legend>Resultado:</legend>';
```

```
    echo '<label for="n">Tamanho da amostra (n):</label>';
```

```
    echo '<input type="text" id="n" name="n" value="' . $data['n'] . '" readonly />';
```

```
    echo '</fieldset>';
```

```
}
```

?>

</form>

</div>

Application\views\calcular\dmaib.php

<div class="content">

<h2>Duas Médias amostrais independentes de mesmo tamanho (Teste Bilateral)</h2>

<form class="pure-form pure-form-stacked" action="duas-medias-amostrais-independentes-mesmo-tamanho-teste-bilateral" method="post" >

<fieldset>

<legend>Informe os parâmetros</legend>

<label for="a">Nível de Confiança (1-a):</label>

<div class="pure-g">

<input type="number" id="a" name="a" class="pure-u-sm-1-6 showTooltip" title="Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo o

mais comum 95%. Quanto maior o nível de confiança, maior a amostra."
placeholder="Informe o nível de confiança." step="0.01" min="0" max="9999999"
value=<?php echo \$data['a']; ?> required />

%

</div>

<?php

if (isset(\$data['za']))

{

echo '<label for="za">Valor Z (Nível de Confiança):</label>';

echo '<input type="number" id="za" name="za" value="" . \$data['za'] . "" readonly />';

}

?>

<label for="b">Poder do teste (β):</label>

<div class="pure-g">

<input type="number" id="b" name="b" class="pure-u-sm-1-6 showTooltip" title="É a probabilidade de se identificar a diferença que realmente existe, geralmente utiliza-se o poder do teste de 80%." placeholder="Informe o poder do teste." step="0.01" min="0" max="9999999" value=<?php echo \$data['b']; ?> required />

%

</div>

```

<?php

if (isset($data['zb']))

{

    echo '<label for="zb">Valor Z (Poder do Teste):</label>';

    echo '<input type="number" id="zb" name="zb" value="" . $data['zb'] . "" readonly />';

}

?>

```

<label for="sd1">Desvio padrão pop. 1 (o1):</label>

<input type="number" id="sd1" name="sd1" class="showTooltip" title="Corresponde ao desvio padrão da população 1, pode ser obtido em uma amostra piloto ou na literatura." placeholder="Informe o desvio padrão da pop. 1." step="0.01" min="0" max="9999999" value=<?php echo \$data['sd1']; ?> />

<label for="sd2">Desvio padrão pop. 2 (o2):</label>

<input type="number" id="sd2" name="sd2" class="showTooltip" title="Corresponde ao desvio padrão da população 2, pode ser obtido em uma amostra piloto ou na literatura." placeholder="Informe o desvio padrão da pop. 2." step="0.01" min="0" max="9999999" value=<?php echo \$data['sd2']; ?> />

<label for="x1">Média pop. 1 (μ_1):</label>


```
<input type="number" id="x1" name="x1" class="showTooltip" title="Corresponde a
média da população 1, pode ser obtido em uma amostra piloto ou na literatura."
placeholder="Informe a média da pop. 1." step="0.01" min="0" max="9999999"
value=<?php echo $data['x1']; ?> />
```

```
<label for="x2">Média pop. 2 ( $\mu_0$ ):</label>
```

```
<input type="number" id="x2" name="x2" class="showTooltip" title="Corresponde a
média da população 2, pode ser obtido em uma amostra piloto ou na literatura."
placeholder="Informe a média da pop. 2." step="0.01" min="0" max="9999999"
value=<?php echo $data['x2']; ?> />
```

```
<button type="submit" class="pure-button pure-button-primary">Calcular!</button>
```

```
</fieldset>
```

```
<?php
```

```
if (isset($data['n']))
```

```
{
```

```
    echo '<fieldset><legend>Resultado:</legend>';
```

```
    echo '<label for="n">Tamanho da amostra (n):</label>';
```

```
    echo '<input type="text" id="n" name="n" value="' . $data['n'] . '" readonly />';
```

```
    echo '</fieldset>';
```

```
}
```

?>

</form>

</div>

Application\views\calcular\dmaind.php

<div class="content">

<h2>Duas Médias amostrais independentes de tamanhos diferentes (Teste Unilateral)</h2>

<form class="pure-form pure-form-stacked" action="duas-medias-amostrais-independentes-tamanho-diferente-teste-unilateral" method="post" >

<fieldset>

<legend>Informe os parâmetros</legend>

<label for="a">Nível de Confiança (1-a):</label>

<div class="pure-g">

<input type="number" id="a" name="a" class="pure-u-sm-1-6 showTooltip" title="Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo o

mais comum 95%. Quanto maior o nível de confiança, maior a amostra."
placeholder="Informe o nível de confiança." step="0.01" min="0" max="9999999"
value=<?php echo \$data['a']; ?> required />

%

</div>

<?php

if (isset(\$data['za']))

{

echo '<label for="za">Valor Z (Nível de Confiança):</label>';

echo '<input type="number" id="za" name="za" value="" . \$data['za'] . "" readonly />';

}

?>

<label for="b">Poder do teste (β):</label>

<div class="pure-g">

<input type="number" id="b" name="b" class="pure-u-sm-1-6 showTooltip" title="É a probabilidade de se identificar a diferença que realmente existe, geralmente utiliza-se o poder do teste de 80%." placeholder="Informe o poder do teste." step="0.01" min="0" max="9999999" value=<?php echo \$data['b']; ?> required />

%

</div>

```

<?php

if (isset($data['zb']))

{

    echo '<label for="zb">Valor Z (Poder do Teste):</label>';

    echo '<input type="number" id="zb" name="zb" value="" . $data['zb'] . "" readonly />';

}

?>

```

```

<label for="sd1">Desvio padrão pop. 1 (o1):</label>

```

```

<input type="number" id="sd1" name="sd1" class="showTooltip" title="Corresponde ao
desvio padrão da população 1, pode ser obtido em uma amostra piloto ou na literatura."
placeholder="Informe o desvio padrão da pop. 1." step="0.01" min="0" max="9999999"
value=<?php echo $data['sd1']; ?> />

```

```

<label for="sd2">Desvio padrão pop. 2 (o2):</label>

```

```

<input type="number" id="sd2" name="sd2" class="showTooltip" title="Corresponde ao
desvio padrão da população 2, pode ser obtido em uma amostra piloto ou na literatura."
placeholder="Informe o desvio padrão da pop. 2." step="0.01" min="0" max="9999999"
value=<?php echo $data['sd2']; ?> />

```

```

<label for="x1">Média pop. 1 ( $\mu_1$ ):</label>

```

```
<input type="number" id="x1" name="x1" class="showTooltip" title="Corresponde a
média da população 1, pode ser obtido em uma amostra piloto ou na literatura."
placeholder="Informe a média da pop. 1." step="0.01" min="0" max="9999999"
value=<?php echo $data['x1']; ?> />
```

```
<label for="x2">Média pop. 2 ( $\mu_0$ ):</label>
```

```
<input type="number" id="x2" name="x2" class="showTooltip" title="Corresponde a
média da população 2, pode ser obtido em uma amostra piloto ou na literatura."
placeholder="Informe a média da pop. 2." step="0.01" min="0" max="9999999"
value=<?php echo $data['x2']; ?> />
```

```
<label for="k">k:</label>
```

```
<input type="number" id="k" name="k" class="showTooltip" title="Corresponde à
razão entre o tamanho das duas amostras." placeholder="Informe o valor de k." step="0.01"
min="0" max="9999999" value=<?php echo $data['k']; ?> />
```

```
<button type="submit" class="pure-button pure-button-primary">Calcular!</button>
```

```
</fieldset>
```

```
<?php
```

```
if (isset($data['n1']) || isset($data['n2']))
```

```
{
```

```
    echo '<fieldset><legend>Resultado:</legend>';
```

```
echo '<label for="n1">Tamanho da amostra 1 (n):</label>';  
  
echo '<input type="text" id="n1" name="n1" value="" . $data['n1'] . "" readonly />';  
  
echo '</fieldset>';
```

```
echo '<label for="n2">Tamanho da amostra 2 (n):</label>';  
  
echo '<input type="text" id="n2" name="n2" value="" . $data['n2'] . "" readonly />';  
  
echo '</fieldset>';
```

```
}
```

```
?>
```

```
</form>
```

```
</div>
```

```
Application\views\calcular\dmaidb.php
```

```
<div class="content">
```

```
<h2>Duas Médias amostrais independentes de tamanhos diferentes (Teste Bilateral)</h2>
```

```
<form class="pure-form pure-form-stacked" action="duas-medias-amostrais-independentes-  
tamanho-diferente-teste-bilateral" method="post" >
```

```
<fieldset>
```

```
  <legend>Informe os parâmetros</legend>
```

```
  <img src= "../assets/img/dmaitdb.png" alt="Fórmula para o cálculo de n." class="pure-  
img">
```

```
  <label for="a">Nível de Confiança (1-a):</label>
```

```
  <div class="pure-g">
```

```
    <input type="number" id="a" name="a" class="pure-u-sm-1-6 showTooltip"  
title="Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo o  
mais comum 95%. Quanto maior o nível de confiança, maior a amostra."  
placeholder="Informe o nível de confiança." step="0.01" min="0" max="9999999"  
value=<?php echo $data['a']; ?> required />
```

```
    <span class="percent">%</span>
```

```
  </div>
```

```
<?php
```

```
if (isset($data['za']))
```

```
{
```

```
  echo '<label for="za">Valor Z (Nível de Confiança):</label>';
```

```
  echo '<input type="number" id="za" name="za" value="" . $data['za'] . "" readonly />';
```

```
}
```

```
?>
```

```
<label for="b">Poder do teste ( $\beta$ ):</label>
```

```
<div class="pure-g">
```

```
    <input type="number" id="b" name="b" class="pure-u-sm-1-6 showTooltip" title="É a
    probabilidade de se identificar a diferença que realmente existe, geralmente utiliza-se o poder
    do teste de 80%." placeholder="Informe o poder do teste." step="0.01" min="0"
    max="9999999" value=<?php echo $data['b']; ?> required />
```

```
    <span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['zb']))
```

```
{
```

```
    echo '<label for="zb">Valor Z (Poder do Teste):</label>';
```

```
    echo '<input type="number" id="zb" name="zb" value="" . $data['zb'] . "" readonly />';
```

```
}
```

```
?>
```

```
<label for="sd1">Desvio padrão pop. 1 ( $\sigma_1$ ):</label>
```



```
<input type="number" id="sd1" name="sd1" class="showTooltip" title="Corresponde ao desvio padrão da população 1, pode ser obtido em uma amostra piloto ou na literatura." placeholder="Informe o desvio padrão da pop. 1." step="0.01" min="0" max="9999999" value=<?php echo $data['sd1']; ?> />
```

```
<label for="sd2">Desvio padrão pop. 2 ( $\sigma_2$ ):</label>
```

```
<input type="number" id="sd2" name="sd2" class="showTooltip" title="Corresponde ao desvio padrão da população 2, pode ser obtido em uma amostra piloto ou na literatura." placeholder="Informe o desvio padrão da pop. 2." step="0.01" min="0" max="9999999" value=<?php echo $data['sd2']; ?> />
```

```
<label for="x1">Média pop. 1 ( $\mu_1$ ):</label>
```

```
<input type="number" id="x1" name="x1" class="showTooltip" title="Corresponde a média da população 1, pode ser obtido em uma amostra piloto ou na literatura." placeholder="Informe a média da pop. 1." step="0.01" min="0" max="9999999" value=<?php echo $data['x1']; ?> />
```

```
<label for="x2">Média pop. 2 ( $\mu_2$ ):</label>
```

```
<input type="number" id="x2" name="x2" class="showTooltip" title="Corresponde a média da população 2, pode ser obtido em uma amostra piloto ou na literatura." placeholder="Informe a média da pop. 2." step="0.01" min="0" max="9999999" value=<?php echo $data['x2']; ?> />
```

```
<label for="k">k:</label>
```

```
<input type="number" id="k" name="k" class="showTooltip" title="Corresponde à
razão entre o tamanho das duas amostras." placeholder="Informe o valor de k." step="0.01"
min="0" max="9999999" value=<?php echo $data['k']; ?> />
```

```
<button type="submit" class="pure-button pure-button-primary">Calcular!</button>
```

```
</fieldset>
```

```
<?php
```

```
if (isset($data['n1']) || isset($data['n2']))
```

```
{
```

```
    echo '<fieldset><legend>Resultado:</legend>';
```

```
    echo '<label for="n1">Tamanho da amostra 1 (n):</label>';
```

```
    echo '<input type="text" id="n1" name="n1" value="" . $data['n1'] . "" readonly />';
```

```
    echo '</fieldset>';
```

```
    echo '<label for="n2">Tamanho da amostra 2 (n):</label>';
```

```
    echo '<input type="text" id="n2" name="n2" value="" . $data['n2'] . "" readonly />';
```

```
    echo '</fieldset>';
```

```
}
```

```
?>
```

</form>

</div>

Application\views\calcular\dmap.php

<div class="content">

<h2>Duas Médias amostrais pareadas (Teste Unilateral)</h2>

<form class="pure-form pure-form-stacked" action="duas-medias-amostrais-pareadas-teste-unilateral" method="post" >

<fieldset>

<legend>Informe os parâmetros</legend>

<label for="a">Nível de Confiança (1-a):</label>

<div class="pure-g">

<input type="number" id="a" name="a" class="pure-u-sm-1-6 showTooltip" title="Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo o mais comum 95%. Quanto maior o nível de confiança, maior a amostra.">

```
placeholder="Informe o nível de confiança." step="0.01" min="0" max="9999999"
value=<?php echo $data['a']; ?> required />
```

```
<span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['za']))
```

```
{
```

```
echo '<label for="za">Valor Z (Nível de Confiança):</label>';
```

```
echo '<input type="number" id="za" name="za" value="" . $data['za'] . "" readonly />';
```

```
}
```

```
?>
```

```
<label for="b">Poder do teste ( $\beta$ ):</label>
```

```
<div class="pure-g">
```

```
<input type="number" id="b" name="b" class="pure-u-sm-1-6 showTooltip" title="É a
probabilidade de se identificar a diferença que realmente existe, geralmente utiliza-se o poder
do teste de 80%." placeholder="Informe o poder do teste." step="0.01" min="0"
max="9999999" value=<?php echo $data['b']; ?> required />
```

```
<span class="percent">%</span>
```

```
</div>
```

```

<?php

if (isset($data['zb']))

{

    echo '<label for="zb">Valor Z (Poder do Teste):</label>';

    echo '<input type="number" id="zb" name="zb" value="" . $data['zb'] . "" readonly />';

}

?>

```

```

<label for="sd">Desvio padrão da diferença (od):</label>

```

```

<input type="number" id="sd" name="sd" class="showTooltip" title="Corresponde ao
desvio padrão das diferenças obtidas entre os dois momentos, pode ser obtido em uma
amostra piloto ou na literatura." placeholder="Informe o desvio padrão." step="0.01" min="0"
max="9999999" value=<?php echo $data['sd']; ?> />

```

```

<label for="dif">Diferença entre as médias (?):</label>

```

```

<input type="number" id="dif" name="dif" class="showTooltip" title="A diferença
estimada entre as médias obtidas nos dois momentos, pode ser obtida em uma amostra piloto
ou na literatura." placeholder="Informe a diferença entre as médias." step="0.01" min="0"
max="9999999" value=<?php echo $data['dif']; ?> />

```

```

<button type="submit" class="pure-button pure-button-primary">Calcular!</button>

```

```
</fieldset>
```

```
<?php
```

```
if (isset($data['n']))
```

```
{
```

```
    echo '<fieldset><legend>Resultado:</legend>';
```

```
    echo '<label for="n">Tamanho da amostra (n):</label>';
```

```
    echo '<input type="text" id="n" name="n" value="' . $data['n'] . '" readonly />';
```

```
    echo '</fieldset>';
```

```
}
```

```
?>
```

```
</form>
```

```
</div>
```

```
Application\views\calcular\dmabp.php
```

```
<div class="content">
```

```
<h2>Duas Médias amostrais pareadas (Teste Bilateral)</h2>
```

```
<form class="pure-form pure-form-stacked" action="duas-medias-amostrais-pareadas-teste-bilateral" method="post" >
```

```
<fieldset>
```

```
<legend>Informe os parâmetros</legend>
```

```
<img src= "../assets/img/dmapb.png" alt="Fórmula para o cálculo de n." class="pure-img">
```

```
<label for="a">Nível de Confiança (1-a):</label>
```

```
<div class="pure-g">
```

```
<input type="number" id="a" name="a" class="pure-u-sm-1-6 showTooltip"
title="Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo o
mais comum 95%. Quanto maior o nível de confiança, maior a amostra."
placeholder="Informe o nível de confiança." step="0.01" min="0" max="9999999"
value=<?php echo $data['a']; ?> required />
```

```
<span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['za']))
```

```
{
```

```
echo '<label for="za">Valor Z (Nível de Confiança):</label>';
```

```

    echo '<input type="number" id="za" name="za" value="" . $data['za'] . "' readonly />';

}

?>

```

```
<label for="b">Poder do teste ( $\beta$ ):</label>
```

```
<div class="pure-g">
```

```

    <input type="number" id="b" name="b" class="pure-u-sm-1-6 showTooltip" title="É a
    probabilidade de se identificar a diferença que realmente existe, geralmente utiliza-se o poder
    do teste de 80%." placeholder="Informe o poder do teste." step="0.01" min="0"
    max="9999999" value=<?php echo $data['b']; ?> required />

```

```
    <span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['zb']))
```

```
{
```

```
    echo '<label for="zb">Valor Z (Poder do Teste):</label>';
```

```
    echo '<input type="number" id="zb" name="zb" value="" . $data['zb'] . "' readonly />';
```

```
}
```

```
?>
```



```
<label for="sd">Desvio padrão da diferença (od):</label>
```

```
<input type="number" id="sd" name="sd" class="showTooltip" title="Corresponde ao desvio padrão das diferenças obtidas entre os dois momentos, pode ser obtido em uma amostra piloto ou na literatura." placeholder="Informe o desvio padrão." step="0.01" min="0" max="9999999" value=<?php echo $data['sd']; ?> />
```

```
<label for="dif">Diferença entre as médias (?):</label>
```

```
<input type="number" id="dif" name="dif" class="showTooltip" title="A diferença estimada entre as médias obtidas nos dois momentos, pode ser obtida em uma amostra piloto ou na literatura." placeholder="Informe a diferença entre as médias." step="0.01" min="0" max="9999999" value=<?php echo $data['dif']; ?> />
```

```
<button type="submit" class="pure-button pure-button-primary">Calcular!</button>
```

```
</fieldset>
```

```
<?php
```

```
if (isset($data['n']))
```

```
{
```

```
    echo '<fieldset><legend>Resultado:</legend>';
```

```
    echo '<label for="n">Tamanho da amostra (n):</label>';
```

```
    echo '<input type="text" id="n" name="n" value="" . $data['n'] . "" readonly />';
```

```
    echo '</fieldset>';
```

```
}  
?>  
</form>  
</div>  
Application\views\calcular\ededp.php  
  
<div class="content">  
  
<h2>Estimar diferença entre duas proporções (intervalo de confiança da diferença)</h2>  
  
<form class="pure-form pure-form-stacked" action="estimar-diferenca-entre-duas-  
proporcoes" method="post" >  
  
<fieldset>  
  
<legend>Informe os parâmetros</legend>  
  
<img src= "../assets/img/ededp.png" alt="Fórmula para o cálculo de n." class="pure-  
img">  
  
<label for="a">Nível de Confiança (1-a):</label>  
  
<div class="pure-g">
```

```
<input type="number" id="a" name="a" class="pure-u-sm-1-6 showTooltip"
title="Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo o
mais comum 95%. Quanto maior o nível de confiança, maior a amostra."
placeholder="Informe o nível de confiança." step="0.01" min="0" max="9999999"
value=<?php echo $data['a']; ?> required />
```

```
<span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['z']))
```

```
{
```

```
echo '<label for="z">Valor de z tabelado (z):</label>';
```

```
echo '<input type="number" class="pure-u-sm-1-6 showTooltip" title="Esse valor
mostra a distância de um certo parâmetro em desvios padrões, para um a de 5% o z tabelado é
de 1,96." id="z" name="z" value="" . $data['z'] . "" readonly />';
```

```
}
```

```
?>
```

```
<label for="p1">Proporção grupo 1 - em % (P1):</label>
```

```
<div class="pure-g">
```

```
<input type="number" id="p1" name="p1" class="showTooltip" title="Percentual
estimado na população 1 do evento que se pretende estudar, pode-se fazer um estudo piloto,
ou no desconhecimento utiliza-se 50% pois tende a dar uma maior amostragem."
```

```
placeholder="Informe a proporção para o grupo 1." step="0.01" min="0" max="9999999"
value=<?php echo $data['p1']; ?> />
```

```
<span class="percent">%</span>
```

```
</div>
```

```
<label for="p2">Proporção grupo 2 - em % (P2):</label>
```

```
<div class="pure-g">
```

```
<input type="number" id="p2" name="p2" class="showTooltip" title="Percentual
estimado na população 2 do evento que se pretende estudar, pode-se fazer um estudo piloto,
ou no desconhecimento utiliza-se 50% pois tende a dar uma maior amostragem."
```

```
placeholder="Informe a proporção para o grupo 2." step="0.01" min="0" max="9999999"
value=<?php echo $data['p2']; ?> />
```

```
<span class="percent">%</span>
```

```
</div>
```

```
<label for="d">Diferença (d):</label>
```

```
<input type="number" id="d" name="d" class="showTooltip" title="É a diferença
mínima que o tamanho amostral é capaz de detectar, definida pelo pesquisador."
```

```
placeholder="Informe a diferença." step="0.01" min="0" max="9999999" value=<?php echo
$data['d']; ?> />
```

```
<button type="submit" class="pure-button pure-button-primary">Calcular!</button>
```

```
</fieldset>
```

```
<?php

if (isset($data['n']))

{

    echo '<fieldset><legend>Resultado:</legend>';

    echo '<label for="n">Tamanho da amostra (n):</label>';

    echo '<input type="text" id="n" name="n" value="" . $data['n'] . "" readonly />';

    echo 'Os tamanhos das amostras são iguais.';

    echo '</fieldset>';

}

?>

</form>

</div>

Application\views\calcular\empf.php

<div class="content">

<h2>Estimativa para Médias - População Finita</h2>
```

```
<form class="pure-form pure-form-stacked" action="estimativa-para-medias-populacao-
finita" method="post" >
```

```
<fieldset>
```

```
<legend>Informe os parâmetros</legend>
```

```
<img src= "../assets/img/empf.png" alt="Fórmula para o cálculo de n." class="pure-
img">
```

```
<label for="a">Nível de Confiança (1-a):</label>
```

```
<div class="pure-g">
```

```
<input type="number" id="a" name="a" class="pure-u-sm-1-6 showTooltip"
title="Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo o
mais comum 95%. Quanto maior o nível de confiança, maior a amostra."
placeholder="Informe o nível de confiança." step="0.01" min="0" max="9999999"
value=<?php echo $data['a']; ?> required />
```

```
<span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['z']))
```

```
{
```

```
echo '<label for="z">Valor de z tabelado (z):</label>';
```

```
echo '<input type="number" class="pure-u-sm-1-6 showTooltip" title="Esse valor mostra a distância de um certo parâmetro em desvios padrões, para um  $\alpha$  de 5% o z tabelado é de 1,96." id="z" name="z" value="" . $data['z'] . "' readonly />';
```

```
}
```

```
?>
```

```
<label for="sd">Desvio padrão (S ou  $\sigma$ ):</label>
```

```
<input type="number" id="sd" name="sd" class="showTooltip" title="Corresponde ao desvio padrão da variável em estudo, se não for conhecido pode ser obtido de um estudo piloto ou de estudos na literatura." placeholder="Informe o desvio padrão." step="0.01" min="0" max="9999999" value=<?php echo $data['sd']; ?> />
```

```
<label for="pop">Tamanho da população (N):</label>
```

```
<input type="number" id="pop" name="pop" class="showTooltip" title="Número de indivíduos que compõe a população a ser estudada." placeholder="Informe o tamanho da população." step="0.01" min="0" max="9999999" value=<?php echo $data['pop']; ?> />
```

```
<label for="e">Margem de erro (e):</label>
```

```
<input type="number" id="e" name="e" class="showTooltip" title="Corresponde ao semi intervalo de confiança IC/2, isto é, a distância a mais ou a menos da média que se deseja que tenha o intervalo de confiança." step="0.01" min="0" max="9999999" value=<?php echo $data['e']; ?> required />
```

```
<button type="submit" class="pure-button pure-button-primary">Calcular!</button>
```

```
</fieldset>
```

```
<?php
```

```
if (isset($data['n']))
```

```
{
```

```
    echo '<fieldset><legend>Resultado:</legend>';
```

```
    echo '<label for="n">Tamanho da amostra (n):</label>';
```

```
    echo '<input type="text" id="n" name="n" value="' . $data['n'] . '" readonly />';
```

```
    echo '</fieldset>';
```

```
}
```

```
?>
```

```
</form>
```

```
</div>
```

```
Application\views\calcular\empi.php
```

```
<div class="content">
```

```
<h2>Estimativa para Médias - População Infinita</h2>
```



```
<form class="pure-form pure-form-stacked" action="estimativa-para-medias-populacao-
infinita" method="post" >
```

```
<fieldset>
```

```
<legend>Informe os parâmetros</legend>
```

```
<img src= "../assets/img/empi.png" alt="Fórmula para o cálculo de n." class="pure-
img">
```

```
<label for="a">Nível de Confiança (1-a):</label>
```

```
<div class="pure-g">
```

```
<input type="number" id="a" name="a" class="pure-u-sm-1-6 showTooltip"
title="Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo o
mais comum 95%. Quanto maior o nível de confiança, maior a amostra."
placeholder="Informe o nível de confiança." step="0.01" min="0" max="9999999"
value=<?php echo $data['a']; ?> required />
```

```
<span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['z']))
```

```
{
```

```
echo '<label for="z">Valor de z tabelado (z):</label>';
```

```
echo '<input type="number" class="pure-u-sm-1-6 showTooltip" title="Esse valor
mostra a distância de um certo parâmetro em desvios padrões, para um a de 5% o z tabelado é
de 1,96." id="z" name="z" value="" . $data['z'] . "' readonly />';
```

```
}
```

```
?>
```

```
<label for="sd">Desvio padrão (S ou o):</label>
```

```
<input type="number" id="sd" name="sd" class="showTooltip" title="Corresponde ao
desvio padrão da variável em estudo, se não for conhecido pode ser obtido de uma estudo
piloto ou de estudos na literatura." placeholder="Informe o desvio padrão." step="0.01"
min="0" max="9999999" value=<?php echo $data['sd']; ?> required />
```

```
<label for="e">Margem de erro (e):</label>
```

```
<input type="number" id="e" name="e" class="showTooltip" title="Corresponde ao
semi intervalo de confiança IC/2, isto é, a distância a mais ou a menos da média que se deseja
que tenha o intervalo de confiança." step="0.01" min="0" max="9999999" value=<?php echo
$data['e']; ?> required />
```

```
<div class="pure-controls">
```

```
<button type="submit" class="pure-button pure-button-primary">Calcular!</button>
```

```
</div>
```

```
</fieldset>
```

```

<?php
    if (isset($data['n']))
    {
        echo '<fieldset><legend>Resultado:</legend>';

        echo '<label for="n">Tamanho da amostra (n): </label>';

        echo '<input type="text" id="n" name="n" value="" . $data['n'] . "" readonly />';

        echo '</fieldset>';
    }
?>

</form>

</div>

```

Application\views\calcular\epppf.php

```
<div class="content">
```

```
<h2>Estimativas para Proporções - População Finita</h2>
```

```
<p>Pode ser usada para calcular a prevalência de uma determinada doença e uma certa população</p>
```

```
<form class="pure-form pure-form-stacked" action="estimativas-para-proporcoes-
populacao-finita" method="post" >
```

```
<fieldset>
```

```
  <legend>Informe os parâmetros</legend>
```

```
  <img src= "../assets/img/epppf.png" alt="Fórmula para o cálculo de n." class="pure-  
img">
```

```
  <label for="a">Nível de Confiança (1-a):</label>
```

```
  <div class="pure-g">
```

```
    <input type="number" id="a" name="a" class="pure-u-sm-1-6 showTooltip"  
title="Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo o  
mais comum 95%. Quanto maior o nível de confiança, maior a amostra."  
placeholder="Informe o nível de confiança." step="0.01" min="0" max="9999999"  
value=<?php echo $data['a']; ?> required />
```

```
    <span class="percent">%</span>
```

```
  </div>
```

```
<?php
```

```
if (isset($data['z']))
```

```
{
```

```
  echo '<label for="z">Valor de z tabelado:</label>';
```

```
  echo '<input type="number" id="z" name="z" value="" . $data['z'] . "" readonly />';
```

```
}
```

?>

<label for="p">Prevalência da condição - em % (p):</label>

<div class="pure-g">

<input type="number" id="p" name="p" class="showTooltip" title="Percentual estimado do evento que se pretende estudar, pode-se fazer um estudo piloto, ou no desconhecimento utiliza-se 50% pois tende a dar um maior tamanho de amostragem." placeholder="Informe a prevalência da condição." step="0.01" min="0" max="9999999" value=<?php echo \$data['p']; ?> />

%

</div>

<label for="e">Erro tolerável - em % (d ou e):</label>

<input type="number" id="e" name="e" class="showTooltip" title="Corresponde ao afastamento entre parâmetro populacional e a estimativa, isto é, o erro máximo convencionado pelo pesquisador. Quanto menor o erro tolerável, maior será o tamanho da amostra, normalmente se usa 5%." placeholder="Informe o erro tolerável." step="0.01" min="0" max="9999999" value=<?php echo \$data['e']; ?> />

<label for="pop">Tamanho da população (N):</label>

<input type="number" id="pop" name="pop" class="showTooltip" title="Número de indivíduos que compõe a população a ser estudada." placeholder="Informe o tamanho da população." step="0.01" min="0" max="9999999" value=<?php echo \$data['pop']; ?> />

```
<button type="submit" class="pure-button pure-button-primary">Calcular!</button>
```

```
</fieldset>
```

```
<?php
```

```
if (isset($data['n']))
```

```
{
```

```
    echo '<fieldset><legend>Resultado:</legend>';
```

```
    echo '<label for="n">Tamanho da amostra (n):</label>';
```

```
    echo '<input type="number" id="n" name="n" value="' . $data['n'] . '" readonly />';
```

```
    echo '</fieldset>';
```

```
}
```

```
?>
```

```
</form>
```

```
</div>
```

```
Application\views\calcular\epppi.php
```

```
<div class="content">
```

```
<h2>Estimativas para Proporções - População Infinita</h2>
```

```
<form class="pure-form pure-form-stacked" action="estimativas-para-proporcoes-
populacao-infinita" method="post" >
```

```
<fieldset>
```

```
<legend>Informe os parâmetros</legend>
```

```
<img src= "../assets/img/epppi.png" alt="Fórmula para o cálculo de n." class="pure-
img">
```

```
<label for="a">Nível de Confiança (1-a):</label>
```

```
<div class="pure-g">
```

```
<input type="number" id="a" name="a" class="pure-u-sm-1-6 showTooltip"
title="Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo o
mais comum 95%. Quanto maior o nível de confiança, maior a amostra."
placeholder="Informe o nível de confiança." step="0.01" min="0" max="9999999"
value=<?php echo $data['a']; ?> required />
```

```
<span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['z']))
```

```
{
```

```
echo '<label for="z">Valor de z tabelado (z):</label>';
```

```
echo '<input type="number" class="pure-u-sm-1-6 showTooltip" title="Esse valor
mostra a distância de um certo parâmetro em desvios padrões, para um a de 5% o z tabelado é
de 1,96." id="z" name="z" value="" . $data['z'] . "' readonly />';
```

```
}
```

```
?>
```

```
<label for="p">Prevalência da condição - em % (p)</label>
```

```
<div class="pure-g">
```

```
<input type="number" id="p" name="p" class="showTooltip" title="Percentual
estimado do evento que se pretende estudar, pode-se fazer um estudo piloto, ou no
desconhecimento utiliza-se 50% pois tende a dar um maior tamanho de amostragem."
placeholder="Informe a prevalência da condição." step="0.01" min="0" max="9999999"
value=<?php echo $data['p']; ?> />
```

```
<span class="percent">%</span>
```

```
</div>
```

```
<label for="e">Erro tolerável - em % (d ou e):</label>
```

```
<input type="number" id="e" name="e" class="showTooltip" title="Corresponde ao
afastamento entre parâmetro populacional e a estimativa, isto é, o erro máximo
convencionado pelo pesquisador. Quanto menor o erro tolerável, maior será o tamanho da
amostra, normalmente se usa 5%." placeholder="Informe o erro tolerável." step="0.01"
min="0" max="9999999" value=<?php echo $data['e']; ?> />
```

```
<button type="submit" class="pure-button pure-button-primary">Calcular!</button>
```



```
</fieldset>
```

```
<?php
```

```
if (isset($data['n']))
```

```
{
```

```
    echo '<fieldset><legend>Resultado:</legend>';
```

```
    echo '<label for="n">Tamanho da amostra (n):</label>';
```

```
    echo '<input type="text" id="n" name="n" value="' . $data['n'] . '" readonly />';
```

```
    echo '</fieldset>';
```

```
}
```

```
?>
```

```
</form>
```

```
</div>
```

```
Application\views\calcular\mecvr.php
```

```
<div class="content">
```

```
<h2>Média experimental contra um valor referência (Teste Unilateral)</h2>
```

```
<form class="pure-form pure-form-stacked" action="media-experimental-contra-um-valor-referencia-teste-unilateral" method="post" >
```

```
<fieldset>
```

```
<legend>Informe os parâmetros</legend>
```

```
<img src= "../assets/img/mecvr.png" alt="Fórmula para o cálculo de n." class="pure-img">
```

```
<label for="a">Nível de Confiança (1-a):</label>
```

```
<div class="pure-g">
```

```
<input type="number" id="a" name="a" class="pure-u-sm-1-6 showTooltip"
title="Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo o
mais comum 95%. Quanto maior o nível de confiança, maior a amostra."
placeholder="Informe o nível de confiança." step="0.01" min="0" max="9999999"
value=<?php echo $data['a']; ?> required />
```

```
<span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['za']))
```

```
{
```

```
echo '<label for="za">Valor Z (Nível de Confiança):</label>';
```

```

    echo '<input type="number" id="za" name="za" value="" . $data['za'] . "' readonly />';

}

?>

```

```
<label for="b">Poder do teste ( $\beta$ ):</label>
```

```
<div class="pure-g">
```

```

    <input type="number" id="b" name="b" class="pure-u-sm-1-6 showTooltip" title="É a
    probabilidade de se identificar a diferença que realmente existe, geralmente utiliza-se o poder
    do teste de 80%." placeholder="Informe o nível de confiança." step="0.01" min="0"
    max="9999999" value=<?php echo $data['b']; ?> required />

```

```
    <span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['zb']))
```

```
{
```

```
    echo '<label for="zb">Valor Z (Poder do Teste):</label>';
```

```
    echo '<input type="number" id="zb" name="zb" value="" . $data['zb'] . "' readonly />';
```

```
}
```

```
?>
```

```
<label for="sd">Desvio padrão (S ou o):</label>
```

```
<input type="number" id="sd" name="sd" class="showTooltip" title="Corresponde ao desvio padrão da variável em estudo, se não for conhecido pode ser obtido de um estudo piloto ou de estudos na literatura." placeholder="Informe o desvio padrão." step="0.01" min="0" max="9999999" value=<?php echo $data['sd']; ?> />
```

```
<label for="x0">Média referência ( $\mu_0$ ):</label>
```

```
<input type="number" id="x0" name="x0" class="showTooltip" title="Corresponde à média da população de referência." placeholder="Informe a média de referência." step="0.01" min="0" max="9999999" value=<?php echo $data['x0']; ?> />
```

```
<label for="x1">Média experimental ( $\mu_1$ ):</label>
```

```
<input type="number" id="x1" name="x1" class="showTooltip" title="Corresponde à média que se espera obter no experimento, ela pode se estimada por um estudo piloto ou pela literatura." placeholder="Informe a média experimental." step="0.01" min="0" max="9999999" value=<?php echo $data['x1']; ?> />
```

```
<button type="submit" class="pure-button pure-button-primary">Calcular!</button>
```

```
</fieldset>
```

```
<?php
```

```
if (isset($data['n']))
```

```
{
```

```

    echo '<fieldset><legend>Resultado:</legend>';

    echo '<label for="n">Tamanho da amostra (n):</label>';

    echo '<input type="text" id="n" name="n" value="" . $data['n'] . "" readonly />';

    echo '</fieldset>';

    }

?>

</form>

</div>

Application\views\calcular\mecvrb.php

<div class="content">

<h2>Média experimental contra um valor referência (Teste Bilateral)</h2>

<form class="pure-form pure-form-stacked" action="media-experimental-contra-um-valor-
referencia-teste-bilateral" method="post" >

<fieldset>

<legend>Informe os parâmetros</legend>

<img src= "../assets/img/mecvrb.png" alt="Fórmula para o cálculo de n." class="pure-
img">

```

```
<label for="a">Nível de Confiança (1-a):</label>
```

```
<div class="pure-g">
```

```
    <input type="number" id="a" name="a" class="pure-u-sm-1-6 showTooltip"
title="Representa a probabilidade esperada de acerto, geralmente entre 80 a 99%, sendo o
mais comum 95%. Quanto maior o nível de confiança, maior a amostra."
placeholder="Informe o nível de confiança." step="0.01" min="0" max="9999999"
value=<?php echo $data['a']; ?> required />
```

```
    <span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['za']))
```

```
{
```

```
    echo '<label for="za">Valor Z (Nível de Confiança):</label>';
```

```
    echo '<input type="number" id="za" name="za" value="" . $data['za'] . "" readonly />';
```

```
}
```

```
?>
```

```
<label for="b">Poder do teste ( $\beta$ ):</label>
```

```
<div class="pure-g">
```

```
<input type="number" id="b" name="b" class="pure-u-sm-1-6 showTooltip" title="É a
probabilidade de se identificar a diferença que realmente existe, geralmente utiliza-se o poder
do teste de 80%." placeholder="Informe o nível de confiança." step="0.01" min="0"
max="9999999" value=<?php echo $data['b']; ?> required />
```

```
<span class="percent">%</span>
```

```
</div>
```

```
<?php
```

```
if (isset($data['zb']))
```

```
{
```

```
echo '<label for="zb">Valor Z (Poder do Teste):</label>';
```

```
echo '<input type="number" id="zb" name="zb" value="" . $data['zb'] . "" readonly />';
```

```
}
```

```
?>
```

```
<label for="sd">Desvio padrão (S ou o):</label>
```

```
<input type="number" id="sd" name="sd" class="showTooltip" title="Corresponde ao
desvio padrão da variável em estudo, se não for conhecido pode ser obtido de um estudo
piloto ou de estudos na literatura." placeholder="Informe o desvio padrão." step="0.01"
min="0" max="9999999" value=<?php echo $data['sd']; ?> />
```

```
<label for="x0">Média referência ( $\mu$ ):</label>
```

```
<input type="number" id="x0" name="x0" class="showTooltip" title="Corresponde à
média da população de referência." placeholder="Informe a média de referência."
step="0.01" min="0" max="9999999" value=<?php echo $data['x0']; ?> />
```

```
<label for="x1">Média experimental ( $\mu_1$ ):</label>
```

```
<input type="number" id="x1" name="x1" class="showTooltip" title="Corresponde a
média que se espera obter no experimento, ela pode se estimada por um estudo piloto ou pela
literatura." placeholder="Informe a média experimental." step="0.01" min="0"
max="9999999" value=<?php echo $data['x1']; ?> />
```

```
<button type="submit" class="pure-button pure-button-primary">Calcular!</button>
```

```
</fieldset>
```

```
<?php
```

```
if (isset($data['n']))
```

```
{
```

```
echo '<fieldset><legend>Resultado:</legend>';
```

```
echo '<label for="n">Tamanho da amostra (n):</label>';
```

```
echo '<input type="text" id="n" name="n" value="' . $data['n'] . '" readonly />';
```

```
echo '</fieldset>';
```

```
}
```

```
?>
```


</form>

</div>

Application\views\home\index.php

<div class="content">

<h2 class="content-subhead">Para estimar ou comparar médias</h2>

<p>Estimativa para Médias - População Infinita</p>

<p>Utiliza-se quando o pesquisador deseja estimar o intervalo de confiança para média populacional, considerando uma população infinita (número de elementos muito elevado. Ex: População de São Paulo, de um País).</p>

<p>Estimativa para Médias - População Finita</p>

<p>Utiliza-se quando o pesquisador deseja estimar o intervalo de confiança para média populacional, considerando uma população finita (número de elementos limitado. Ex: Número de estudantes de uma escola, pacientes de um hospital).</p>

<p>Média experimental contra um valor referência - Teste Unilateral</p>

<p>Média experimental contra um valor referência - Teste Bilateral</p>

<p>Utiliza-se quando se deseja comparar a média obtida experimentalmente com uma média tida como referência (populacional).</p>

<p>Duas Médias amostrais pareadas - Teste Unilateral</p>

<p>Duas Médias amostrais pareadas - Teste Bilateral</p>

<p>Utiliza-se quando deseja-se comparar médias de um mesmo grupo avaliado em momentos distintos, ou por avaliadores/métodos distintos.</p>

<p>Duas Médias amostrais independentes de mesmo tamanho - Teste Unilateral</p>

<p>Duas Médias amostrais independentes de mesmo tamanho - Teste Bilateral</p>

<p>Duas Médias amostrais independentes de tamanho diferente - Teste Unilateral</p>

<p>Duas Médias amostrais independentes de tamanho diferente - Teste Bilateral</p>

<p>Utiliza-se quando deseja-se comparar médias de dois grupos distintos.</p>

<h2 class="content-subhead">Para estimar e comparar proporções</h2>

<p>Utilizado para estudos comparativos em ensaios clínicos, estudos observacionais do tipo coorte e caso controle; onde se pretende estudar a associação entre duas variáveis dicotômicas.</p>

<p>Estimativas para Proporções - População Infinita</p>

<p>Utiliza-se quando o pesquisador deseja estimar o intervalo de confiança para proporção populacional, considerando uma população infinita (número de elementos muito elevado. Ex: População de São Paulo, de um País).</p>

<p>Estimativas para Proporções - População Finita</p>

<p>Utiliza-se quando o pesquisador deseja estimar o intervalo de confiança para proporção populacional, considerando uma população finita (número de elementos limitado. Ex: Número de estudantes de uma escola, pacientes de um hospital).</p>

<p>Estimar diferença entre duas proporções - intervalo de confiança da diferença</p>

<p>Utiliza-se quando deseja-se estimar intervalo de confiança para a diferença entre as proporções de dois grupos distintos.</p>

<p>Comparar duas proporções - uma experimental com uma referência</p>

<p>Utiliza-se quando se deseja comparar a proporção obtida experimentalmente com uma proporção tida como referência (populacional).</p>

</div>

Application\views\sobre\index.php

<div class="content">

<h2 class="content-subhead">Equipe</h2>

<h3>Desenvolvedores:</h3>

<p>Roberto César Faria e Silva

Lattes

ORCID

</p>

<p>Ismael Mendes dos Santos Junior

Lattes

ORCID

</p>

<p>Marise Fagundes Silveira

Lattes

ORCID

</p>

<p>Renato Sobral Monteiro Junior

Lattes

ORCID

</p>

<h2 class="content-subhead">Como citar?</h2>

<p>Silva, R. C. F.; Santos Junior, I. M.; Silveira, M. F.; Monteiro Junior, R. S. (2022). PPGCS Sample Size [Computer software]. Acesso em: <http://www.ppgcs.org.br/samplesize></p>

<h2 class="content-subhead">Referências bibliográficas utilizadas</h2>

<p>Arango, H. G. Bioestatística: teórica e computacional: com banco de dados reais. 3. ed. Rio de Janeiro: Guanabara Koogan, 2011.</p>

<p>Bolfarine, H.; Bussab, W.O. Elementos de amostragem. São Paulo: Editora Blücher, 2005.</p>

<p>Medronho, R; Carvalho, D.M.; Bloch, K.V.; Luiz, R.R.; Werneck, G.L. (eds.); Epidemiologia. Atheneu: São Paulo, 2002.</p>

</div>

public\index.php

<?php

//dev

//\$root = 'http://localhost/ppgcssize.com.br/public/';

//produção

\$root = 'https://www.oraculoestatistico.com.br/ppgcssize/public/';

//\$root = 'https://www.paineiseletricos.com.br/ppgcssize/public/';

?>

<!DOCTYPE html>

<html lang="pt-br">

<head>

```
<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="https://unpkg.com/purecss@2.0.6/build/pure-min.css"
integrity="sha384-
Uu6IeWbM+gzNVXJcM9XV3SohHtmWE+3VGi496jvgX1jyvDTXfdK+rfZc8C1Aehk5"
crossorigin="anonymous">

<link rel="stylesheet" href="<?php echo $root; ?>/assets/css/styles.css">

<link rel="stylesheet" href="<?php echo $root; ?>/assets/css/tooltip.css">

<title>PPGCS Sample Size</title>

<meta name="author" content="Roberto Cesar Faria e Silva">

<meta name="author" content="Ismael Mendes dos Santos Junior">

<meta name="author" content="Marise Fagundes Silva">

<meta name="author" content="Renato Sobral Monteiro Junior">

<meta name="description" content="Calcule aqui o tamanho da amostra do seu estudo">

<meta name="keywords" content="tamanho amostra pesquisa estudo calculo size sample
size research calculation estimar comparar media proporcao proporcoes finita infinita
experimental referencia pareadas independentes bioestatistica">

</head>

<body>

<div id="layout">
```

```
<!-- Menu toggle -->
```

```
<a href="home" id="menuLink" class="menu-link">
```

```
<!-- Hamburger icon -->
```

```
<span></span>
```

```
</a>
```

```
<div id="menu">
```

```
<div class="pure-menu">
```

```
<a class="pure-menu-heading" href="<?php echo $root; ?>">Sample Size</a>
```

```
<ul class="pure-menu-list">
```

```
<li class="pure-menu-item"><a href="<?php echo $root; ?>" class="pure-menu-  
link">Início</a></li>
```

```
<li class="pure-menu-item"><a href="<?php echo $root; ?>sobre" class="pure-  
menu-link">Sobre</a></li>
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
<div id="main">
```

```
<div class="header">
```



```
<h1>PPGCS Sample Size</h1>
```

```
<h2>Calcule aqui o tamanho da amostra do seu estudo!</h2>
```

```
</div>
```

```
<?php
```

```
require '../Application/autoload.php';
```

```
use Application\core\App;
```

```
use Application\core\Controller;
```

```
$app = new App();
```

```
?>
```

```
</div>
```

```
</div>
```

```
<script src="<?php echo $root; ?>/assets/js/ui.js"></script>
```

```
<script src="<?php echo $root; ?>/assets/js/tooltip.js"></script>
```

```
</body>
```

```
</html>
```

```
public/assets/css/styles.css
```

```
body {  
  
    color: #777;  
  
}
```

```
.pure-img-responsive {  
  
    max-width: 100%;  
  
    height: auto;  
  
}
```

```
.percent {  
  
    display: flex;  
  
    justify-content: center;  
  
    align-items: center;  
  
    padding-left: 8px;  
  
}
```

```
/*
```

Add transition to containers so they can push in and out.

```
*/  
  
#layout,  
  
#menu,  
  
.menu-link {  
  
    -webkit-transition: all 0.2s ease-out;  
  
    -moz-transition: all 0.2s ease-out;  
  
    -ms-transition: all 0.2s ease-out;  
  
    -o-transition: all 0.2s ease-out;  
  
    transition: all 0.2s ease-out;  
  
}
```

```
/*
```

This is the parent ``<div>`` that contains the menu and the content area.

```
*/
```

```
#layout {  
  
    position: relative;  
  
    left: 0;  
  
    padding-left: 0;  
  
}  
  
#layout.active #menu {
```

```
left: 150px;

width: 150px;

}
```

```
#layout.active .menu-link {

left: 150px;

}
```

```
/*
```

The content ``<div>`` is where all your content goes.

```
*/
```

```
.content {

margin: 0 auto;

padding: 0 2em;

max-width: 800px;

margin-bottom: 50px;

line-height: 1.6em;

}
```

```
.header {

margin: 0;
```

```
color: #333;  
  
text-align: center;  
  
padding: 2.5em 2em 0;  
  
border-bottom: 1px solid #eee;  
  
}
```

```
.header h1 {  
  
    margin: 0.2em 0;  
  
    font-size: 3em;  
  
    font-weight: 300;  
  
}
```

```
.header h2 {  
  
    font-weight: 300;  
  
    color: #ccc;  
  
    padding: 0;  
  
    margin-top: 0;  
  
}
```

```
.content-subhead {  
  
    margin: 50px 0 20px 0;  
  
    font-weight: 300;
```

```
color: #888;  
  
}
```

```
/*
```

The `#menu` `<div>` is the parent `<div>` that contains the `.pure-menu` that appears on the left side of the page.

```
*/
```

```
#menu {  
  
    margin-left: -150px; /* "#menu" width */  
  
    width: 150px;  
  
    position: fixed;  
  
    top: 0;  
  
    left: 0;  
  
    bottom: 0;  
  
    z-index: 1000; /* so the menu or its navicon stays above all content */  
  
    background: #191818;  
  
    overflow-y: auto;
```

```
}
```

```
/*
```

All anchors inside the menu should be styled like this.

```
*/
```

```
#menu a {
```

```
    color: #999;
```

```
    border: none;
```

```
    padding: 0.6em 0 0.6em 0.6em;
```

```
}
```

```
/*
```

Remove all background/borders, since we are applying them to #menu.

```
*/
```

```
#menu .pure-menu,
```

```
#menu .pure-menu ul {
```

```
    border: none;
```

```
    background: transparent;
```

```
}
```

```
/*
```

Add that light border to separate items into groups.

```
*/
```

```
#menu .pure-menu ul,
```

```
#menu .pure-menu .menu-item-divided {
```

```
    border-top: 1px solid #333;
```

```
}
```

```
/*
```

Change color of the anchor links on hover/focus.

```
*/
```

```
#menu .pure-menu li a:hover,
```

```
#menu .pure-menu li a:focus {
```

```
    background: #333;
```

```
}
```

```
/*
```

This styles the selected menu item `- `.

```
*/
```

```
#menu .pure-menu-selected,
```

```
#menu .pure-menu-heading {
```

```
    background: #1f8dd6;
```



```
}
```

```
/*
```

```
This styles a link within a selected menu item `<li>`.
```

```
*/
```

```
#menu .pure-menu-selected a {
```

```
    color: #fff;
```

```
}
```

```
/*
```

```
This styles the menu heading.
```

```
*/
```

```
#menu .pure-menu-heading {
```

```
    font-size: 110%;
```

```
    color: #fff;
```

```
    margin: 0;
```

```
}
```

```
/* -- Dynamic Button For Responsive Menu -----*/
```

```
/*
```

The button to open/close the Menu is custom-made and not part of Pure. Here's

how it works:

```
*/
```

```
/*
```

``.menu-link`` represents the responsive menu toggle that shows/hides on small screens.

```
*/
```

```
.menu-link {
```

```
    position: fixed;
```

```
    display: block; /* show this only on small screens */
```

```
    top: 0;
```

```
    left: 0; /* "#menu width" */
```

```
    background: #000;
```

```
    background: rgba(0,0,0,0.7);
```

```
    font-size: 10px; /* change this value to increase/decrease button size */
```

```
    z-index: 10;
```

```
    width: 2em;
```

```
    height: auto;
```

```
    padding: 2.1em 1.6em;
```

```
}
```

```
.menu-link:hover,
```

```
.menu-link:focus {
```

```
    background: #000;
```

```
}
```

```
.menu-link span {
```

```
    position: relative;
```

```
    display: block;
```

```
}
```

```
.menu-link span,
```

```
.menu-link span:before,
```

```
.menu-link span:after {
```

```
    background-color: #fff;
```

```
    pointer-events: none;
```

```
    width: 100%;
```

```
    height: 0.2em;
```

```
}
```

```
.menu-link span:before,  
  
.menu-link span:after {  
  
    position: absolute;  
  
    margin-top: -0.6em;  
  
    content: " ";  
  
}
```

```
.menu-link span:after {  
  
    margin-top: 0.6em;  
  
}
```

```
/* -- Responsive Styles (Media Queries) ----- */
```

```
/*
```

Hides the menu at `48em`, but modify this based on your app's needs.

```
*/
```

```
@media (min-width: 48em) {
```

```
.header,
```

```
.content {  
  
    padding-left: 2em;  
  
    padding-right: 2em;  
  
}  
  
#layout {  
  
    padding-left: 150px; /* left col width "#menu" */  
  
    left: 0;  
  
}  
  
#menu {  
  
    left: 150px;  
  
}  
  
.menu-link {  
  
    position: fixed;  
  
    left: 150px;  
  
    display: none;  
  
}  
  
#layout.active .menu-link {
```

```
        left: 150px;

    }

}

@media (max-width: 48em) {

    /* Only apply this when the window is small. Otherwise, the following
    case results in extra padding on the left:

        * Make the window small.

        * Tap the menu to trigger the active state.

        * Make the window large again.

    */

    #layout.active {

        position: relative;

        left: 150px;

    }

}

public/assets/css/tooltip.css

.tooltip {

    display: none;
```

```
position: absolute;

width: 150px;

padding: 5px 10px;

border-radius: 3px;

box-shadow: 2px 2px 2px #999;

background-color: #AEC6CF;

font-size: .8em;

color: #000;

}
```

```
.tooltip p {

    position: relative;

    z-index: 1;

    margin: 0;

}
```

```
.tooltip span {
```

```
position: absolute;

left: 0;

top: 17px;

width: 15px;

height: 15px;

border-right: 1px solid #AEC6CF;

border-top: 1px solid #AEC6CF;

margin-left: 20px;

margin-top: -19px;

background: inherit;

-webkit-transform: rotate(-45deg);

-moz-transform: rotate(-45deg);

-ms-transform: rotate(-45deg);

transform: rotate(-45deg);

}
```

```
.tooltip.error {
```

```
background: brown;
```



```
    color: #fff;
}
```

```
.tooltip.error span {
    border-right: 1px solid brown;
    border-top: 1px solid brown;
}
```

```
public/assets/js/tooltip.js
```

```
(function () {
    'use strict';
    var Utils = Utils || {};
    Utils.Tooltip = function (el) {
        if (el) {
            this.el = el;
            this.buildDOM();
        }
    };
}
```

```

        this.attachEvents();
    }
};

Utils.Tooltip.prototype.attachEvents = function () {

    var el = this.el,

        that = this;

    el.addEventListener('focus', function () {

        that.showTooltip.call(that);

    });

    el.addEventListener('blur', function () {

        that.hideTooltip.call(that);

    });

};

```

```

Utils.Tooltip.prototype.showTooltip = function () {

    var tipText;

```

```
this.tooltip.style.left = this.el.offsetLeft+ 'px';

this.tooltip.style.top = this.el.offsetTop + this.el.offsetHeight + 7 + 'px';

this.tooltip.style.display = 'block';

tipText = this.tooltip.getElementsByTagName('p')[0];

tipText.innerHTML = this.el.title;

};
```

```
Utils.Tooltip.prototype.hideTooltip = function () {

    this.tooltip.style.display = 'none';

};
```

```
Utils.Tooltip.prototype.buildDOM = function () {

    var theTooltip = document.getElementById('tooltip'),

        body = document.getElementsByTagName('body')[0],

        tipText,

        tipArrow;

    if (!theTooltip) {

        theTooltip = document.createElement('div');
```

```
        theTooltip.id = 'tooltip';

        theTooltip.className = 'tooltip';

        tipArrow = document.createElement('span');

        tipText = document.createElement('p');

        body.appendChild(theTooltip);

        theTooltip.appendChild(tipText);

        theTooltip.appendChild(tipArrow);

    }

    this.tooltip = theTooltip;

    return;

};

Utils.ErrorTooltip = function (el) {

    Utils.Tooltip.call(this, el); // super

};
```

```
Utils.ErrorTooltip.prototype = new Utils.Tooltip();
```

```
Utils.ErrorTooltip.prototype.showTooltip = function () {
```

```
    Utils.Tooltip.prototype.showTooltip.call(this);
```

```
    this.tooltip.className = this.tooltip.className + ' error';
```

```
};
```

```
Utils.ErrorTooltip.prototype.hideTooltip = function () {
```

```
    this.tooltip.className = this.tooltip.className.replace('error', "");
```

```
    Utils.Tooltip.prototype.hideTooltip.call(this);
```

```
};
```

```
window.addEventListener('load', function () {
```

```
    var inputs = document.getElementsByClassName('showTooltip'),
```

```
        errorInputs = document.getElementsByClassName('showErrorTooltip');
```

```
    Array.prototype.forEach.call(inputs, function (key) {
```

```
        new Utils.Tooltip(key);
```

```
    });
```

```
        Array.prototype.forEach.call(errorInputs, function (key) {  
            new Utils.ErrorTooltip(key);  
        });  
    });  
});  
})();
```

```
public/assets/js/ui.js
```

```
(function (window, document) {  
  
    // we fetch the elements each time because docusaurus removes the previous  
    // element references on page navigation  
  
    function getElements() {  
  
        return {  
  
            layout: document.getElementById('layout'),  
  
            menu: document.getElementById('menu'),  
  
            menuLink: document.getElementById('menuLink')  
  
        };  
  
    }  
  
}
```

```
function toggleClass(element, className) {  
  
    var classes = element.className.split(/\s+/);  
  
    var length = classes.length;  
  
    var i = 0;  
  
    for (; i < length; i++) {  
  
        if (classes[i] === className) {  
  
            classes.splice(i, 1);  
  
            break;  
  
        }  
  
    }  
  
    // The className is not found  
  
    if (length === classes.length) {  
  
        classes.push(className);  
  
    }  
  
    element.className = classes.join(' ');  
  
}
```

```
function toggleAll() {
```

```
var active = 'active';

var elements = getElements();

toggleClass(elements.layout, active);

toggleClass(elements.menu, active);

toggleClass(elements.menuLink, active);

}

function handleEvent(e) {

    var elements = getElements();

    if (e.target.id === elements.menuLink.id) {

        toggleAll();

        e.preventDefault();

    } else if (elements.menu.className.indexOf('active') !== -1) {

        toggleAll();

    }

}

document.addEventListener('click', handleEvent);
```


}(this, this.document));

Apendice D : Código Fonte do Oráculo Estatístico

Oraculo\Oraculo.csproj

```
?<?xml version="1.0" encoding="utf-8"?>
```

```
<Project ToolsVersion="15.0"
```

```
xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
```

```
  <Import Project="..\packages\NUnit.3.13.1\build\NUnit.props"
  Condition="Exists('..\packages\NUnit.3.13.1\build\NUnit.props')" />
```

```
  <Import
  Project="..\packages\NUnit3TestAdapter.3.17.0\build\net35\NUnit3TestAdapter.props"
  Condition="Exists('..\packages\NUnit3TestAdapter.3.17.0\build\net35\NUnit3TestAdapter.pr
  ops')" />
```

```
  <Import
  Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props"
  Condition="Exists('$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common
  .props')" />
```

```
<PropertyGroup>
```

```
  <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
```

```
  <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
```

```
  <ProjectGuid>{ 6111FA9E-6B0C-4359-9326-3636281FCFA8}</ProjectGuid>
```

```
  <OutputType>WinExe</OutputType>
```

<RootNamespace>Oraculo</RootNamespace>

<AssemblyName>Oraculo</AssemblyName>

<TargetFrameworkVersion>v4.6.1</TargetFrameworkVersion>

<FileAlignment>512</FileAlignment>

<AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>

<IsWebBootstrapper>>false</IsWebBootstrapper>

<NuGetPackageImportStamp>

</NuGetPackageImportStamp>

<PublishUrl>publish\</PublishUrl>

<Install>true</Install>

<InstallFrom>Disk</InstallFrom>

<UpdateEnabled>>false</UpdateEnabled>

<UpdateMode>Foreground</UpdateMode>

<UpdateInterval>7</UpdateInterval>

<UpdateIntervalUnits>Days</UpdateIntervalUnits>

<UpdatePeriodically>>false</UpdatePeriodically>

<UpdateRequired>>false</UpdateRequired>

<MapFileExtensions>true</MapFileExtensions>

<ApplicationRevision>0</ApplicationRevision>

<ApplicationVersion>1.1.0.%2a</ApplicationVersion>

```
<UseApplicationTrust>>false</UseApplicationTrust>

<BootstrapperEnabled>>true</BootstrapperEnabled>

</PropertyGroup>

<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">

  <PlatformTarget>AnyCPU</PlatformTarget>

  <DebugSymbols>>true</DebugSymbols>

  <DebugType>full</DebugType>

  <Optimize>>false</Optimize>

  <OutputPath>bin\Debug\</OutputPath>

  <DefineConstants>DEBUG;TRACE</DefineConstants>

  <ErrorReport>prompt</ErrorReport>

  <WarningLevel>4</WarningLevel>

</PropertyGroup>

<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">

  <PlatformTarget>AnyCPU</PlatformTarget>

  <DebugType>pdbonly</DebugType>

  <Optimize>>true</Optimize>

  <OutputPath>bin\Release\</OutputPath>

  <DefineConstants>TRACE</DefineConstants>

  <ErrorReport>prompt</ErrorReport>
```

```
<WarningLevel>4</WarningLevel>

</PropertyGroup>

<PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|x86'">

  <DebugSymbols>true</DebugSymbols>

  <OutputPath>bin\x86\Debug\</OutputPath>

  <DefineConstants>DEBUG;TRACE</DefineConstants>

  <DebugType>full</DebugType>

  <PlatformTarget>x86</PlatformTarget>

  <ErrorReport>prompt</ErrorReport>

  <CodeAnalysisRuleSet>MinimumRecommendedRules.ruleset</CodeAnalysisRuleSet>

  <Prefer32Bit>true</Prefer32Bit>

</PropertyGroup>

<PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Release|x86'">

  <OutputPath>bin\x86\Release\</OutputPath>

  <DefineConstants>TRACE</DefineConstants>

  <Optimize>true</Optimize>

  <DebugType>pdbonly</DebugType>

  <PlatformTarget>x86</PlatformTarget>

  <ErrorReport>prompt</ErrorReport>

  <CodeAnalysisRuleSet>MinimumRecommendedRules.ruleset</CodeAnalysisRuleSet>
```

```
<Prefer32Bit>true</Prefer32Bit>

</PropertyGroup>

<PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|x64'">

  <DebugSymbols>true</DebugSymbols>

  <OutputPath>bin\x64\Debug\</OutputPath>

  <DefineConstants>DEBUG;TRACE</DefineConstants>

  <DebugType>full</DebugType>

  <PlatformTarget>x64</PlatformTarget>

  <ErrorReport>prompt</ErrorReport>

  <CodeAnalysisRuleSet>MinimumRecommendedRules.ruleset</CodeAnalysisRuleSet>

  <Prefer32Bit>true</Prefer32Bit>

</PropertyGroup>

<PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Release|x64'">

  <OutputPath>bin\x64\Release\</OutputPath>

  <DefineConstants>TRACE</DefineConstants>

  <Optimize>true</Optimize>

  <DebugType>pdbonly</DebugType>

  <PlatformTarget>x64</PlatformTarget>

  <ErrorReport>prompt</ErrorReport>

  <CodeAnalysisRuleSet>MinimumRecommendedRules.ruleset</CodeAnalysisRuleSet>
```

```
<Prefer32Bit>true</Prefer32Bit>

</PropertyGroup>

<PropertyGroup>

  <ApplicationIcon>Resources\oraculo.ico</ApplicationIcon>

</PropertyGroup>

<ItemGroup>

  <Reference Include="Accord, Version=3.8.0.0, Culture=neutral,
PublicKeyToken=fa1a88e29555ccf7, processorArchitecture=MSIL">

    <HintPath>..\packages\Accord.3.8.0\lib\net46\Accord.dll</HintPath>

  </Reference>

  <Reference Include="Accord.Extensions.Core, Version=1.0.0.0, Culture=neutral,
processorArchitecture=MSIL">

    <HintPath>..\packages\Accord.Extensions.Core.3.0.1\lib\net45\Accord.Extensions.Core.dll</
HintPath>

  </Reference>

  <Reference Include="Accord.Extensions.Math, Version=1.0.0.0, Culture=neutral,
processorArchitecture=MSIL">

    <HintPath>..\packages\Accord.Extensions.Math.3.0.1\lib\net45\Accord.Extensions.Math.dll</
HintPath>

  </Reference>
```

<Reference Include="Accord.Extensions.Statistics, Version=1.0.0.0, Culture=neutral, processorArchitecture=MSIL">

<HintPath>..\packages\Accord.Extensions.Statistics.3.0.1\lib\net45\Accord.Extensions.Statistics.dll</HintPath>

</Reference>

<Reference Include="Accord.Math, Version=3.8.0.0, Culture=neutral, PublicKeyToken=fa1a88e29555ccf7, processorArchitecture=MSIL">

<HintPath>..\packages\Accord.Math.3.8.0\lib\net46\Accord.Math.dll</HintPath>

</Reference>

<Reference Include="Accord.Math.Core, Version=3.8.0.0, Culture=neutral, PublicKeyToken=fa1a88e29555ccf7, processorArchitecture=MSIL">

<HintPath>..\packages\Accord.Math.3.8.0\lib\net46\Accord.Math.Core.dll</HintPath>

</Reference>

<Reference Include="Accord.Statistics, Version=3.8.0.0, Culture=neutral, PublicKeyToken=fa1a88e29555ccf7, processorArchitecture=MSIL">

<HintPath>..\packages\Accord.Statistics.3.8.0\lib\net46\Accord.Statistics.dll</HintPath>

</Reference>

<Reference Include="BouncyCastle.Crypto, Version=1.8.5.0, Culture=neutral, PublicKeyToken=0e99375e54769942">

<HintPath>..\packages\BouncyCastle.1.8.5\lib\BouncyCastle.Crypto.dll</HintPath>

</Reference>

<Reference Include="DotImaging.Primitives2D, Version=2.0.0.0, Culture=neutral, processorArchitecture=MSIL">

<HintPath>..\packages\DotImaging.Primitives2D.2.5.3\lib\net45\DotImaging.Primitives2D.dll</HintPath>

</Reference>

<Reference Include="ExcelDataReader, Version=3.6.0.0, Culture=neutral, PublicKeyToken=93517dbe6a4012fa, processorArchitecture=MSIL">

<HintPath>..\packages\ExcelDataReader.3.6.0\lib\net45\ExcelDataReader.dll</HintPath>

</Reference>

<Reference Include="ExcelDataReader.DataSet, Version=3.6.0.0, Culture=neutral, PublicKeyToken=93517dbe6a4012fa, processorArchitecture=MSIL">

<HintPath>..\packages\ExcelDataReader.DataSet.3.6.0\lib\net35\ExcelDataReader.DataSet.dll</HintPath>

</Reference>

<Reference Include="Google.Protobuf, Version=3.14.0.0, Culture=neutral, PublicKeyToken=a7d26565bac4d604, processorArchitecture=MSIL">

<HintPath>..\packages\Google.Protobuf.3.14.0\lib\net45\Google.Protobuf.dll</HintPath>

</Reference>

<Reference Include="K4os.Compression.LZ4, Version=1.2.6.0, Culture=neutral, PublicKeyToken=2186fa9121ef231d, processorArchitecture=MSIL">

<HintPath>..\packages\K4os.Compression.LZ4.1.2.6\lib\net46\K4os.Compression.LZ4.dll</HintPath>

</Reference>

<Reference Include="K4os.Compression.LZ4.Streams, Version=1.2.6.0, Culture=neutral, PublicKeyToken=2186fa9121ef231d, processorArchitecture=MSIL">

<HintPath>..\packages\K4os.Compression.LZ4.Streams.1.2.6\lib\net46\K4os.Compression.LZ4.Streams.dll</HintPath>

</Reference>

<Reference Include="K4os.Hash.xxHash, Version=1.0.6.0, Culture=neutral, PublicKeyToken=32cd54395057cec3, processorArchitecture=MSIL">

<HintPath>..\packages\K4os.Hash.xxHash.1.0.6\lib\net46\K4os.Hash.xxHash.dll</HintPath>

</Reference>

<Reference Include="Microsoft.Win32.Primitives, Version=4.0.2.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

<HintPath>..\packages\Microsoft.Win32.Primitives.4.3.0\lib\net46\Microsoft.Win32.Primitives.dll</HintPath>

</Reference>

<Reference Include="Microsoft.Win32.Registry, Version=5.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

<HintPath>..\packages\Microsoft.Win32.Registry.5.0.0\lib\net461\Microsoft.Win32.Registry.dll</HintPath>

</Reference>

<Reference Include="MySQL.Data, Version=8.0.28.0, Culture=neutral, PublicKeyToken=c5687fc88969c44d, processorArchitecture=MSIL">

<HintPath>..\packages\MySQL.Data.8.0.28\lib\net452\MySQL.Data.dll</HintPath>

</Reference>

<Reference Include="nunit.framework, Version=3.13.1.0, Culture=neutral, PublicKeyToken=2638cd05610744eb, processorArchitecture=MSIL">

<HintPath>..\packages\NUnit.3.13.1\lib\net45\nunit.framework.dll</HintPath>

</Reference>

<Reference Include="System" />

<Reference Include="System.AppContext, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

<HintPath>..\packages\System.AppContext.4.3.0\lib\net46\System.AppContext.dll</HintPath>

</Reference>

<Reference Include="System.Buffers, Version=4.0.3.0, Culture=neutral, PublicKeyToken=cc7b13ffcd2ddd51, processorArchitecture=MSIL">

<HintPath>..\packages\System.Buffers.4.5.1\lib\net461\System.Buffers.dll</HintPath>

</Reference>

```
<Reference Include="System.ComponentModel" />

<Reference Include="System.ComponentModel.Composition" />

<Reference Include="System.ComponentModel.DataAnnotations" />

<Reference Include="System.Configuration" />

<Reference Include="System.Configuration.Install" />

<Reference Include="System.Console, Version=4.0.1.1, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

  <HintPath>..\packages\System.Console.4.3.1\lib\net46\System.Console.dll</HintPath>

</Reference>

<Reference Include="System.Core" />

<Reference Include="System.Diagnostics.DiagnosticSource, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=cc7b13ffcd2ddd51, processorArchitecture=MSIL">

<HintPath>..\packages\System.Diagnostics.DiagnosticSource.4.0.0\lib\net46\System.Diagnos
tics.DiagnosticSource.dll</HintPath>

</Reference>

<Reference Include="System.Globalization.Calendars, Version=4.0.1.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

<HintPath>..\packages\System.Globalization.Calendars.4.0.1\lib\net46\System.Globalization.
Calendars.dll</HintPath>

</Reference>
```

```
<Reference Include="System.IO.Compression, Version=4.1.2.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089, processorArchitecture=MSIL">
```

```
<HintPath>..\packages\System.IO.Compression.4.3.0\lib\net46\System.IO.Compression.dll</
HintPath>
```

```
</Reference>
```

```
<Reference Include="System.IO.Compression.FileSystem" />
```

```
<Reference Include="System.IO.Compression.ZipFile, Version=4.0.2.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089, processorArchitecture=MSIL">
```

```
<HintPath>..\packages\System.IO.Compression.ZipFile.4.3.0\lib\net46\System.IO.Compressi
on.ZipFile.dll</HintPath>
```

```
</Reference>
```

```
<Reference Include="System.IO.FileSystem, Version=4.0.2.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">
```

```
<HintPath>..\packages\System.IO.FileSystem.4.3.0\lib\net46\System.IO.FileSystem.dll</Hint
Path>
```

```
</Reference>
```

```
<Reference Include="System.IO.FileSystem.Primitives, Version=4.0.2.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">
```

```
<HintPath>..\packages\System.IO.FileSystem.Primitives.4.3.0\lib\net46\System.IO.FileSyste
m.Primitives.dll</HintPath>
```

```
</Reference>
```

```
<Reference Include="System.Management" />

<Reference Include="System.Memory, Version=4.0.1.1, Culture=neutral,
PublicKeyToken=cc7b13ffcd2ddd51, processorArchitecture=MSIL">

  <HintPath>..\packages\System.Memory.4.5.4\lib\net461\System.Memory.dll</HintPath>

</Reference>

<Reference Include="System.Net.Http, Version=4.1.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

  <HintPath>..\packages\System.Net.Http.4.1.0\lib\net46\System.Net.Http.dll</HintPath>

</Reference>

<Reference Include="System.Net.Sockets, Version=4.1.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

  <HintPath>..\packages\System.Net.Sockets.4.1.0\lib\net46\System.Net.Sockets.dll</HintPath
  >

</Reference>

<Reference Include="System.Numerics" />

<Reference Include="System.Numerics.Vectors, Version=4.1.4.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

  <HintPath>..\packages\System.Numerics.Vectors.4.5.0\lib\net46\System.Numerics.Vectors.dll
  </HintPath>

</Reference>
```

<Reference Include="System.Runtime.CompilerServices.Unsafe, Version=5.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

<HintPath>..\packages\System.Runtime.CompilerServices.Unsafe.5.0.0\lib\net45\System.Runtime.CompilerServices.Unsafe.dll</HintPath>

</Reference>

<Reference Include="System.Runtime.InteropServices.RuntimeInformation, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

<HintPath>..\packages\System.Runtime.InteropServices.RuntimeInformation.4.0.0\lib\net45\System.Runtime.InteropServices.RuntimeInformation.dll</HintPath>

</Reference>

<Reference Include="System.Security.AccessControl, Version=5.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

<HintPath>..\packages\System.Security.AccessControl.5.0.0\lib\net461\System.Security.AccessControl.dll</HintPath>

</Reference>

<Reference Include="System.Security.Cryptography.Algorithms, Version=4.1.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

<HintPath>..\packages\System.Security.Cryptography.Algorithms.4.2.0\lib\net461\System.Security.Cryptography.Algorithms.dll</HintPath>

</Reference>

<Reference Include="System.Security.Cryptography.Encoding, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

<HintPath>..\packages\System.Security.Cryptography.Encoding.4.0.0\lib\net46\System.Security.Cryptography.Encoding.dll</HintPath>

</Reference>

<Reference Include="System.Security.Cryptography.Primitives, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

<HintPath>..\packages\System.Security.Cryptography.Primitives.4.0.0\lib\net46\System.Security.Cryptography.Primitives.dll</HintPath>

</Reference>

<Reference Include="System.Security.Cryptography.X509Certificates, Version=4.1.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

<HintPath>..\packages\System.Security.Cryptography.X509Certificates.4.1.0\lib\net461\System.Security.Cryptography.X509Certificates.dll</HintPath>

</Reference>

<Reference Include="System.Security.Principal.Windows, Version=5.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

<HintPath>..\packages\System.Security.Principal.Windows.5.0.0\lib\net461\System.Security.Principal.Windows.dll</HintPath>

</Reference>

<Reference Include="System.Transactions" />


```
<Reference Include="System.Xml.Linq" />

<Reference Include="System.Data.DataSetExtensions" />

<Reference Include="Microsoft.CSharp" />

<Reference Include="System.Data" />

<Reference Include="System.Deployment" />

<Reference Include="System.Drawing" />

<Reference Include="System.Windows.Forms" />

<Reference Include="System.Xml" />

<Reference Include="System.Xml.ReaderWriter, Version=4.1.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL">

<HintPath>..\packages\System.Xml.ReaderWriter.4.3.1\lib\net46\System.Xml.ReaderWriter.
dll</HintPath>

</Reference>

<Reference Include="Ubiety.Dns.Core, Version=2.2.1.0, Culture=neutral,
PublicKeyToken=c5687fc88969c44d, processorArchitecture=MSIL">

<HintPath>..\packages\MySql.Data.8.0.28\lib\net452\Ubiety.Dns.Core.dll</HintPath>

</Reference>

<Reference Include="ZstdNet, Version=1.4.5.0, Culture=neutral,
PublicKeyToken=c5687fc88969c44d, processorArchitecture=MSIL">

<HintPath>..\packages\MySql.Data.8.0.28\lib\net452\ZstdNet.dll</HintPath>

</Reference>
```

</ItemGroup>

<ItemGroup>

<Compile Include="Global\Combinatorics.cs" />

<Compile Include="Global\EditForm.cs">

<SubType>Form</SubType>

</Compile>

<Compile Include="Global\EditForm.Designer.cs">

<DependentUpon>EditForm.cs</DependentUpon>

</Compile>

<Compile Include="Global\Info.cs">

<SubType>UserControl</SubType>

</Compile>

<Compile Include="Global\Info.Designer.cs">

<DependentUpon>Info.cs</DependentUpon>

</Compile>

<Compile Include="Global\MoveBetwenListboxesControl.cs">

<SubType>UserControl</SubType>

</Compile>

<Compile Include="Global\MoveBetwenListboxesControl.Designer.cs">

<DependentUpon>MoveBetwenListboxesControl.cs</DependentUpon>

```
</Compile>

<Compile Include="Global\SectionHeader.cs">

  <SubType>UserControl</SubType>

</Compile>

<Compile Include="Global\SectionHeader.Designer.cs">

  <DependentUpon>SectionHeader.cs</DependentUpon>

</Compile>

<Compile Include="Global\SideNavigationMenu.cs">

  <SubType>UserControl</SubType>

</Compile>

<Compile Include="Global\SideNavigationMenu.Designer.cs">

  <DependentUpon>SideNavigationMenu.cs</DependentUpon>

</Compile>

<Compile Include="Global\TreeNodes.cs" />

<Compile Include="Global\ExtensionMethods.cs" />

<Compile Include="Global\User.cs" />

<Compile Include="Global\WizardControl.cs">

  <SubType>UserControl</SubType>

</Compile>

<Compile Include="Global\WizardControl.Designer.cs">
```

<DependentUpon>WizardControl.cs</DependentUpon>

</Compile>

<Compile Include="Screens\About.cs">

<SubType>UserControl</SubType>

</Compile>

<Compile Include="Screens\About.Designer.cs">

<DependentUpon>About.cs</DependentUpon>

</Compile>

<Compile Include="Screens\AddColumnForm.cs">

<SubType>Form</SubType>

</Compile>

<Compile Include="Screens\AddColumnForm.Designer.cs">

<DependentUpon>AddColumnForm.cs</DependentUpon>

</Compile>

<Compile Include="Screens\Analyze.cs">

<SubType>UserControl</SubType>

</Compile>

<Compile Include="Screens\Analyze.Designer.cs">

<DependentUpon>Analyze.cs</DependentUpon>

</Compile>

```
<Compile Include="Screens\DataView.cs">

  <SubType>UserControl</SubType>

</Compile>

<Compile Include="Screens\DataView.Designer.cs">

  <DependentUpon>DataView.cs</DependentUpon>

</Compile>

<Compile Include="Screens\ExcelFileImportWizardControl.cs">

  <SubType>UserControl</SubType>

</Compile>

<Compile Include="Screens\ExcelFileImportWizardControl.Designer.cs">

  <DependentUpon>ExcelFileImportWizardControl.cs</DependentUpon>

</Compile>

<Compile Include="Screens\LoginForm.cs">

  <SubType>Form</SubType>

</Compile>

<Compile Include="Screens\LoginForm.Designer.cs">

  <DependentUpon>LoginForm.cs</DependentUpon>

</Compile>

<Compile Include="Screens\MainForm.cs">

  <SubType>Form</SubType>
```

```
</Compile>

<Compile Include="Screens\MainForm.Designer.cs">

  <DependentUpon>MainForm.cs</DependentUpon>

</Compile>

<Compile Include="Program.cs" />

<Compile Include="Properties\AssemblyInfo.cs" />

<Compile Include="Screens\CSVFileImportWizardControl.cs">

  <SubType>UserControl</SubType>

</Compile>

<Compile Include="Screens\CSVFileImportWizardControl.Designer.cs">

  <DependentUpon>CSVFileImportWizardControl.cs</DependentUpon>

</Compile>

<Compile Include="Screens\RemoveColumnsForm.cs">

  <SubType>Form</SubType>

</Compile>

<Compile Include="Screens\RemoveColumnsForm.Designer.cs">

  <DependentUpon>RemoveColumnsForm.cs</DependentUpon>

</Compile>

<Compile Include="Screens\ReportViewer.cs">

  <SubType>UserControl</SubType>
```

```
</Compile>

<Compile Include="Screens\ReportViewer.Designer.cs">

  <DependentUpon>ReportViewer.cs</DependentUpon>

</Compile>

<Compile Include="Screens\VideoPlayer.cs">

  <SubType>Form</SubType>

</Compile>

<Compile Include="Screens\VideoPlayer.Designer.cs">

  <DependentUpon>VideoPlayer.cs</DependentUpon>

</Compile>

<Compile Include="Tests\Bartlet.cs" />

<Compile Include="Tests\Bonferroni.cs" />

<Compile Include="Tests\BonferroniTest.cs" />

<Compile Include="Tests\DescriptiveTable.cs" />

<Compile Include="Tests\DMS.cs" />

<Compile Include="Tests\FDistribution.cs" />

<Compile Include="Tests\Friedman.cs" />

<Compile Include="Tests\FriedmanTest.cs" />

<Compile Include="Tests\FTest.cs" />

<Compile Include="Tests\GreenhouseGeisserEpsilon.cs" />
```

<Compile Include="Tests\GreenhouseGeisserEpsilonTest.cs" />

<Compile Include="Tests\Helpers\DescriptiveTableResultsTable.cs" />

<Compile Include="Tests\Helpers\EqualityOfVariancesTestsResultsTable.cs" />

<Compile Include="Tests\Helpers\OneGroupTestResultTable.cs" />

<Compile Include="Tests\Helpers\HypothesisTestsResultsTable.cs" />

<Compile Include="Tests\Helpers\NormalityTestsResultsTable.cs" />

<Compile Include="Tests\Helpers\ResultsTables.cs" />

<Compile Include="Tests\Helpers\ResultsTextSection.cs" />

<Compile Include="Tests\Helpers\ResultsText.cs" />

<Compile Include="Tests\Helpers\SphericityTestsResultsTable.cs" />

<Compile Include="Tests\HuynhFeldtEpsilon.cs" />

<Compile Include="Tests\HuynhFeldtEpsilonTest.cs" />

<Compile Include="Tests\KruskalWallis.cs" />

<Compile Include="Tests\KruskalWallisTest.cs" />

<Compile Include="Tests\LeveneTest.cs" />

<Compile Include="Tests\MannWhitney.cs" />

<Compile Include="Tests\ManWhitneyTest.cs" />

<Compile Include="Tests\Mauchly.cs" />

<Compile Include="Tests\MauchlyTest.cs" />

<Compile Include="Tests\OneWayAnova.cs" />


```
<Compile Include="Tests\OneWayAnovaTest.cs" />

<Compile Include="Tests\RepeatedMeasuresAnova.cs" />

<Compile Include="Tests\RepeatedMeasuresAnovaTest.cs" />

<Compile Include="Tests\Helpers\ResultsTable.cs" />

<Compile Include="Tests\TTest.cs" />

<Compile Include="Tests\KolmogorovSmirnovTest.cs" />

<Compile Include="Tests\PairedTTest.cs" />

<Compile Include="Tests\ShapiroTest.cs" />

<Compile Include="Tests\StatisticalTest.cs" />

<Compile Include="Tests\TwoSampleTTestTest.cs" />

<Compile Include="Tests\Welch.cs" />

<Compile Include="Tests\WelchTest.cs" />

<Compile Include="Tests\WilcoxonTest.cs" />

<Compile Include="Unit Tests\StatisticsTestsTest.cs" />

<EmbeddedResource Include="Global\EditForm.resx">

  <DependentUpon>EditForm.cs</DependentUpon>

  <SubType>Designer</SubType>

</EmbeddedResource>

<EmbeddedResource Include="Global\Info.resx">

  <DependentUpon>Info.cs</DependentUpon>
```

<SubType>Designer</SubType>

</EmbeddedResource>

<EmbeddedResource Include="Global\MoveBetwenListboxesControl.resx">

<DependentUpon>MoveBetwenListboxesControl.cs</DependentUpon>

<SubType>Designer</SubType>

</EmbeddedResource>

<EmbeddedResource Include="Global\SectionHeader.resx">

<DependentUpon>SectionHeader.cs</DependentUpon>

</EmbeddedResource>

<EmbeddedResource Include="Global\SideNavigationMenu.resx">

<DependentUpon>SideNavigationMenu.cs</DependentUpon>

</EmbeddedResource>

<EmbeddedResource Include="Global\WizardControl.resx">

<DependentUpon>WizardControl.cs</DependentUpon>

<SubType>Designer</SubType>

</EmbeddedResource>

<EmbeddedResource Include="Properties\Resources.resx">

<Generator>ResXFileCodeGenerator</Generator>

<LastGenOutput>Resources.Designer.cs</LastGenOutput>

<SubType>Designer</SubType>

</EmbeddedResource>

<Compile Include="Properties\Resources.Designer.cs">

<AutoGen>True</AutoGen>

<DependentUpon>Resources.resx</DependentUpon>

<DesignTime>True</DesignTime>

</Compile>

<EmbeddedResource Include="Screens>About.resx">

<DependentUpon>About.cs</DependentUpon>

</EmbeddedResource>

<EmbeddedResource Include="Screens>AddColumnForm.resx">

<DependentUpon>AddColumnForm.cs</DependentUpon>

</EmbeddedResource>

<EmbeddedResource Include="Screens>Analyze.resx">

<DependentUpon>Analyze.cs</DependentUpon>

</EmbeddedResource>

<EmbeddedResource Include="Screens\CSVFileImportWizardControl.resx">

<DependentUpon>CSVFileImportWizardControl.cs</DependentUpon>

</EmbeddedResource>

<EmbeddedResource Include="Screens\DataView.resx">

<DependentUpon>DataView.cs</DependentUpon>

</EmbeddedResource>

<EmbeddedResource Include="Screens\ExcelFileImportWizardControl.resx">

<DependentUpon>ExcelFileImportWizardControl.cs</DependentUpon>

</EmbeddedResource>

<EmbeddedResource Include="Screens\LoginForm.resx">

<DependentUpon>LoginForm.cs</DependentUpon>

</EmbeddedResource>

<EmbeddedResource Include="Screens\MainForm.resx">

<DependentUpon>MainForm.cs</DependentUpon>

</EmbeddedResource>

<EmbeddedResource Include="Screens\RemoveColumnsForm.resx">

<DependentUpon>RemoveColumnsForm.cs</DependentUpon>

</EmbeddedResource>

<EmbeddedResource Include="Screens\ReportViewer.resx">

<DependentUpon>ReportViewer.cs</DependentUpon>

</EmbeddedResource>

<EmbeddedResource Include="Screens\VideoPlayer.resx">

<DependentUpon>VideoPlayer.cs</DependentUpon>

</EmbeddedResource>

<None Include="packages.config" />

```
<None Include="Properties\Settings.settings">

  <Generator>SettingsSingleFileGenerator</Generator>

  <LastGenOutput>Settings.Designer.cs</LastGenOutput>

</None>

<Compile Include="Properties\Settings.Designer.cs">

  <AutoGen>True</AutoGen>

  <DependentUpon>Settings.settings</DependentUpon>

  <DesignTimeSharedInput>True</DesignTimeSharedInput>

</Compile>

</ItemGroup>

<ItemGroup>

  <None Include="App.config" />

</ItemGroup>

<ItemGroup>

  <None Include="Resources\001-csv-file-format-symbol.png" />

</ItemGroup>

<ItemGroup>

  <None Include="Resources\002-xlsx-file-format.png" />

</ItemGroup>

<ItemGroup>
```

<None Include="Resources\003-import.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\004-server.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\005-chart.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\006-type.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\007-browser-visualization.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\008-analysis.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\009-bar-chart.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\ok.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\cancel.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\database.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\import_excel.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\clipboard.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\next.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\previous.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\type_data.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\analysis.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\import.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\import_csv.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\oraculo.ico" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\find.png" />

</ItemGroup>

<ItemGroup>


```
<None Include="Resources\delete.png" />

</ItemGroup>

<ItemGroup>

  <None Include="Resources\add.png" />

</ItemGroup>

<ItemGroup>

  <None Include="Resources\information.png" />

</ItemGroup>

<ItemGroup>

  <BootstrapperPackage Include=".NETFramework,Version=v4.6.1">

    <Visible>False</Visible>

    <ProductName>Microsoft .NET Framework 4.6.1 %28x86 e x64%29</ProductName>

    <Install>true</Install>

  </BootstrapperPackage>

  <BootstrapperPackage Include="Microsoft.Net.Framework.3.5.SP1">

    <Visible>False</Visible>

    <ProductName>.NET Framework 3.5 SP1</ProductName>

    <Install>false</Install>

  </BootstrapperPackage>

</ItemGroup>
```

```
<ItemGroup>

  <COMReference Include="Microsoft.Office.Core">

    <Guid>{2DF8D04C-5BFA-101B-BDE5-00AA0044DE52}</Guid>

    <VersionMajor>2</VersionMajor>

    <VersionMinor>8</VersionMinor>

    <Lcid>0</Lcid>

    <WrapperTool>primary</WrapperTool>

    <Isolated>False</Isolated>

    <EmbedInteropTypes>True</EmbedInteropTypes>

  </COMReference>

  <COMReference Include="VBIDE">

    <Guid>{0002E157-0000-0000-C000-000000000046}</Guid>

    <VersionMajor>5</VersionMajor>

    <VersionMinor>3</VersionMinor>

    <Lcid>0</Lcid>

    <WrapperTool>primary</WrapperTool>

    <Isolated>False</Isolated>

    <EmbedInteropTypes>True</EmbedInteropTypes>

  </COMReference>

</ItemGroup>
```

<ItemGroup>

<None Include="Resources\home.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\DataByColumns.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\DataByLines.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\001-edit-table.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\002-home.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\003-magic-wand.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\004-csv.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\005-xls.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\006-next.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\007-previous.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\008-delete-row.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\009-menu.png" />

</ItemGroup>

<ItemGroup>

<None Include="Resources\file.png" />

</ItemGroup>

<ItemGroup>

```

    <None Include="Resources\logo aprovada.png" />

</ItemGroup>

<ItemGroup>

    <None Include="Resources\add-column.png" />

</ItemGroup>

<ItemGroup>

    <None Include="Resources\delete-column.png" />

</ItemGroup>

<ItemGroup>

    <None Include="Resources\about.png" />

</ItemGroup>

<ItemGroup>

    <None Include="Resources\video_print.png" />

</ItemGroup>

<Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />

<Import Project="..\packages\Accord.3.8.0\build\Accord.targets"
Condition="Exists('..\packages\Accord.3.8.0\build\Accord.targets')" />

<Target Name="EnsureNuGetPackageBuildImports" BeforeTargets="PrepareForBuild">

    <PropertyGroup>

```

<ErrorText>Este projeto faz referência a pacotes do NuGet que não estão presentes neste computador. Use a Restauração de Pacotes do NuGet para baixá-los. Para obter mais

informações, consulte <http://go.microsoft.com/fwlink/?LinkID=322105>. O arquivo ausente é {0}.</ErrorText>

</PropertyGroup>

<Error Condition="!Exists('..\packages\Accord.3.8.0\build\Accord.targets)'" Text="\$([System.String]::Format('\${ErrorText}', '..\packages\Accord.3.8.0\build\Accord.targets'))" />

<Error Condition="!Exists('..\packages\NUnit3TestAdapter.3.17.0\build\net35\NUnit3TestAdapter.props)'" Text="\$([System.String]::Format('\${ErrorText}', '..\packages\NUnit3TestAdapter.3.17.0\build\net35\NUnit3TestAdapter.props'))" />

<Error Condition="!Exists('..\packages\NETStandard.Library.2.0.3\build\netstandard2.0\NETStandard.Library.targets)'" Text="\$([System.String]::Format('\${ErrorText}', '..\packages\NETStandard.Library.2.0.3\build\netstandard2.0\NETStandard.Library.targets'))" />

<Error Condition="!Exists('..\packages\NUnit.3.13.1\build\NUnit.props)'" Text="\$([System.String]::Format('\${ErrorText}', '..\packages\NUnit.3.13.1\build\NUnit.props'))" />

</Target>

<Import Project="..\packages\NETStandard.Library.2.0.3\build\netstandard2.0\NETStandard.Library.targets" Condition="Exists('..\packages\NETStandard.Library.2.0.3\build\netstandard2.0\NETStandard.Library.targets)'" />

</Project>

Oraculo\Program.cs

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

using Oraculo.Global;

namespace Oraculo
{
    static class Program
    {
        /// <summary>
        /// Ponto de entrada principal para o aplicativo.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();

            Application.SetCompatibleTextRenderingDefault(false);

```

```
        Application.Run(new MainForm());  
    }  
}  
}
```

Oraculo\Global\Combinatorics.cs

```
?using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;  
  
namespace Oraculo.Global  
{  
  
    public static class Combinatorics  
    {  
  
        //Faz a combinação binominal entre k e n  
  
        public static IEnumerable<int[]> Combinations(int k, int n)  
  
        {
```



```
var result = new int[k];

var stack = new Stack<int>();

stack.Push(1);

while (stack.Count > 0)

{

    var index = stack.Count - 1;

    var value = stack.Pop();

    while (value <= n)

    {

        result[index++] = value++;

        stack.Push(value);

        if (index == k)

        {

            yield return result;

            break;

        }

    }

}
```

```
    }  
  }  
}
```

Oraculo\Global\EditForm.cs

```
?using System;  
  
using System.Collections.Generic;  
  
using System.ComponentModel;  
  
using System.Data;  
  
using System.Drawing;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;  
  
using System.Windows.Forms;  
  
namespace Oraculo.Global  
{  
  
    public partial class EditForm : Form  
  
    {
```

```
#region Properties
```

```
public bool HasModified { get { return (this.DialogResult == DialogResult.OK); } }
```

```
#endregion
```

```
#region Methods
```

```
//abre a caixa de diálogo apenas para exibição (somente leitura)
```

```
public virtual void ShowReadOnly ()
```

```
{
```

```
    Clear();
```

```
    DisableControls();
```

```
    Fill();
```

```
    ShowDialog();
```

```
}
```

```
//abre a caixa de diálogo para inserção
```

```
public virtual void ShowEditMode(object value)
```

```
{
```

```
Clear();

Fill();

ShowDialog();

}

//abre a caixa de diálogo para inserção

public virtual void ShowInsertMode()

{

    Clear();

    ShowDialog();

}

//valida os valores informados e grava as alterações no objeto

protected new virtual bool Validate()

{

    return true;

}

//preenche os componentes

protected virtual void Fill()
```

```
{  
  
}  
  
//limpa e inicializa os componentes  
  
protected virtual void Clear()  
  
{  
  
}  
  
//  
  
protected virtual void DisableControls()  
  
{  
  
    OKButton.Visible = false;  
  
    CancelButton.Text = "Fechar";  
  
}  
  
#endregion
```

```
public EditForm()
{
    InitializeComponent();
}

private void OKButton_Click(object sender, EventArgs e)
{
    if (!Validate()) { return; } else { this.DialogResult = DialogResult.OK; }
}

private void CancelButton_Click(object sender, EventArgs e)
{
    this.DialogResult = DialogResult.Cancel;
}
}
}
```

Oraculo\Global\EditForm.Designer.cs

?namespace Oraculo.Global

```

{

partial class EditForm

{

    /// <summary>

    /// Required designer variable.

    /// </summary>

    private System.ComponentModel.IContainer components = null;

    /// <summary>

    /// Clean up any resources being used.

    /// </summary>

    /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>

    protected override void Dispose(bool disposing)

    {

        if (disposing && (components != null))

        {

            components.Dispose();

        }

        base.Dispose(disposing);

```

```
}
```

```
#region Windows Form Designer generated code
```

```
/// <summary>
```

```
/// Required method for Designer support - do not modify
```

```
/// the contents of this method with the code editor.
```

```
/// </summary>
```

```
private void InitializeComponent()
```

```
{
```

```
    this.CancelButton = new System.Windows.Forms.Button();
```

```
    this.OKButton = new System.Windows.Forms.Button();
```

```
    this.SuspendLayout();
```

```
    //
```

```
    // CancelButton
```

```
    //
```

```
    this.CancelButton.Anchor =  
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |  
System.Windows.Forms.AnchorStyles.Right)));
```

```
    this.CancelButton.BackColor = System.Drawing.Color.Gainsboro;
```

```
    this.CancelButton.DialogResult = System.Windows.Forms.DialogResult.Cancel;
```



```

this.CancelButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;

this.CancelButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.CancelButton.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.CancelButton.Location = new System.Drawing.Point(504, 384);

this.CancelButton.Margin = new System.Windows.Forms.Padding(10, 20, 20, 20);

this.CancelButton.Name = "CancelButton";

this.CancelButton.Size = new System.Drawing.Size(91, 28);

this.CancelButton.TabIndex = 10;

this.CancelButton.Text = "Cancelar";

this.CancelButton.UseVisualStyleBackColor = false;

this.CancelButton.Click += new System.EventHandler(this.CancelButton_Click);

//

// OKButton

//

this.OKButton.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));

this.OKButton.BackColor = System.Drawing.Color.Gainsboro;

this.OKButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;

this.OKButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.OKButton.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

```

```
this.OKButton.Location = new System.Drawing.Point(403, 384);

this.OKButton.Margin = new System.Windows.Forms.Padding(20, 20, 0, 20);

this.OKButton.Name = "OKButton";

this.OKButton.Size = new System.Drawing.Size(91, 28);

this.OKButton.TabIndex = 9;

this.OKButton.Text = "OK";

this.OKButton.UseVisualStyleBackColor = false;

this.OKButton.Click += new System.EventHandler(this.OKButton_Click);

//

// EditForm

//

this.AcceptButton = this.OKButton;

this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(624, 441);

this.Controls.Add(this.CancelButton);

this.Controls.Add(this.OKButton);

this.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.MinimumSize = new System.Drawing.Size(347, 340);
```

```
        this.Name = "EditForm";

        this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;

        this.Text = "EditForm";

        this.ResumeLayout(false);

    }

#endregion

    protected new System.Windows.Forms.Button CancelButton;

    protected System.Windows.Forms.Button OKButton;

}

}

Oraculo\Global\ExtensionMethods.cs

?using System;

using System.Windows.Forms;

using System.Drawing;

using System.Runtime.InteropServices;

using System.Collections.Generic;
```

```

using System.Data;

namespace Oraculo.Global
{
    public static class ObjectExtensions
    {
        //testa se um valor é nulo

        public static bool IsNullOrEquivalent(this object value)
        {
            return value == null

                || value is DBNull

                || DBNull.Value.Equals(value)

                || string.IsNullOrWhiteSpace(value.ToString());
        }
    }

    public static class ArrayExtensions
    {
        public static T[] Concatenate<T>(this T[] array1, T[] array2)
        {

```

```

    T[] result = new T[array1.Length + array2.Length];

    array1.CopyTo(result, 0);

    array2.CopyTo(result, array1.Length);

    return result;

}

}

```

```

public static class DataTableExtensions

{

    //Conta quantos valores numéricos existem nas linhas de uma determinada coluna de um
datatable

    public static int CountNumericValuesAt(this DataTable dt, int column)

    {

        int n = 0;

        foreach (DataRow dr in dt.Rows)

            if (!dr[column].IsNullOrEquivalent()) n++;

        return n;

    }

    //Conta quantos valores numéricos existem nas linhas de uma determinada coluna de um
datatable

```

```

public static int CountNumericValuesAt(this DataTable dt, string column)
{
    int n = 0;

    foreach (DataRow dr in dt.Rows)
        if (!dr[column].IsNullOrEquivalent()) n++;

    return n;
}

```

//obtem uma lista de todos os index de linhas que estejam com valores nulos nas colunas selecionadas

```

public static List<int> GetEmptyRowsList(this DataTable dt, List<string> columns)
{
    List<int> emptyrows = new List<int>();

    foreach (DataRow linha in dt.Rows)
    {
        foreach (string coluna in columns)
        {
            if (linha[coluna].IsNullOrEquivalent())
            {
                if (!emptyrows.Contains(dt.Rows.IndexOf(linha)))

```

```

        {
            emptyrows.Add(dt.Rows.IndexOf(linha));
        }
    }
}
}

return emptyrows;
}

```

```

public static double[][] ToBidimensionalDoubleArray(this DataTable dt, List<string>
columns)

```

```

{
    double[][] dr = new double[columns.Count][];

    //percorre cada coluna selecionada do dataset

    foreach (string coluna in columns)
    {
        //cria variável para armazenar a lista

        List<double> dl = new List<double>();
    }
}

```

```

//percorre cada linha do dataset

foreach (DataRow linha in dt.Rows)

{

    if (!linha[coluna].IsNullOrEquivalent())

        dl.Add(Convert.ToDouble(linha[coluna]));

}

//converte para array

dr[columns.IndexOf(coluna)] = dl.ToArray();

}

return dr;

}

```

```

public static double[][] ToBidimensionalDoubleArray (this DataTable dt, List<string>
columns, List<int> emptyrows)

```

```

{

    double[][] dr = new double[columns.Count][];

//percorre cada coluna selecionada do dataset

```



```

foreach (string coluna in columns)
{
    //cria variável para armazenar a lista

    List<double> dl = new List<double>();

    //percorre cada linha do dataset

    foreach (DataRow linha in dt.Rows)
    {
        //insere apenas as linhas que não tenham valores nulos

        if (!emptyrows.Contains(dt.Rows.IndexOf(linha)))
        {
            dl.Add(Convert.ToDouble(linha[coluna]));
        }
    }

    //converte para array

    dr[columns.IndexOf(coluna)] = dl.ToArray();
}

return dr;

```

```
}
```

```
public static bool IsBalanced(this Double[][] array)
```

```
{
```

```
    double size = 0;
```

```
    for (int i = 0; i < array.Length; i++)
```

```
    {
```

```
        if (i == 0)
```

```
        {
```

```
            size = array[i].Length;
```

```
        }
```

```
    else
```

```
    {
```

```
        if (array[i].Length != size)
```

```
        {
```

```
            return false;
```

```
        }
```

```
    }
```

```
}
```

```

return true;

}

public static bool IsEqualTo(this DataTable dt, DataTable otherdt)
{
    //testa se alguma delas está nula ou vazia

    if (dt == null || otherdt == null) return false;

    if (dt.Rows.Count == 0 || otherdt.Rows.Count == 0) return false;

    //testa se tem número diferente de linhas

    if (dt.Rows.Count != otherdt.Rows.Count) return false;

    //testa se as linhas são iguais

    for (int i = 0; i < dt.Rows.Count; i++)

        for (int c = 0; c < dt.Columns.Count; c++)

            if (!Equals(dt.Rows[i][c], otherdt.Rows[i][c]))

                return false;

    //foreach (DataRow drA in dt.Rows)

```

```

//{

//  foreach (DataRow drB in otherdt.Rows)

//  {

//    if (drA.ItemArray != drB.ItemArray)

//    {

//      return false;

//    }

//  }

//}

return true;

}

}

```

```

public static class RichTextBoxExtensions

{

    public static void SetInnerMargins(this TextBoxBase textBox, int left, int top, int right,
int bottom)

    {

        var rect = textBox.GetFormattingRect();

```

```
var newRect = new Rectangle(left, top, rect.Width - left - right, rect.Height - top -
bottom);

textBox.SetFormattingRect(newRect);

}
```

```
[StructLayout(LayoutKind.Sequential)]
```

```
private struct RECT
```

```
{
```

```
public readonly int Left;
```

```
public readonly int Top;
```

```
public readonly int Right;
```

```
public readonly int Bottom;
```

```
private RECT(int left, int top, int right, int bottom)
```

```
{
```

```
Left = left;
```

```
Top = top;
```

```
Right = right;
```

```
Bottom = bottom;
```

```

    }

    public RECT(Rectangle r) : this(r.Left, r.Top, r.Right, r.Bottom)

    {

    }

}

```

```

[DllImport(@"User32.dll", EntryPoint = @"SendMessage", CharSet = CharSet.Auto)]

private static extern int SendMessageRefRect(IntPtr hWnd, uint msg, int wParam, ref
RECT rect);

```

```

[DllImport(@"user32.dll", EntryPoint = @"SendMessage", CharSet = CharSet.Auto)]

private static extern int SendMessage(IntPtr hwnd, int wParam, IntPtr lParam, ref
Rectangle lParam);

```

```

private const int EmGetrect = 0xB2;

```

```

private const int EmSetrect = 0xB3;

```

```

private static void SetFormattingRect(this TextBoxBase textbox, Rectangle rect)

```

```

{

    var rc = new RECT(rect);

```

```

        SendMessageRefRect(textbox.Handle, EmSetrect, 0, ref rc);
    }

    private static Rectangle GetFormattingRect(this TextBoxBase textbox)
    {
        var rect = new Rectangle();

        SendMessage(textbox.Handle, EmGetrect, (IntPtr)0, ref rect);

        return rect;
    }
}
}
}

```

Oraculo\Global\Info.cs

```

?using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

```

```
using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Oraculo.Global

{

    public partial class Info : UserControl

    {

        public string Message { get; set; }

        public bool ShowOnMove = true;

        public Info()

        {

            InitializeComponent();

        }

        private void Info_MouseClick(object sender, MouseEventArgs e)

        {

            toolTip1.Show(Message,this);

        }

    }

}
```



```
private void Info_MouseHover(object sender, EventArgs e)

{

    if (ShowOnMove) toolTip1.Show(Message, this);

}

}

}
```

Oraculo\Global\Info.Designer.cs

```
?namespace Oraculo.Global

{

    partial class Info

    {

        /// <summary>

        /// Variável de designer necessária.

        /// </summary>

        private System.ComponentModel.IContainer components = null;

        /// <summary>
```

```

/// Limpar os recursos que estão sendo usados.

/// </summary>

/// <param name="disposing">true se for necessário descartar os recursos gerenciados;
caso contrário, false.</param>

protected override void Dispose(bool disposing)

{

    if (disposing && (components != null))

    {

        components.Dispose();

    }

    base.Dispose(disposing);

}

#region Código gerado pelo Designer de Componentes

/// <summary>

/// Método necessário para suporte ao Designer - não modifique

/// o conteúdo deste método com o editor de código.

/// </summary>

private void InitializeComponent()

```

```
{  
  
    this.components = new System.ComponentModel.Container();  
  
    this.toolTip1 = new System.Windows.Forms.ToolTip(this.components);  
  
    this.SuspendLayout();  
  
    //  
  
    // toolTip1  
  
    //  
  
    this.toolTip1.ToolTipIcon = System.Windows.Forms.ToolTipIcon.Info;  
  
    this.toolTip1.ToolTipTitle = "Ajuda";  
  
    //  
  
    // Info  
  
    //  
  
    this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);  
  
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;  
  
    this.BackgroundImage = global::Oraculo.Properties.Resources.information;  
  
    this.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Center;  
  
    this.DoubleBuffered = true;  
  
    this.Name = "Info";  
  
    this.Size = new System.Drawing.Size(21, 21);  
  
}
```

```
        this.MouseClick += new
System.Windows.Forms.MouseEventHandler(this.Info_MouseClick);

        this.MouseHover += new System.EventHandler(this.Info_MouseHover);

        this.ResumeLayout(false);

    }

#endregion

    private System.Windows.Forms.ToolTip toolTip1;
}
}
```

Oraculo\Global\MoveBetwenListboxesControl.cs

```
?using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;
```

```

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Oraculo.Global
{
    public delegate void ListBoxItemsChangedEvent();

    public partial class MoveBetwenListboxesControl : UserControl
    {
        private List<string> OriginalList;

        public MoveBetwenListboxesControl()
        {
            InitializeComponent();
        }

        public string UnselectedCaption
        {
            get

```

```
{  
    return UnselectedLabel.Text;  
}  
  
set  
  
{  
    UnselectedLabel.Text = value;  
}  
}
```

```
public string SelectedCaption
```

```
{  
    get  
  
    {  
        return SelectedLabel.Text;  
    }  
  
    set  
  
    {  
        SelectedLabel.Text = value;  
    }  
}
```

```
public event ListBoxItemsChangedEvent SelectedListBoxItemsChanged;
```

```
// Move selected items from one ListBox to another.
```

```
private void MoveSelectedItems(ListBox lstFrom, ListBox lstTo)
```

```
{
```

```
    while (lstFrom.SelectedItems.Count > 0)
```

```
    {
```

```
        string item = (string)lstFrom.SelectedItems[0];
```

```
        //decide se intere antes ou depois
```

```
        if (lstTo.Items.Count > 0)
```

```
        {
```

```
            foreach (string i in lstTo.Items)
```

```
            {
```

```
                if (OriginalList.IndexOf(item) < OriginalList.IndexOf(i))
```

```
                {
```

```
                    lstTo.Items.Insert(lstTo.Items.IndexOf(i), item);
```

```
                    break;
```

```
                }
```

```
        else
        {
            if (lstTo.Items.IndexOf(i) == lstTo.Items.Count - 1)
            {
                lstTo.Items.Add(item);

                break;
            }
        }
    }
}

else
{
    lstTo.Items.Add(item);
}

lstFrom.Items.Remove(item);
}

SelectedListBoxItemsChanged.Invoke();

SetButtonsEditable();
```



```

}

// Move all items from one ListBox to another.

private void MoveAllItems(ListBox lstFrom, ListBox lstTo)
{
    lstTo.Items.AddRange(lstFrom.Items);

    lstFrom.Items.Clear();

    SelectedListBoxItemsChanged.Invoke();

    SetButtonsEditable();
}

// Enable and disable buttons.

private void SetButtonsEditable()
{
    SelectButton.Enabled = (UnselectedListBox.SelectedItems.Count > 0);

    SelectAllButton.Enabled = (UnselectedListBox.Items.Count > 0);

    DeselectButton.Enabled = (SelectedListBox.SelectedItems.Count > 0);

    DeselectAllButton.Enabled = (SelectedListBox.Items.Count > 0);
}

```

```

public void FillListBoxFromData(ListBox unsel, ListBox sel, DataTable data, string
ignoreItem)
{
    //recarrega o listBox das variáveis não selecionadas

    unsel.Items.Clear();

    OriginalList = new List<string>();

    foreach (DataColumn item in data.Columns)
    {
        if (!item.ColumnName.Equals(ignoreItem))
        {
            unsel.Items.Add(item.ColumnName);

            OriginalList.Add(item.ColumnName);
        }
    }

    //recarrega o listBox das variáveis selecionadas

    sel.Items.Clear();

    SetButtonsEditable();
}

```

```
        SelectedListBoxItemsChanged.Invoke();
    }

    private void SelectButton_Click(object sender, EventArgs e)
    {
        MoveSelectedItems(UnselectedListBox, SelectedListBox);
    }

    private void DeselectButton_Click(object sender, EventArgs e)
    {
        MoveSelectedItems(SelectedListBox, UnselectedListBox);
    }

    private void SelectAllButton_Click(object sender, EventArgs e)
    {
        MoveAllItems(UnselectedListBox, SelectedListBox);
    }

    private void DeselectAllButton_Click(object sender, EventArgs e)
    {
```

```

        MoveAllItems(SelectedListBox, UnselectedListBox);
    }

private void SelectedListBox_SelectedIndexChanged(object sender, EventArgs e)
{
    SetButtonsEditable();
}

private void MoveBetwenListboxesControl_Load(object sender, EventArgs e)
{
    SetButtonsEditable();
}

private void UnselectedListBox_SelectedIndexChanged(object sender, EventArgs e)
{
    SetButtonsEditable();
}
}
}

```

Oraculo\Global\MoveBetwenListboxesControl.Designer.cs

```
?namespace Oraculo.Global
```

```
{
```

```
partial class MoveBetwenListboxesControl
```

```
{
```

```
/// <summary>
```

```
/// Variável de designer necessária.
```

```
/// </summary>
```

```
private System.ComponentModel.IContainer components = null;
```

```
/// <summary>
```

```
/// Limpar os recursos que estão sendo usados.
```

```
/// </summary>
```

```
/// <param name="disposing">true se for necessário descartar os recursos gerenciados;  
caso contrário, false.</param>
```

```
protected override void Dispose(bool disposing)
```

```
{
```

```
if (disposing && (components != null))
```

```
{
```

```

        components.Dispose();

    }

    base.Dispose(disposing);

}

#region Código gerado pelo Designer de Componentes

/// <summary>

/// Método necessário para suporte ao Designer - não modifique

/// o conteúdo deste método com o editor de código.

/// </summary>

private void InitializeComponent()

{

    this.DeselectAllButton = new System.Windows.Forms.Button();

    this.SelectAllButton = new System.Windows.Forms.Button();

    this.DeselectButton = new System.Windows.Forms.Button();

    this.SelectButton = new System.Windows.Forms.Button();

    this.SelectedListBox = new System.Windows.Forms.ListBox();

    this.SelectedLabel = new System.Windows.Forms.Label();

    this.UnselectedListBox = new System.Windows.Forms.ListBox();

```

```

this.UnselectedLabel = new System.Windows.Forms.Label();

this.tableLayoutPanel1 = new System.Windows.Forms.TableLayoutPanel();

this.tableLayoutPanel1.SuspendLayout();

this.SuspendLayout();

//

// DeselectAllButton

//

this.DeselectAllButton.BackColor = System.Drawing.Color.Gainsboro;

this.DeselectAllButton.FlatAppearance.BorderColor =
System.Drawing.Color.Gainsboro;

this.DeselectAllButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.DeselectAllButton.Location = new System.Drawing.Point(199, 126);

this.DeselectAllButton.Name = "DeselectAllButton";

this.DeselectAllButton.Size = new System.Drawing.Size(90, 29);

this.DeselectAllButton.TabIndex = 5;

this.DeselectAllButton.Text = "<<";

this.DeselectAllButton.UseVisualStyleBackColor = false;

this.DeselectAllButton.Click += new
System.EventHandler(this.DeselectAllButton_Click);

//

// SelectAllButton

```

```

//

this.SelectAllButton.BackColor = System.Drawing.Color.Gainsboro;

this.SelectAllButton.FlatAppearance.BorderColor =
System.Drawing.Color.Gainsboro;

this.SelectAllButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.SelectAllButton.Location = new System.Drawing.Point(199, 91);

this.SelectAllButton.Name = "SelectAllButton";

this.SelectAllButton.Size = new System.Drawing.Size(90, 29);

this.SelectAllButton.TabIndex = 4;

this.SelectAllButton.Text = ">>";

this.SelectAllButton.UseVisualStyleBackColor = false;

this.SelectAllButton.Click += new System.EventHandler(this.SelectAllButton_Click);

//

// DeselectButton

//

this.DeselectButton.BackColor = System.Drawing.Color.Gainsboro;

this.DeselectButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;

this.DeselectButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.DeselectButton.Location = new System.Drawing.Point(199, 56);

this.DeselectButton.Name = "DeselectButton";

```



```

this.DeselectButton.Size = new System.Drawing.Size(90, 29);

this.DeselectButton.TabIndex = 3;

this.DeselectButton.Text = "<";

this.DeselectButton.UseVisualStyleBackColor = false;

this.DeselectButton.Click += new System.EventHandler(this.DeselectButton_Click);

//

// SelectButton

//

this.SelectButton.BackColor = System.Drawing.Color.Gainsboro;

this.SelectButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;

this.SelectButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.SelectButton.Location = new System.Drawing.Point(199, 21);

this.SelectButton.Name = "SelectButton";

this.SelectButton.Size = new System.Drawing.Size(90, 29);

this.SelectButton.TabIndex = 2;

this.SelectButton.Text = ">";

this.SelectButton.UseVisualStyleBackColor = false;

this.SelectButton.Click += new System.EventHandler(this.SelectButton_Click);

//

// SelectedListBox

```

```

//

this.SelectedListBox.Dock = System.Windows.Forms.DockStyle.Fill;

this.SelectedListBox.FormattingEnabled = true;

this.SelectedListBox.ItemHeight = 15;

this.SelectedListBox.Location = new System.Drawing.Point(297, 21);

this.SelectedListBox.Name = "SelectedListBox";

this.tableLayoutPanel1.SetRowSpan(this.SelectedListBox, 5);

this.SelectedListBox.ScrollAlwaysVisible = true;

this.SelectedListBox.SelectionMode =
System.Windows.Forms.SelectionMode.MultiExtended;

this.SelectedListBox.Size = new System.Drawing.Size(191, 159);

this.SelectedListBox.TabIndex = 7;

this.SelectedListBox.SelectedIndexChanged += new
System.EventHandler(this.SelectedListBox_SelectedIndexChanged);

//

// SelectedLabel

//

this.SelectedLabel.AutoSize = true;

this.SelectedLabel.Location = new System.Drawing.Point(297, 0);

this.SelectedLabel.Name = "SelectedLabel";

this.SelectedLabel.Padding = new System.Windows.Forms.Padding(0, 0, 0, 3);

```

```

this.SelectedLabel.Size = new System.Drawing.Size(85, 18);

this.SelectedLabel.TabIndex = 6;

this.SelectedLabel.Text = "Selected Label:";

//

// UnselectedListBox

//

this.UnselectedListBox.Dock = System.Windows.Forms.DockStyle.Fill;

this.UnselectedListBox.FormattingEnabled = true;

this.UnselectedListBox.ItemHeight = 15;

this.UnselectedListBox.Location = new System.Drawing.Point(3, 21);

this.UnselectedListBox.Name = "UnselectedListBox";

this.tableLayoutPanel1.SetRowSpan(this.UnselectedListBox, 5);

this.UnselectedListBox.ScrollAlwaysVisible = true;

this.UnselectedListBox.SelectionMode =
System.Windows.Forms.SelectionMode.MultiExtended;

this.UnselectedListBox.Size = new System.Drawing.Size(190, 159);

this.UnselectedListBox.TabIndex = 1;

this.UnselectedListBox.SelectedIndexChanged += new
System.EventHandler(this.UnselectedListBox_SelectedIndexChanged);

//

// UnselectedLabel

```

```

//

this.UnselectedLabel.AutoSize = true;

this.UnselectedLabel.Location = new System.Drawing.Point(3, 0);

this.UnselectedLabel.Name = "UnselectedLabel";

this.UnselectedLabel.Padding = new System.Windows.Forms.Padding(0, 0, 0, 3);

this.UnselectedLabel.Size = new System.Drawing.Size(99, 18);

this.UnselectedLabel.TabIndex = 0;

this.UnselectedLabel.Text = "Unselected Label: ";

//

// tableLayoutPanel1

//

this.tableLayoutPanel1.ColumnCount = 3;

this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 50F));

this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Absolute, 98F));

this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 50F));

this.tableLayoutPanel1.Controls.Add(this.UnselectedListBox, 0, 1);

this.tableLayoutPanel1.Controls.Add(this.SelectedLabel, 2, 0);

this.tableLayoutPanel1.Controls.Add(this.DeselectAllButton, 1, 4);

```

```
this.tableLayoutPanel1.Controls.Add(this.UnselectedLabel, 0, 0);

this.tableLayoutPanel1.Controls.Add(this.SelectedListBox, 2, 1);

this.tableLayoutPanel1.Controls.Add(this.SelectButton, 1, 1);

this.tableLayoutPanel1.Controls.Add(this.DeselectButton, 1, 2);

this.tableLayoutPanel1.Controls.Add(this.SelectAllButton, 1, 3);

this.tableLayoutPanel1.Dock = System.Windows.Forms.DockStyle.Fill;

this.tableLayoutPanel1.Location = new System.Drawing.Point(0, 0);

this.tableLayoutPanel1.Name = "tableLayoutPanel1";

this.tableLayoutPanel1.RowCount = 6;

this.tableLayoutPanel1.RowStyles.Add(new System.Windows.Forms.RowStyle());

this.tableLayoutPanel1.RowStyles.Add(new System.Windows.Forms.RowStyle());

this.tableLayoutPanel1.RowStyles.Add(new System.Windows.Forms.RowStyle());

this.tableLayoutPanel1.RowStyles.Add(new System.Windows.Forms.RowStyle());

this.tableLayoutPanel1.RowStyles.Add(new System.Windows.Forms.RowStyle());

this.tableLayoutPanel1.RowStyles.Add(new System.Windows.Forms.RowStyle());

this.tableLayoutPanel1.Size = new System.Drawing.Size(491, 183);

this.tableLayoutPanel1.TabIndex = 16;

//

// MoveBetwenListboxesControl

//
```

```
this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.Controls.Add(this.tableLayoutPanel1);

this.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.Name = "MoveBetwenListboxesControl";

this.Size = new System.Drawing.Size(491, 183);

this.Load += new System.EventHandler(this.MoveBetwenListboxesControl_Load);

this.tableLayoutPanel1.ResumeLayout(false);

this.tableLayoutPanel1.PerformLayout();

this.ResumeLayout(false);

}
```

```
#endregion
```

```
private System.Windows.Forms.Button DeselectAllButton;
```

```
private System.Windows.Forms.Button SelectAllButton;
```

```
private System.Windows.Forms.Button DeselectButton;
```

```
private System.Windows.Forms.Button SelectButton;
```

```
public System.Windows.Forms.ListBox SelectedListBox;

public System.Windows.Forms.ListBox UnselectedListBox;

private System.Windows.Forms.TableLayoutPanel tableLayoutPanel1;

public System.Windows.Forms.Label SelectedLabel;

public System.Windows.Forms.Label UnselectedLabel;

}

}
```

Oraculo\Global\SectionHeader.cs

```
?using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;
```

```

namespace Oraculo.Global

{

    public partial class SectionHeader : UserControl

    {

        public string BreadCrumbText

        {

            get

            {

                return labelBreadcrumb.Text;

            }

            set

            {

                labelBreadcrumb.Text = value;

            }

        }

        public SectionHeader()

        {

            InitializeComponent();

            //labelVersion.Text = "Versão " + Application.ProductVersion;

            labelVersion.Text = "Versão 1.0";

```



```
    }  
  }  
}
```

Oraculo\Global\SectionHeader.Designer.cs

?

```
namespace Oraculo.Global
```

```
{
```

```
    partial class SectionHeader
```

```
    {
```

```
        /// <summary>
```

```
        /// Variável de designer necessária.
```

```
        /// </summary>
```

```
        private System.ComponentModel.IContainer components = null;
```

```
        /// <summary>
```

```
        /// Limpar os recursos que estão sendo usados.
```

```
        /// </summary>
```

```
/// <param name="disposing">true se for necessário descartar os recursos gerenciados;  
caso contrário, false.</param>
```

```
protected override void Dispose(bool disposing)
```

```
{  
  
    if (disposing && (components != null))  
  
    {  
  
        components.Dispose();  
  
    }  
  
    base.Dispose(disposing);  
  
}
```

```
#region Código gerado pelo Designer de Componentes
```

```
/// <summary>
```

```
/// Método necessário para suporte ao Designer - não modifique
```

```
/// o conteúdo deste método com o editor de código.
```

```
/// </summary>
```

```
private void InitializeComponent()
```

```
{  
  
    this.panelBreadcrumb = new System.Windows.Forms.Panel();
```

```
this.labelBreadcrumb = new System.Windows.Forms.Label();

this.panelTitle = new System.Windows.Forms.Panel();

this.labelVersion = new System.Windows.Forms.Label();

this.LabelTitle = new System.Windows.Forms.Label();

this.pictureBoxLogo = new System.Windows.Forms.PictureBox();

this.panelBreadcrumb.SuspendLayout();

this.panelTitle.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.pictureBoxLogo)).BeginInit();

this.SuspendLayout();

//

// panelBreadcrumb

//

this.panelBreadcrumb.BackColor = System.Drawing.Color.WhiteSmoke;

this.panelBreadcrumb.Controls.Add(this.labelBreadcrumb);

this.panelBreadcrumb.Dock = System.Windows.Forms.DockStyle.Top;

this.panelBreadcrumb.Location = new System.Drawing.Point(0, 96);

this.panelBreadcrumb.Margin = new System.Windows.Forms.Padding(0);

this.panelBreadcrumb.Name = "panelBreadcrumb";

this.panelBreadcrumb.Size = new System.Drawing.Size(640, 23);

this.panelBreadcrumb.TabIndex = 9;
```

```
//  
  
// labelBreadcrumb  
  
//  
  
this.labelBreadcrumb.AutoSize = true;  
  
this.labelBreadcrumb.Font = new System.Drawing.Font("Segoe UI", 8.25F,  
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));  
  
this.labelBreadcrumb.ForeColor = System.Drawing.SystemColors.ControlText;  
  
this.labelBreadcrumb.Location = new System.Drawing.Point(20, 5);  
  
this.labelBreadcrumb.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);  
  
this.labelBreadcrumb.Name = "labelBreadcrumb";  
  
this.labelBreadcrumb.Size = new System.Drawing.Size(35, 13);  
  
this.labelBreadcrumb.TabIndex = 16;  
  
this.labelBreadcrumb.Text = "Início";  
  
//  
  
// panelTitle  
  
//  
  
this.panelTitle.BackColor = System.Drawing.Color.White;  
  
this.panelTitle.Controls.Add(this.labelVersion);  
  
this.panelTitle.Controls.Add(this.LabelTitle);  
  
this.panelTitle.Controls.Add(this.pictureBoxLogo);
```

```
this.panelTitle.Dock = System.Windows.Forms.DockStyle.Top;

this.panelTitle.Location = new System.Drawing.Point(0, 0);

this.panelTitle.Margin = new System.Windows.Forms.Padding(0);

this.panelTitle.Name = "panelTitle";

this.panelTitle.Size = new System.Drawing.Size(640, 96);

this.panelTitle.TabIndex = 8;

//

// labelVersion

//

this.labelVersion.AutoSize = true;

this.labelVersion.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.labelVersion.Location = new System.Drawing.Point(102, 53);

this.labelVersion.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);

this.labelVersion.Name = "labelVersion";

this.labelVersion.Size = new System.Drawing.Size(59, 15);

this.labelVersion.TabIndex = 15;

this.labelVersion.Text = "Versão 1.0";

//

// LabelTitle
```

```

//

this.LabelTitle.AutoSize = true;

this.LabelTitle.Font = new System.Drawing.Font("Segoe UI", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.LabelTitle.Location = new System.Drawing.Point(98, 28);

this.LabelTitle.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);

this.LabelTitle.Name = "LabelTitle";

this.LabelTitle.Size = new System.Drawing.Size(168, 25);

this.LabelTitle.TabIndex = 13;

this.LabelTitle.Text = "Oráculo Estadístico";

//

// pictureBoxLogo

//

this.pictureBoxLogo.Image = global::Oraculo.Properties.Resources.logo;

this.pictureBoxLogo.Location = new System.Drawing.Point(20, 0);

this.pictureBoxLogo.Margin = new System.Windows.Forms.Padding(0);

this.pictureBoxLogo.Name = "pictureBoxLogo";

this.pictureBoxLogo.Size = new System.Drawing.Size(76, 96);

this.pictureBoxLogo.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.Zoom;

this.pictureBoxLogo.TabIndex = 14;

```

```

this.pictureBoxLogo.TabStop = false;

//

// SectionHeader

//

this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.BackColor = System.Drawing.Color.White;

this.Controls.Add(this.panelBreadcrumb);

this.Controls.Add(this.panelTitle);

this.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.Margin = new System.Windows.Forms.Padding(0);

this.Name = "SectionHeader";

this.Size = new System.Drawing.Size(640, 119);

this.panelBreadcrumb.ResumeLayout(false);

this.panelBreadcrumb.PerformLayout();

this.panelTitle.ResumeLayout(false);

this.panelTitle.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.pictureBoxLogo)).EndInit();

this.ResumeLayout(false);

```

```
}

#endregion

private System.Windows.Forms.Panel panelBreadcrumb;

private System.Windows.Forms.Label labelBreadcrumb;

private System.Windows.Forms.Panel panelTitle;

private System.Windows.Forms.Label labelVersion;

private System.Windows.Forms.Label LabelTitle;

private System.Windows.Forms.PictureBox pictureBoxLogo;

}

}
```

Oraculo\Global\SideNavigationMenu.cs

```
?using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;
```



```

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Oraculo.Global
{
    public partial class SideNavigationMenu : UserControl
    {
        private bool IsMenuHiden { get; set; }

        public Panel SelectedItem { get; set; }

        private Color SelectedItemBackColor =
System.Drawing.ColorTranslator.FromHtml("#E5E5E5");

        private Color UnselectedItemBackColor = Color.WhiteSmoke;

        private Color MouseOverItemBackColor = Color.LightGray;

        public delegate void SelectedItemChangedEvent();

        public event SelectedItemChangedEvent SelectedItemChanged;

```

```
public SideNavigationMenu()

{

    InitializeComponent();

}

private void HideMenu()

{

    for (int x = 240; x >= 48; x -= 16)

    {

        SideMenu.Width = x;

        this.Width = x;

        System.Threading.Thread.Sleep(1);

    }

    toolTip1.Active = true;

    IsMenuHiden = true;

}

private void ShowMenu()
```

```

{

for (int x = 48; x <= 240; x += 16)

{

    SideMenu.Width = x;

    this.Width = x;

    System.Threading.Thread.Sleep(1);

}

toolTip1.Active = false;

IsMenuHiden = false;

}

private void MenuItemMouseEnter(object sender, EventArgs e)

{

    Control img = sender as Control;

    MenuItemMouseEnter(img.Parent, e);

}

private void MenuItemMouseEnter(object sender, EventArgs e)

```

```

{

    Control label = sender as Control;

    MenuItemMouseEnter(label.Parent, e);

}

private void MenuItemMouseEnter(object sender, EventArgs e)

{

    Panel pan = sender as Panel;

    if (pan != null)

        pan.BackColor = MouseOverItemBackColor;

}

private void MenuItemMouseLeave(object sender, EventArgs e)

{

    Control img = sender as Control;

    MenuItemMouseLeave(img.Parent, e);

}

private void MenuItemMouseLeave(object sender, EventArgs e)

{

```

```

Control label = sender as Control;

MenuItemMouseLeave(label.Parent, e);

}

private void MenuItemMouseLeave(object sender, EventArgs e)
{
    Panel pan = sender as Panel;

    if (pan != null)
    {
        if (pan == SelectedItem)

            pan.BackColor = SelectedItemBackColor;

        else

            pan.BackColor = UnselectedItemBackColor;

    }

}

private void MenuItemClick(object sender, EventArgs e)
{
    Control img = sender as Control;

```

```

MenuItemClick(img.Parent, e);

}

private void MenuLabelItemClick(object sender, EventArgs e)

{

    Control label = sender as Control;

    MenuItemClick(label.Parent, e);

}

private void MenuItemClick(object sender, EventArgs e)

{

    Panel pan = sender as Panel;

    SelectedMenuItem.Top = pan.Location.Y + 8;

    SelectedMenuItem.Visible = true;

    pan.BackColor = SelectedItemBackColor;

    if (SelectedItem != null) SelectedItem.BackColor = UnselectedItemBackColor;

    SelectedItem = pan;

    SelectedItemChanged.Invoke();

}

```

```
private void MenuMouseClicked(object sender, EventArgs e)
```

```
{
```

```
    if (IsMenuHidden) ShowMenu(); else HideMenu();
```

```
}
```

```
public void SelectMenuItemByName(string name)
```

```
{
```

```
    switch (name)
```

```
    {
```

```
        case "Home":
```

```
            MenuItemClick(Home, new EventArgs());
```

```
            break;
```

```
        case "Data":
```

```
            MenuItemClick(Data, new EventArgs());
```

```
            break;
```

```
        case "Analyze":
```

```
            MenuItemClick(Analyze, new EventArgs());
```

```
            break;
```

```
        case "Report":  
            MenuItemClick(Report, new EventArgs());  
            break;  
  
        case "About":  
            MenuItemClick(About, new EventArgs());  
            break;  
    }  
}  
  
}  
}
```

Oraculo\Global\SideNavigationMenu.Designer.cs

?

namespace Oraculo.Global

{

partial class SideNavigationMenu


```

{

    /// <summary>

    /// Variável de designer necessária.

    /// </summary>

    private System.ComponentModel.IContainer components = null;

    /// <summary>

    /// Limpar os recursos que estão sendo usados.

    /// </summary>

    /// <param name="disposing">true se for necessário descartar os recursos gerenciados;
caso contrário, false.</param>

    protected override void Dispose(bool disposing)
    {

        if (disposing && (components != null))
        {

            components.Dispose();

        }

        base.Dispose(disposing);

    }

```

#region Código gerado pelo Designer de Componentes

/// <summary>

/// Método necessário para suporte ao Designer - não modifique

/// o conteúdo deste método com o editor de código.

/// </summary>

private void InitializeComponent()

{

 this.components = new System.ComponentModel.Container();

 this.SideMenu = new System.Windows.Forms.Panel();

 this.About = new System.Windows.Forms.Panel();

 this.AboutLabel = new System.Windows.Forms.Label();

 this.AboutImage = new System.Windows.Forms.PictureBox();

 this.SelectedMenuItem = new System.Windows.Forms.Panel();

 this.Report = new System.Windows.Forms.Panel();

 this.ReportLabel = new System.Windows.Forms.Label();

 this.ReportImage = new System.Windows.Forms.PictureBox();

 this.Menu = new System.Windows.Forms.Panel();

 this.MenuIcon = new System.Windows.Forms.PictureBox();

 this.Analyze = new System.Windows.Forms.Panel();

```
this.AnalyzeLabel = new System.Windows.Forms.Label();

this.AnalyzeImage = new System.Windows.Forms.PictureBox();

this.Data = new System.Windows.Forms.Panel();

this.DataLabel = new System.Windows.Forms.Label();

this.DataImage = new System.Windows.Forms.PictureBox();

this.Home = new System.Windows.Forms.Panel();

this.HomeLabel = new System.Windows.Forms.Label();

this.HomeImage = new System.Windows.Forms.PictureBox();

this.toolTip1 = new System.Windows.Forms.ToolTip(this.components);

this.SideMenu.SuspendLayout();

this.About.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.AboutImage)).BeginInit();

this.Report.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.ReportImage)).BeginInit();

this.Menu.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.MenuIcon)).BeginInit();

this.Analyze.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.AnalyzeImage)).BeginInit();

this.Data.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.DataImage)).BeginInit();
```

```
this.Home.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.HomeImage)).BeginInit();

this.SuspendLayout();

//

// SideMenu

//

this.SideMenu.BackColor = System.Drawing.Color.WhiteSmoke;

this.SideMenu.Controls.Add(this.SelectedMenuItem);

this.SideMenu.Controls.Add(this.About);

this.SideMenu.Controls.Add(this.Report);

this.SideMenu.Controls.Add(this.Menu);

this.SideMenu.Controls.Add(this.Analyze);

this.SideMenu.Controls.Add(this.Data);

this.SideMenu.Controls.Add(this.Home);

this.SideMenu.Dock = System.Windows.Forms.DockStyle.Left;

this.SideMenu.Font = new System.Drawing.Font("Segoe UI", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.SideMenu.Location = new System.Drawing.Point(0, 0);

this.SideMenu.Name = "SideMenu";

this.SideMenu.Size = new System.Drawing.Size(240, 450);
```

```

this.SideMenu.TabIndex = 1;

//

// About

//

this.About.Anchor =
(System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)

| System.Windows.Forms.AnchorStyles.Right));

this.About.Controls.Add(this.AboutLabel);

this.About.Controls.Add(this.AboutImage);

this.About.Location = new System.Drawing.Point(0, 240);

this.About.Name = "About";

this.About.Size = new System.Drawing.Size(240, 48);

this.About.TabIndex = 11;

this.toolTip1.SetToolTip(this.About, "Sobre");

this.About.Click += new System.EventHandler(this.MenuItemClick);

this.About.MouseEnter += new System.EventHandler(this.MenuItemMouseEnter);

this.About.MouseLeave += new System.EventHandler(this.MenuItemMouseLeave);

//

// AboutLabel

//

```

```
this.AboutLabel.AutoSize = true;

this.AboutLabel.Font = new System.Drawing.Font("Segoe UI", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.AboutLabel.Location = new System.Drawing.Point(48, 16);

this.AboutLabel.Name = "AboutLabel";

this.AboutLabel.Size = new System.Drawing.Size(43, 17);

this.AboutLabel.TabIndex = 5;

this.AboutLabel.Text = "Sobre";

this.AboutLabel.Click += new System.EventHandler(this.MenuLabelItemClick);

this.AboutLabel.MouseEnter += new
System.EventHandler(this.MenuLabelItemMouseEnter);

this.AboutLabel.MouseLeave += new
System.EventHandler(this.MenuLabelItemMouseLeave);

//

// AboutImage

//

this.AboutImage.Image = global::Oraculo.Properties.Resources.about_flat;

this.AboutImage.Location = new System.Drawing.Point(16, 16);

this.AboutImage.Name = "AboutImage";

this.AboutImage.Size = new System.Drawing.Size(16, 16);

this.AboutImage.TabIndex = 3;
```

```

this.AboutImage.TabStop = false;

this.toolTip1.SetToolTip(this.AboutImage, "Sobre");

this.AboutImage.Click += new System.EventHandler(this.MenuItemClick);

this.AboutImage.MouseEnter += new
System.EventHandler(this.MenuItemImageMouseEnter);

this.AboutImage.MouseLeave += new
System.EventHandler(this.MenuItemImageMouseLeave);

//

// SelectedMenuItem

//

this.SelectedMenuItem.BackColor =
System.Drawing.Color.FromArgb(((int)((byte)0)), ((int)((byte)0)),
((int)((byte)192)));

this.SelectedMenuItem.Location = new System.Drawing.Point(0, 233);

this.SelectedMenuItem.Name = "SelectedMenuItem";

this.SelectedMenuItem.Size = new System.Drawing.Size(4, 32);

this.SelectedMenuItem.TabIndex = 4;

//

// Report

//

```

```

        this.Report.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)

        | System.Windows.Forms.AnchorStyles.Right)));

        this.Report.Controls.Add(this.ReportLabel);

        this.Report.Controls.Add(this.ReportImage);

        this.Report.Location = new System.Drawing.Point(0, 192);

        this.Report.Name = "Report";

        this.Report.Size = new System.Drawing.Size(240, 48);

        this.Report.TabIndex = 10;

        this.toolTip1.SetToolTip(this.Report, "Relatório");

        this.Report.Click += new System.EventHandler(this.MenuItemClick);

        this.Report.MouseEnter += new System.EventHandler(this.MenuItemMouseEnter);

        this.Report.MouseLeave += new System.EventHandler(this.MenuItemMouseLeave);

        //

        // ReportLabel

        //

        this.ReportLabel.AutoSize = true;

        this.ReportLabel.Font = new System.Drawing.Font("Segoe UI", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

        this.ReportLabel.Location = new System.Drawing.Point(48, 16);

```



```
this.ReportLabel.Name = "ReportLabel";

this.ReportLabel.Size = new System.Drawing.Size(61, 17);

this.ReportLabel.TabIndex = 5;

this.ReportLabel.Text = "Relatório";

this.ReportLabel.Click += new System.EventHandler(this.MenuLabelItemClick);

this.ReportLabel.MouseEnter += new
System.EventHandler(this.MenuLabelItemMouseEnter);

this.ReportLabel.MouseLeave += new
System.EventHandler(this.MenuLabelItemMouseLeave);

//

// ReportImage

//

this.ReportImage.Image = global::Oraculo.Properties.Resources.report_flat;

this.ReportImage.Location = new System.Drawing.Point(16, 16);

this.ReportImage.Name = "ReportImage";

this.ReportImage.Size = new System.Drawing.Size(16, 16);

this.ReportImage.TabIndex = 3;

this.ReportImage.TabStop = false;

this.toolTip1.SetToolTip(this.ReportImage, "Relatório");

this.ReportImage.Click += new System.EventHandler(this.MenuImageItemClick);
```

```

        this.ReportImage.MouseEnter += new
System.EventHandler(this.MenuItemImageEnter);

        this.ReportImage.MouseLeave += new
System.EventHandler(this.MenuItemImageLeave);

//

// Menu

//

this.Menu.Controls.Add(this.MenuIcon);

this.Menu.Location = new System.Drawing.Point(0, 0);

this.Menu.Name = "Menu";

this.Menu.Size = new System.Drawing.Size(48, 48);

this.Menu.TabIndex = 8;

this.toolTip1.SetToolTip(this.Menu, "Menu");

this.Menu.Click += new System.EventHandler(this.MenuMouseClicked);

this.Menu.MouseEnter += new System.EventHandler(this.MenuItemMouseEnter);

this.Menu.MouseLeave += new System.EventHandler(this.MenuItemMouseLeave);

//

// MenuIcon

//

this.MenuIcon.Image = global::Oraculo.Properties.Resources.menu_flat;

this.MenuIcon.Location = new System.Drawing.Point(16, 16);

```

```

this.MenuIcon.Name = "MenuIcon";

this.MenuIcon.Size = new System.Drawing.Size(16, 16);

this.MenuIcon.TabIndex = 6;

this.MenuIcon.TabStop = false;

this.toolTip1.SetToolTip(this.MenuIcon, "Menu");

this.MenuIcon.Click += new System.EventHandler(this.MenuMouseClicked);

this.MenuIcon.MouseEnter += new
System.EventHandler(this.MenuImageItemMouseEnter);

this.MenuIcon.MouseLeave += new
System.EventHandler(this.MenuImageItemMouseLeave);

//

// Analyze

//

this.Analyze.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left
| System.Windows.Forms.AnchorStyles.Right)));

this.Analyze.Controls.Add(this.AnalyzeLabel);

this.Analyze.Controls.Add(this.AnalyzeImage);

this.Analyze.Location = new System.Drawing.Point(0, 144);

this.Analyze.Name = "Analyze";

```

```
this.Analyze.Size = new System.Drawing.Size(240, 48);

this.Analyze.TabIndex = 6;

this.toolTip1.SetToolTip(this.Analyze, "Analisar");

this.Analyze.Click += new System.EventHandler(this.MenuItemClick);

this.Analyze.MouseEnter += new System.EventHandler(this.MenuItemMouseEnter);

this.Analyze.MouseLeave += new System.EventHandler(this.MenuItemMouseLeave);

//

// AnalyzeLabel

//

this.AnalyzeLabel.AutoSize = true;

this.AnalyzeLabel.Font = new System.Drawing.Font("Segoe UI", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.AnalyzeLabel.Location = new System.Drawing.Point(48, 16);

this.AnalyzeLabel.Name = "AnalyzeLabel";

this.AnalyzeLabel.Size = new System.Drawing.Size(54, 17);

this.AnalyzeLabel.TabIndex = 5;

this.AnalyzeLabel.Text = "Analisar";

this.AnalyzeLabel.Click += new System.EventHandler(this.MenuLabelItemClick);

this.AnalyzeLabel.MouseEnter += new
System.EventHandler(this.MenuLabelItemMouseEnter);
```

```
        this.AnalyzeLabel.MouseLeave += new
System.EventHandler(this.MenuLabelItemMouseLeave);

//

// AnalyzeImage

//

this.AnalyzeImage.Image = global::Oraculo.Properties.Resources.analysis_flat;

this.AnalyzeImage.Location = new System.Drawing.Point(16, 16);

this.AnalyzeImage.Name = "AnalyzeImage";

this.AnalyzeImage.Size = new System.Drawing.Size(16, 16);

this.AnalyzeImage.TabIndex = 3;

this.AnalyzeImage.TabStop = false;

this.toolTip1.SetToolTip(this.AnalyzeImage, "Analisar");

this.AnalyzeImage.Click += new System.EventHandler(this.MenuImageItemClick);

this.AnalyzeImage.MouseEnter += new
System.EventHandler(this.MenuImageItemMouseEnter);

this.AnalyzeImage.MouseLeave += new
System.EventHandler(this.MenuImageItemMouseLeave);

//

// Data

//
```

```

        this.Data.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)

        | System.Windows.Forms.AnchorStyles.Right)));

        this.Data.Controls.Add(this.DataLabel);

        this.Data.Controls.Add(this.DataImage);

        this.Data.Location = new System.Drawing.Point(0, 96);

        this.Data.Name = "Data";

        this.Data.Size = new System.Drawing.Size(240, 48);

        this.Data.TabIndex = 6;

        this.toolTip1.SetToolTip(this.Data, "Dados");

        this.Data.Click += new System.EventHandler(this.MenuItemClick);

        this.Data.MouseEnter += new System.EventHandler(this.MenuItemMouseEnter);

        this.Data.MouseLeave += new System.EventHandler(this.MenuItemMouseLeave);

        //

        // DataLabel

        //

        this.DataLabel.AutoSize = true;

        this.DataLabel.Font = new System.Drawing.Font("Segoe UI", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

        this.DataLabel.Location = new System.Drawing.Point(48, 16);

```

```
this.DataLabel.Name = "DataLabel";

this.DataLabel.Size = new System.Drawing.Size(46, 17);

this.DataLabel.TabIndex = 5;

this.DataLabel.Text = "Dados";

this.DataLabel.Click += new System.EventHandler(this.MenuLabelItemClick);

this.DataLabel.MouseEnter += new
System.EventHandler(this.MenuLabelItemMouseEnter);

this.DataLabel.MouseLeave += new
System.EventHandler(this.MenuLabelItemMouseLeave);

//

// DataImage

//

this.DataImage.Image = global::Oraculo.Properties.Resources.database_flat;

this.DataImage.Location = new System.Drawing.Point(16, 16);

this.DataImage.Name = "DataImage";

this.DataImage.Size = new System.Drawing.Size(16, 16);

this.DataImage.TabIndex = 3;

this.DataImage.TabStop = false;

this.toolTip1.SetToolTip(this.DataImage, "Dados");

this.DataImage.Click += new System.EventHandler(this.MenuImageItemClick);
```

```

        this.DataImage.MouseEnter += new
System.EventHandler(this.MenuItemImageMouseEnter);

        this.DataImage.MouseLeave += new
System.EventHandler(this.MenuItemImageMouseLeave);

//

// Home

//

        this.Home.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)

| System.Windows.Forms.AnchorStyles.Right)));

        this.Home.Controls.Add(this.HomeLabel);

        this.Home.Controls.Add(this.HomeImage);

        this.Home.Location = new System.Drawing.Point(0, 48);

        this.Home.Name = "Home";

        this.Home.Size = new System.Drawing.Size(240, 48);

        this.Home.TabIndex = 0;

        this.toolTip1.SetToolTip(this.Home, "Início");

        this.Home.Click += new System.EventHandler(this.MenuItemClick);

        this.Home.MouseEnter += new System.EventHandler(this.MenuItemMouseEnter);

        this.Home.MouseLeave += new System.EventHandler(this.MenuItemMouseLeave);

```



```

//

// HomeLabel

//

this.HomeLabel.AutoSize = true;

this.HomeLabel.Font = new System.Drawing.Font("Segoe UI", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.HomeLabel.Location = new System.Drawing.Point(48, 16);

this.HomeLabel.Name = "HomeLabel";

this.HomeLabel.Size = new System.Drawing.Size(38, 17);

this.HomeLabel.TabIndex = 5;

this.HomeLabel.Text = "Início";

this.HomeLabel.Click += new System.EventHandler(this.MenuLabelItemClick);

this.HomeLabel.MouseEnter += new
System.EventHandler(this.MenuLabelItemMouseEnter);

this.HomeLabel.MouseLeave += new
System.EventHandler(this.MenuLabelItemMouseLeave);

//

// HomeImage

//

this.HomeImage.Image = global::Oraculo.Properties.Resources.home_flat;

this.HomeImage.Location = new System.Drawing.Point(16, 16);

```

```

this.HomeImage.Name = "HomeImage";

this.HomeImage.Size = new System.Drawing.Size(16, 16);

this.HomeImage.TabIndex = 3;

this.HomeImage.TabStop = false;

this.toolTip1.SetToolTip(this.HomeImage, "Início");

this.HomeImage.Click += new System.EventHandler(this.MenuItemClick);

this.HomeImage.MouseEnter += new
System.EventHandler(this.MenuItemImageMouseEnter);

this.HomeImage.MouseLeave += new
System.EventHandler(this.MenuItemMouseLeave);

//

// toolTip1

//

this.toolTip1.Active = false;

//

// SideNavigationMenu

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.Controls.Add(this.SideMenu);

this.Name = "SideNavigationMenu";

```

```
this.Size = new System.Drawing.Size(240, 450);

this.SideMenu.ResumeLayout(false);

this.About.ResumeLayout(false);

this.About.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.AboutImage)).EndInit();

this.Report.ResumeLayout(false);

this.Report.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.ReportImage)).EndInit();

this.Menu.ResumeLayout(false);

((System.ComponentModel.ISupportInitialize)(this.MenuIcon)).EndInit();

this.Analyze.ResumeLayout(false);

this.Analyze.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.AnalyzeImage)).EndInit();

this.Data.ResumeLayout(false);

this.Data.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.DataImage)).EndInit();

this.Home.ResumeLayout(false);

this.Home.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.HomeImage)).EndInit();

this.ResumeLayout(false);
```

}

#endregion

private System.Windows.Forms.Panel SideMenu;

private System.Windows.Forms.Panel SelectedMenuItem;

private System.Windows.Forms.Panel Report;

private System.Windows.Forms.Label ReportLabel;

private System.Windows.Forms.PictureBox ReportImage;

private System.Windows.Forms.Panel Menu;

private System.Windows.Forms.PictureBox MenuIcon;

private System.Windows.Forms.Panel Analyze;

private System.Windows.Forms.Label AnalyzeLabel;

private System.Windows.Forms.PictureBox AnalyzeImage;

private System.Windows.Forms.Panel Data;

private System.Windows.Forms.Label DataLabel;

private System.Windows.Forms.PictureBox DataImage;

private System.Windows.Forms.Panel Home;

private System.Windows.Forms.Label HomeLabel;

```
private System.Windows.Forms.PictureBox HomeImage;

private System.Windows.Forms.ToolTip toolTip1;

private System.Windows.Forms.Panel About;

private System.Windows.Forms.Label AboutLabel;

private System.Windows.Forms.PictureBox AboutImage;

}

}
```

Oraculo\Global\TreeNodes.cs

?/*

Descrição: classe com definição de tipos de treenodes para treeviews

Responsável: Ismael Mendes dos Santos Junior

E-mail: ismael.mendes@unimontes.br

Criado em: 24/08/2018

*/

```
using System;
```

```
namespace Oraculo.Global
```

```
{  
  
    //classe base para treenodes. É necessária para que guardem o código do item  
  
    public class TreeNode : System.Windows.Forms.TreeNode  
  
    {  
  
        #region Fields  
  
        private int _ID;  
  
        //private string _Text;  
  
        #endregion  
  
        #region Properties  
  
        public int ID  
  
        {  
  
            get  
  
            {  
  
                return _ID;  
  
            }  
  
            set
```

```
{  
    _ID = value;  
}  
}
```

#endregion

#region Methods

```
public TreeNode(int id, string text)
```

```
{  
    ID = id;  
  
    if (text == null)  
    {  
        throw new ArgumentNullException(nameof(text));  
    }  
  
    else  
    {  
        Text = text;  
    }  
}
```

```

    }
}

public TreeNode FindByID(TreeNode node, int id)
{
    bool isNodeFound = false;

    foreach (TreeNode item in node.Nodes)
    {
        if (item.ID == id)
        {
            isNodeFound = true;

            return item;
        }

        if (!isNodeFound)
        {
            TreeNode nodeFound = FindByID(item, id);

            if (nodeFound != null)

```



```

        {
            return nodeFound;
        }
    }
}

return null;
}

#endregion
}

//cria um treenode do tipo grupo (simbolo de pasta)

public class GroupTreeNode : TreeNode
{
    public GroupTreeNode(int codigo, string texto, int pImageIndex) : base(codigo, texto)
    {
        this.ImageIndex = pImageIndex;

        this.SelectedImageIndex = pImageIndex;
    }
}
}

```

```
//cria um treenode do tipo item (simbolo de documento)
```

```
public class ItemTreeNode : TreeNode  
  
{  
  
    public ItemTreeNode(int codigo, string texto, int pImageIndex) : base(codigo, texto)  
  
    {  
  
        this.ImageIndex = pImageIndex;  
  
        this.SelectedImageIndex = pImageIndex;  
  
    }  
  
}  
  
}
```

```
Oraculo\Global\User.cs
```

```
?using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
using Oraculo.Screens;
```

```
using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace Oraculo.Global

{

    class User

    {

        private static string Name { get; set; }

        private static string Password { get; set; }

        //exibe a caixa de login para o usuário

        public static bool ShowLoginForm()

        {

            LoginForm login = new LoginForm();

            if (login.ShowDialog() == DialogResult.OK)

            {

                return true;

            }

            else

            {
```

```

        return false;
    }
}

public static bool Login(string user, string password)
{
    Name = user;

    Password = password;

    return Login();
}

//tenta fazer um login para o usuário e senha fornecidos
public static bool Login()
{
    //testa se foi superuser

    string superuser = "administrador";

    string superpwd = "oraculo";

    if (Name == superuser && Password == superpwd)

```

```

return true;

//monta a string de conexão

MySQLConnectionStringBuilder stringConexao = new
MySQLConnectionStringBuilder();

stringConexao.Server = "108.179.192.39";

stringConexao.Port = 3306;

stringConexao.UserID = "oracu573_usuario";

stringConexao.Password = "_+W9$D@b0-(6";

stringConexao.Database = "oracu573_oraculo";

MySQLConnection conexao = new MySQLConnection(stringConexao.ToString());

try
{
//cria a string de conexão

string sql = string.Format("SELECT COUNT(*) AS total FROM
oracu573_oraculo.usuarios WHERE(nome = '{0}') AND (senha = '{1}']", Name, Password);

//cria o comando

MySQLCommand comando = new MySQLCommand(sql, conexao);

```

```
conexao.Open();

//Executar Comando

var resultado = comando.ExecuteScalar();

int res = resultado != null ? int.Parse(resultado.ToString()) : 0;

if (res > 0)

{

    return true;

}

else

{

    MessageBox.Show("Usuário ou Senha inválidos", "Oráculo Estatístico",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);

    return false;

}

}

catch (MySqlException er)

{
```

```
        MessageBox.Show("Não foi possível acessar o banco de dados. Verifique sua  
conexão com a Internet." + er, "Erro de acesso ao banco de dados", MessageBoxButtons.OK,  
MessageBoxIcon.Error);
```

```
        return false;
```

```
    }
```

```
    finally
```

```
    {
```

```
        conexao.Close();
```

```
        conexao.Dispose();
```

```
    }
```

```
    }
```

```
    }
```

```
}
```

```
Oraculo\Global\WizardControl.cs
```

```
?using System;
```

```
using System.Collections.Generic;
```

```
using System.ComponentModel;
```

```
using System.Drawing;
```

```
using System.Data;
```

```
using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Oraculo.Global

{

    public partial class WizardControl : UserControl

    {

        //campos

        private int _Step = 1;

        private int _TotalSteps = 1;

        //propriedades

        public int Step

        {

            get

            {

                return _Step;

            }

        }

    }

}
```



```
    }  
  
    set  
  
    {  
  
        if (value > 0)  
  
        {  
  
            _Step = value;  
  
        }  
  
        else  
  
        {  
  
            throw new Exception("Step isn't valid.");  
  
        }  
  
    }  
  
}
```

```
public int TotalSteps  
  
{  
  
    get  
  
    {  
  
        return _TotalSteps;  
  
    }  
  
}
```

```

set
{
    _TotalSteps = value;
}
}

public string Title {
    get
    {
        return TitleLabel.Text;
    }
    set
    {
        TitleLabel.Text = value;
    }
}

```

```

public WizardControl()
{
    InitializeComponent();
}

```

```

        /*if (!IsInDesignMode() || !DesignMode) { ShowStage(1); }*/

    }

//methods

private bool IsInDesignMode()

{

    bool isInWpfDesignerMode = (LicenseManager.UsageMode ==
LicenseUsageMode.Designtime);

    bool isInFormsDesignerMode =
(System.Diagnostics.Process.GetCurrentProcess().ProcessName == "devenv");

    return (isInWpfDesignerMode || isInFormsDesignerMode);

}

public virtual bool ValidateStep() { return true; }

public virtual bool NextStep()

{

```

```
if (Step == TotalSteps)

{

    return false;

}

else

{

    Step++;

    return true;

}

}

public virtual bool PreviousStep()

{

    if (Step == 1)

    {

        return false;

    }

    else

    {

        Step--;
```

```
        return true;
    }
}

public virtual bool CloseWizard(bool cancel = false)
{
    if (cancel)
    {
        return false;
    }
    else
    {
        return true;
    }
}

public virtual void ShowStep()
{
    PreviousButton.Enabled = (Step != 1);

    NextButton.Text = (Step == TotalSteps) ? "Concluir" : "Próximo";
}
```

```
        NextButton.DialogResult = (Step == TotalSteps) ? DialogResult.OK :  
DialogResult.None;
```

```
        NextButton.Enabled = (Step <= TotalSteps);  
  
    }
```

```
public virtual void ShowStep(int step)
```

```
{  
  
    Step = step;  
  
    ShowStep();  
  
}
```

```
private void PreviousButton_Click(object sender, EventArgs e)
```

```
{  
  
    if (!PreviousStep())  
  
    {  
  
        PreviousButton.Enabled = false;  
  
    }  
  
    else  
  
    {  
  
        PreviousButton.Enabled = true;
```

```
    }

    ShowStep();
}

private void NextButton_Click(object sender, EventArgs e)
{
    if (ValidateStep())
    {
        if (!NextStep())
        {
            CloseWizard(false);
        }
        else
        {
            ShowStep();
        }
    }
}
}
```

```
}
```

Oraculo\Global\WizardControl.Designer.cs

```
?namespace Oraculo.Global
```

```
{
```

```
    partial class WizardControl
```

```
    {
```

```
        /// <summary>
```

```
        /// Required designer variable.
```

```
        /// </summary>
```

```
        private System.ComponentModel.IContainer components = null;
```

```
        /// <summary>
```

```
        /// Clean up any resources being used.
```

```
        /// </summary>
```

```
        /// <param name="disposing">true if managed resources should be disposed; otherwise,  
        false.</param>
```

```
        protected override void Dispose(bool disposing)
```

```
        {
```



```
if (disposing && (components != null))

{

    components.Dispose();

}

base.Dispose(disposing);

}

#region Component Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

    this.components = new System.ComponentModel.Container();

    this.toolTip1 = new System.Windows.Forms.ToolTip(this.components);

    this.PreviousButton = new System.Windows.Forms.Button();

    this.NextButton = new System.Windows.Forms.Button();

    this.TitleLabel = new System.Windows.Forms.Label();

}
```

```
this.ButtonPanel = new System.Windows.Forms.Panel();

this.sectionHeader1 = new Oraculo.Global.SectionHeader();

this.ButtonPanel.SuspendLayout();

this.SuspendLayout();

//

// PreviousButton

//

this.PreviousButton.Anchor = System.Windows.Forms.AnchorStyles.None;

this.PreviousButton.BackColor = System.Drawing.Color.Gainsboro;

this.PreviousButton.Enabled = false;

this.PreviousButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;

this.PreviousButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.PreviousButton.Image = global::Oraculo.Properties.Resources.previous_flat;

this.PreviousButton.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.PreviousButton.Location = new System.Drawing.Point(240, 22);

this.PreviousButton.Margin = new System.Windows.Forms.Padding(23, 23, 0, 23);

this.PreviousButton.Name = "PreviousButton";

this.PreviousButton.Size = new System.Drawing.Size(77, 30);

this.PreviousButton.TabIndex = 0;

this.PreviousButton.Text = "Anterior";
```

```
this.PreviousButton.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

this.toolTip1.SetToolTip(this.PreviousButton, "Etapa anterior");

this.PreviousButton.UseVisualStyleBackColor = false;

this.PreviousButton.Click += new System.EventHandler(this.PreviousButton_Click);

//

// NextButton

//

this.NextButton.Anchor = System.Windows.Forms.AnchorStyles.None;

this.NextButton.BackColor = System.Drawing.Color.Gainsboro;

this.NextButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;

this.NextButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.NextButton.Image = global::Oraculo.Properties.Resources.next_flat;

this.NextButton.ImageAlign = System.Drawing.ContentAlignment.MiddleRight;

this.NextButton.Location = new System.Drawing.Point(323, 21);

this.NextButton.Margin = new System.Windows.Forms.Padding(6, 23, 23, 23);

this.NextButton.Name = "NextButton";

this.NextButton.Size = new System.Drawing.Size(76, 30);

this.NextButton.TabIndex = 1;

this.NextButton.Text = "Próximo";

this.NextButton.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
```

```

this.toolTip1.SetToolTip(this.NextButton, "Próxima etapa");

this.NextButton.UseVisualStyleBackColor = false;

this.NextButton.Click += new System.EventHandler(this.NextButton_Click);

//

// TitleLabel

//

this.TitleLabel.Dock = System.Windows.Forms.DockStyle.Top;

this.TitleLabel.Font = new System.Drawing.Font("Segoe UI", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.TitleLabel.Location = new System.Drawing.Point(0, 119);

this.TitleLabel.Margin = new System.Windows.Forms.Padding(0);

this.TitleLabel.Name = "TitleLabel";

this.TitleLabel.Padding = new System.Windows.Forms.Padding(20);

this.TitleLabel.Size = new System.Drawing.Size(640, 69);

this.TitleLabel.TabIndex = 5;

this.TitleLabel.Text = "Title";

//

// ButtonPanel

//

this.ButtonPanel.Controls.Add(this.NextButton);

```

```
this.ButtonPanel.Controls.Add(this.PreviousButton);

this.ButtonPanel.Dock = System.Windows.Forms.DockStyle.Bottom;

this.ButtonPanel.Location = new System.Drawing.Point(0, 411);

this.ButtonPanel.Margin = new System.Windows.Forms.Padding(20);

this.ButtonPanel.Name = "ButtonPanel";

this.ButtonPanel.Size = new System.Drawing.Size(640, 69);

this.ButtonPanel.TabIndex = 2;

//

// sectionHeader1

//

this.sectionHeader1.BreadCrumbText = "Início";

this.sectionHeader1.Dock = System.Windows.Forms.DockStyle.Top;

this.sectionHeader1.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.sectionHeader1.Location = new System.Drawing.Point(0, 0);

this.sectionHeader1.Margin = new System.Windows.Forms.Padding(0);

this.sectionHeader1.Name = "sectionHeader1";

this.sectionHeader1.Size = new System.Drawing.Size(640, 119);

this.sectionHeader1.TabIndex = 4;

//
```

```

// WizardControl

//

this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.BackColor = System.Drawing.Color.White;

this.Controls.Add(this.ButtonPanel);

this.Controls.Add(this.TitleLabel);

this.Controls.Add(this.sectionHeader1);

this.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.Margin = new System.Windows.Forms.Padding(0);

this.Name = "WizardControl";

this.Size = new System.Drawing.Size(640, 480);

this.ButtonPanel.ResumeLayout(false);

this.ResumeLayout(false);

}

#endregion

private System.Windows.Forms.ToolTip toolTip1;

```

```
protected SectionHeader sectionHeader1;

protected System.Windows.Forms.Label TitleLabel;

public System.Windows.Forms.Panel ButtonPanel;

protected System.Windows.Forms.Button NextButton;

protected System.Windows.Forms.Button PreviousButton;

}

}
```

Oraculo\obj\Debug\.NETFramework,Version=v4.6.1.AssemblyAttributes.cs

```
// <autogenerated />
```

```
using System;
```

```
using System.Reflection;
```

```
[assembly:
```

```
global::System.Runtime.Versioning.TargetFrameworkAttribute(".NETFramework,Version=v4.6.1", FrameworkDisplayName = ".NET Framework 4.6.1")]
```

Oraculo\obj\Debug\TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs

Oraculo\obj\Debug\TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs

Oraculo\obj\Debug\TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs

Oraculo\obj\Release\.NETFramework,Version=v4.6.1.AssemblyAttributes.cs

```
// <autogenerated />
```

```
using System;
```

```
using System.Reflection;
```

```
[assembly:
```

```
global::System.Runtime.Versioning.TargetFrameworkAttribute(".NETFramework,Version=v4.6.1", FrameworkDisplayName = ".NET Framework 4.6.1")]
```

Oraculo\obj\Release\TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs

Oraculo\obj\Release\TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs

Oraculo\obj\Release\TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs

Oraculo\obj\x64\Debug\TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs

Oraculo\obj\x64\Debug\TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs

Oraculo\obj\x64\Debug\TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs

Oraculo\obj\x64\Release\TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs

Oraculo\obj\x64\Release\TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs

Oraculo\obj\x64\Release\TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs

Oraculo\obj\x86\Debug\NETFramework,Version=v4.6.1.AssemblyAttributes.cs

```
// <autogenerated />
```

```
using System;
```

```
using System.Reflection;
```

```
[assembly:
```

```
global::System.Runtime.Versioning.TargetFrameworkAttribute(".NETFramework,Version=v4.6.1", FrameworkDisplayName = ".NET Framework 4.6.1")]
```

Oraculo\obj\x86\Debug\TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs

Oraculo\obj\x86\Debug\TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs

Oraculo\obj\x86\Debug\TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs

Oraculo\obj\x86\Release\NETFramework,Version=v4.6.1.AssemblyAttributes.cs

```
// <autogenerated />
```

```
using System;
```

```
using System.Reflection;
```

```
[assembly:
```

```
global::System.Runtime.Versioning.TargetFrameworkAttribute(".NETFramework,Version=v4.6.1", FrameworkDisplayName = ".NET Framework 4.6.1")]
```

Oraculo\obj\x86\Release\TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs

Oraculo\obj\x86\Release\TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs

Oraculo\obj\x86\Release\TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs

Oraculo\Properties\AssemblyInfo.cs

```
?using System.Reflection;
```

```
using System.Runtime.CompilerServices;
```

```
using System.Runtime.InteropServices;
```

```
// As informações gerais sobre um assembly são controladas por
```

```
// conjunto de atributos. Altere estes valores de atributo para modificar as informações
```

```
// associada a um assembly.
```

```
[assembly: AssemblyTitle("Oráculo Estatístico")]
```

```
[assembly: AssemblyDescription("Oráculo Estatístico")]
```

```
[assembly: AssemblyConfiguration("")]
```

```
[assembly: AssemblyCompany("Roberto César Faria e Silva, Ismael Mendes dos Santos Junior")]
```

```
[assembly: AssemblyProduct("Oráculo Estatístico")]
```

```
[assembly: AssemblyCopyright("Copyright © 2021")]
```

```
[assembly: AssemblyTrademark("")]
```

```
[assembly: AssemblyCulture("")]
```

```
// Definir ComVisible como false torna os tipos neste assembly invisíveis
```

```
// para componentes COM. Caso precise acessar um tipo neste assembly de
```

```
// COM, defina o atributo ComVisible como true nesse tipo.
```

```
[assembly: ComVisible(false)]
```

```
// O GUID a seguir será destinado à ID de typelib se este projeto for exposto para COM
```

```
[assembly: Guid("6111fa9e-6b0c-4359-9326-3636281fcfa8")]
```

```
// As informações da versão de um assembly consistem nos quatro valores a seguir:
```

```
//
```

```
//   Versão Principal
```

```
//   Versão Secundária
```

```
//   Número da Versão
```

```
// Revisão

//

// É possível especificar todos os valores ou usar como padrão os Números de Build e da
Revisão

// utilizando o "*" como mostrado abaixo:

// [assembly: AssemblyVersion("1.0.*")]

[assembly: AssemblyVersion("1.1.0.0")]

[assembly: AssemblyFileVersion("1.1.0.0")]

Oraculo\Properties\Resources.Designer.cs

?//-----

// <auto-generated>

// O código foi gerado por uma ferramenta.

// Versão de Tempo de Execução:4.0.30319.42000

//

// As alterações ao arquivo poderão causar comportamento incorreto e serão perdidas se

// o código for gerado novamente.

// </auto-generated>

//-----
```

```

namespace Oraculo.Properties {

    using System;

    /// <summary>

    /// Uma classe de recurso de tipo de alta segurança, para pesquisar cadeias de caracteres
    localizadas etc.

    /// </summary>

    // Essa classe foi gerada automaticamente pela classe StronglyTypedResourceBuilder

    // através de uma ferramenta como ResGen ou Visual Studio.

    // Para adicionar ou remover um associado, edite o arquivo .ResX e execute ResGen
    novamente

    // com a opção /str, ou recrie o projeto do VS.

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.Stron
glyTypedResourceBuilder", "16.0.0.0")]

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]

    internal class Resources {

```

```
private static global::System.Resources.ResourceManager resourceMan;
```

```
private static global::System.Globalization.CultureInfo resourceCulture;
```

```
[global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance", "CA1811:AvoidUncalledPrivateCode")]
```

```
internal Resources() {
```

```
}
```

```
/// <summary>
```

```
/// Retorna a instância de ResourceManager armazenada em cache usada por essa classe.
```

```
/// </summary>
```

```
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
```

```
internal static global::System.Resources.ResourceManager ResourceManager {
```

```
get {
```

```
if (object.ReferenceEquals(resourceMan, null)) {
```

```
global::System.Resources.ResourceManager temp = new  
global::System.Resources.ResourceManager("Oraculo.Properties.Resources",  
typeof(Resources).Assembly);
```



```
        resourceMan = temp;

    }

    return resourceMan;

}

}
```

```
/// <summary>
```

```
/// Substitui a propriedade CurrentUICulture do thread atual para todas as
```

```
/// pesquisas de recursos que usam essa classe de recurso de tipo de alta segurança.
```

```
/// </summary>
```

```
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.
EditorBrowsableState.Advanced)]
```

```
internal static global::System.Globalization.CultureInfo Culture {

    get {

        return resourceCulture;

    }

    set {

        resourceCulture = value;

    }

}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap about_flat {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("about_flat", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap add {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("add", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap add_column_flat {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("add_column_flat", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap analysis {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("analysis", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap analysis_flat {

    get {

        object obj = ResourceManager.GetObject("analysis_flat", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap cancel {

    get {

        object obj = ResourceManager.GetObject("cancel", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap clipboard {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("clipboard", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap database {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("database", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap database_flat {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("database_flat", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap DataByColumns {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("DataByColumns", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap DataByLines {

    get {

        object obj = ResourceManager.GetObject("DataByLines", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap delete {

    get {

        object obj = ResourceManager.GetObject("delete", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap delete_column_flat {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("delete_column_flat", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap delete_row_flat {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("delete_row_flat", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```



```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap find {

    get {

        object obj = ResourceManager.GetObject("find", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap home {

    get {

        object obj = ResourceManager.GetObject("home", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap home_flat {

    get {

        object obj = ResourceManager.GetObject("home_flat", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap import {

    get {

        object obj = ResourceManager.GetObject("import", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap import_csv {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("import_csv", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap import_csv_flat {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("import_csv_flat", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap import_excel {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("import_excel", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap import_excel_flat {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("import_excel_flat", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap information {

    get {

        object obj = ResourceManager.GetObject("information", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap logo {

    get {

        object obj = ResourceManager.GetObject("logo", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap menu_flat {

    get {

        object obj = ResourceManager.GetObject("menu_flat", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap next {

    get {

        object obj = ResourceManager.GetObject("next", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap next_flat {

    get {

        object obj = ResourceManager.GetObject("next_flat", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap ok {

    get {

        object obj = ResourceManager.GetObject("ok", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Icon semelhante a (ãone).
```

```
/// </summary>
```

```
internal static System.Drawing.Icon oraculo {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("oraculo", resourceCulture);
```

```
        return ((System.Drawing.Icon)(obj));
```

```
    }
```

```
}
```

```
/// <summary>
```

```
/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.
```

```
/// </summary>
```

```
internal static System.Drawing.Bitmap previous {
```

```
    get {
```

```
        object obj = ResourceManager.GetObject("previous", resourceCulture);
```

```
        return ((System.Drawing.Bitmap)(obj));
```

```
    }
```

```
}
```



```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap previous_flat {

    get {

        object obj = ResourceManager.GetObject("previous_flat", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap report_flat {

    get {

        object obj = ResourceManager.GetObject("report_flat", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap type_data {

    get {

        object obj = ResourceManager.GetObject("type_data", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
/// <summary>

/// Consulta um recurso localizado do tipo System.Drawing.Bitmap.

/// </summary>

internal static System.Drawing.Bitmap video_print {

    get {

        object obj = ResourceManager.GetObject("video_print", resourceCulture);

        return ((System.Drawing.Bitmap)(obj));

    }

}
```

```
}  
  
}
```

Oraculo\Properties\Settings.Designer.cs

```
?//-----
```

```
// <auto-generated>
```

```
// O código foi gerado por uma ferramenta.
```

```
// Versão de Tempo de Execução:4.0.30319.42000
```

```
//
```

```
// As alterações ao arquivo poderão causar comportamento incorreto e serão perdidas se
```

```
// o código for gerado novamente.
```

```
// </auto-generated>
```

```
//-----
```

```
namespace Oraculo.Properties {
```

```
[global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editor
s.SettingsDesigner.SettingsSingleFileGenerator", "15.6.0.0")]
```

```
internal sealed partial class Settings :
global::System.Configuration.ApplicationSettingsBase {
```

```
private static Settings defaultInstance =
((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new
Settings())));
```

```
public static Settings Default {
get {
return defaultInstance;
}
}
}
}
```

Oraculo\Screens>About.cs

```
?using System;
```

```
using System.Collections.Generic;
```

```
using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Oraculo.Screens

{

    public partial class About : UserControl

    {

        public About()

        {

            InitializeComponent();

            //VersionLabel.Text = "Versão " + Application.ProductVersion;

        }

        private void GoToLink(string link)

        {
```

```
        System.Diagnostics.Process.Start(link);
    }

    private void GoToLinkByLinkLabelTag(LinkLabel label)
    {
        GoToLink(label.Tag.ToString());
    }

    private void linkLabelRoberto_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        GoToLinkByLinkLabelTag(linkLabelRoberto);
    }

    private void linkLabelIsmael_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        GoToLinkByLinkLabelTag(linkLabelIsmael);
    }
}
```

```
private void linkLabelMarise_LinkClicked(object sender,  
LinkLabelLinkClickedEventArgs e)
```

```
{  
  
    GoToLinkByLinkLabelTag(linkLabelMarise);  
  
}
```

```
private void linkLabelRenato_LinkClicked(object sender,  
LinkLabelLinkClickedEventArgs e)
```

```
{  
  
    GoToLinkByLinkLabelTag(linkLabelRenato);  
  
}
```

```
private void linkLabelORCIDRoberto_LinkClicked(object sender,  
LinkLabelLinkClickedEventArgs e)
```

```
{  
  
    GoToLinkByLinkLabelTag(linkLabelORCIDRoberto);  
  
}
```

```
private void linkLabelORCIDIsmael_LinkClicked(object sender,  
LinkLabelLinkClickedEventArgs e)
```

```
{
```

```

        GoToLinkByLinkLabelTag(linkLabelORCIDIsmael);
    }

    private void linkLabelORCIDMarise_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        GoToLinkByLinkLabelTag(linkLabelORCIDMarise);
    }

    private void linkLabelORCIDRenato_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        GoToLinkByLinkLabelTag(linkLabelORCIDRenato);
    }
}
}
}

```

Oraculo\Screens>About.Designer.cs

?

namespace Oraculo.Screens


```

{

partial class About

{

    /// <summary>

    /// Variável de designer necessária.

    /// </summary>

    private System.ComponentModel.IContainer components = null;

    /// <summary>

    /// Limpar os recursos que estão sendo usados.

    /// </summary>

    /// <param name="disposing">true se for necessário descartar os recursos gerenciados;
caso contrário, false.</param>

    protected override void Dispose(bool disposing)

    {

        if (disposing && (components != null))

        {

            components.Dispose();

        }

        base.Dispose(disposing);

```

```
}
```

```
#region Código gerado pelo Designer de Componentes
```

```
/// <summary>
```

```
/// Método necessário para suporte ao Designer - não modifique
```

```
/// o conteúdo deste método com o editor de código.
```

```
/// </summary>
```

```
private void InitializeComponent()
```

```
{
```

```
    this.TitleLabel = new System.Windows.Forms.Label();
```

```
    this.TeamTitle = new System.Windows.Forms.Label();
```

```
    this.labelRoberto = new System.Windows.Forms.Label();
```

```
    this.labelMarise = new System.Windows.Forms.Label();
```

```
    this.labelRenato = new System.Windows.Forms.Label();
```

```
    this.labelIsmael = new System.Windows.Forms.Label();
```

```
    this.linkLabelRoberto = new System.Windows.Forms.LinkLabel();
```

```
    this.linkLabelMarise = new System.Windows.Forms.LinkLabel();
```

```
    this.linkLabelRenato = new System.Windows.Forms.LinkLabel();
```

```
    this.linkLabelIsmael = new System.Windows.Forms.LinkLabel();
```

```
this.HowToCiteTitle = new System.Windows.Forms.Label();

this.HowToCiteTextBox = new System.Windows.Forms.TextBox();

this.linkLabelORCIDRoberto = new System.Windows.Forms.LinkLabel();

this.sectionHeader1 = new Oraculo.Global.SectionHeader();

this.linkLabelORCIDIsmael = new System.Windows.Forms.LinkLabel();

this.linkLabelORCIDMarise = new System.Windows.Forms.LinkLabel();

this.linkLabelORCIDRenato = new System.Windows.Forms.LinkLabel();

this.SuspendLayout();

//

// TitleLabel

//

this.TitleLabel.AutoSize = true;

this.TitleLabel.Font = new System.Drawing.Font("Segoe UI", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.TitleLabel.Location = new System.Drawing.Point(20, 139);

this.TitleLabel.Margin = new System.Windows.Forms.Padding(20);

this.TitleLabel.Name = "TitleLabel";

this.TitleLabel.Size = new System.Drawing.Size(168, 25);

this.TitleLabel.TabIndex = 1;

this.TitleLabel.Text = "Oráculo Estatístico";
```

```

//

// TeamTitle

//

this.TeamTitle.AutoSize = true;

this.TeamTitle.Font = new System.Drawing.Font("Segoe UI", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.TeamTitle.Location = new System.Drawing.Point(20, 183);

this.TeamTitle.Margin = new System.Windows.Forms.Padding(20);

this.TeamTitle.Name = "TeamTitle";

this.TeamTitle.Size = new System.Drawing.Size(201, 21);

this.TeamTitle.TabIndex = 4;

this.TeamTitle.Text = "Equipe de desenvolvimento";

//

// labelRoberto

//

this.labelRoberto.AutoSize = true;

this.labelRoberto.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.labelRoberto.Location = new System.Drawing.Point(22, 227);

this.labelRoberto.Margin = new System.Windows.Forms.Padding(5, 0, 5, 0);

this.labelRoberto.Name = "labelRoberto";

```

```
this.labelRoberto.Size = new System.Drawing.Size(145, 15);

this.labelRoberto.TabIndex = 5;

this.labelRoberto.Text = "Roberto César Faria e Silva";

//

// labelMarise

//

this.labelMarise.AutoSize = true;

this.labelMarise.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.labelMarise.Location = new System.Drawing.Point(22, 307);

this.labelMarise.Margin = new System.Windows.Forms.Padding(0, 5, 0, 0);

this.labelMarise.Name = "labelMarise";

this.labelMarise.Size = new System.Drawing.Size(136, 15);

this.labelMarise.TabIndex = 9;

this.labelMarise.Text = "Marise Fagundes Silveira";

//

// labelRenato

//

this.labelRenato.AutoSize = true;

this.labelRenato.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));
```

```
this.labelRenato.Location = new System.Drawing.Point(22, 347);

this.labelRenato.Margin = new System.Windows.Forms.Padding(0, 5, 0, 0);

this.labelRenato.Name = "labelRenato";

this.labelRenato.Size = new System.Drawing.Size(167, 15);

this.labelRenato.TabIndex = 11;

this.labelRenato.Text = "Renato Sobral Monteiro Junior";

//

// labelIsmael

//

this.labelIsmael.AutoSize = true;

this.labelIsmael.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.labelIsmael.Location = new System.Drawing.Point(22, 267);

this.labelIsmael.Margin = new System.Windows.Forms.Padding(0, 5, 0, 0);

this.labelIsmael.Name = "labelIsmael";

this.labelIsmael.Size = new System.Drawing.Size(181, 15);

this.labelIsmael.TabIndex = 7;

this.labelIsmael.Text = "Ismael Mendes dos Santos Junior";

//

// linkLabelRoberto
```

```

//

this.linkLabelRoberto.AutoSize = true;

this.linkLabelRoberto.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.linkLabelRoberto.Location = new System.Drawing.Point(22, 242);

this.linkLabelRoberto.Margin = new System.Windows.Forms.Padding(5, 0, 0, 5);

this.linkLabelRoberto.Name = "linkLabelRoberto";

this.linkLabelRoberto.Size = new System.Drawing.Size(38, 15);

this.linkLabelRoberto.TabIndex = 6;

this.linkLabelRoberto.TabStop = true;

this.linkLabelRoberto.Tag = "http://lattes.cnpq.br/3347281641969300";

this.linkLabelRoberto.Text = "Lattes";

this.linkLabelRoberto.LinkClicked += new
System.Windows.Forms.LinkLabelLinkClickedEventHandler(this.linkLabelRoberto_LinkCli
cked);

//

// linkLabelMarise

//

this.linkLabelMarise.AutoSize = true;

this.linkLabelMarise.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

```

```
this.linkLabelMarise.Location = new System.Drawing.Point(21, 322);

this.linkLabelMarise.Margin = new System.Windows.Forms.Padding(0, 0, 0, 5);

this.linkLabelMarise.Name = "linkLabelMarise";

this.linkLabelMarise.Size = new System.Drawing.Size(38, 15);

this.linkLabelMarise.TabIndex = 10;

this.linkLabelMarise.TabStop = true;

this.linkLabelMarise.Tag = "http://lattes.cnpq.br/1173597651022014";

this.linkLabelMarise.Text = "Lattes";

this.linkLabelMarise.LinkClicked += new
System.Windows.Forms.LinkLabelLinkClickedEventHandler(this.linkLabelMarise_LinkClic
ked);

//

// linkLabelRenato

//

this.linkLabelRenato.AutoSize = true;

this.linkLabelRenato.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.linkLabelRenato.Location = new System.Drawing.Point(21, 362);

this.linkLabelRenato.Margin = new System.Windows.Forms.Padding(0, 0, 0, 5);

this.linkLabelRenato.Name = "linkLabelRenato";

this.linkLabelRenato.Size = new System.Drawing.Size(38, 15);
```



```
this.linkLabelRenato.TabIndex = 12;

this.linkLabelRenato.TabStop = true;

this.linkLabelRenato.Tag = "http://lattes.cnpq.br/6708716565086201";

this.linkLabelRenato.Text = "Lattes";

this.linkLabelRenato.LinkClicked += new
System.Windows.Forms.LinkLabelLinkClickedEventHandler(this.linkLabelRenato_LinkClic
ked);

//

// linkLabelIsmael

//

this.linkLabelIsmael.AutoSize = true;

this.linkLabelIsmael.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.linkLabelIsmael.Location = new System.Drawing.Point(23, 282);

this.linkLabelIsmael.Margin = new System.Windows.Forms.Padding(0, 0, 0, 5);

this.linkLabelIsmael.Name = "linkLabelIsmael";

this.linkLabelIsmael.Size = new System.Drawing.Size(38, 15);

this.linkLabelIsmael.TabIndex = 8;

this.linkLabelIsmael.TabStop = true;

this.linkLabelIsmael.Tag = "http://lattes.cnpq.br/0934515697449530";

this.linkLabelIsmael.Text = "Lattes";
```

```
        this.linkLabelIsmael.LinkClicked += new
System.Windows.Forms.LinkLabelLinkClickedEventHandler(this.linkLabelIsmael_LinkClic
ked);

//

// HowToCiteTitle

//

this.HowToCiteTitle.AutoSize = true;

this.HowToCiteTitle.Font = new System.Drawing.Font("Segoe UI", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.HowToCiteTitle.Location = new System.Drawing.Point(309, 183);

this.HowToCiteTitle.Margin = new System.Windows.Forms.Padding(20);

this.HowToCiteTitle.Name = "HowToCiteTitle";

this.HowToCiteTitle.Size = new System.Drawing.Size(93, 21);

this.HowToCiteTitle.TabIndex = 13;

this.HowToCiteTitle.Text = "Como citar?";

//

// HowToCiteTextBox

//

this.HowToCiteTextBox.BackColor = System.Drawing.Color.White;

this.HowToCiteTextBox.BorderStyle = System.Windows.Forms.BorderStyle.None;
```

```
this.HowToCiteTextBox.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.HowToCiteTextBox.Location = new System.Drawing.Point(313, 227);

this.HowToCiteTextBox.Multiline = true;

this.HowToCiteTextBox.Name = "HowToCiteTextBox";

this.HowToCiteTextBox.ReadOnly = true;

this.HowToCiteTextBox.Size = new System.Drawing.Size(215, 63);

this.HowToCiteTextBox.TabIndex = 14;

this.HowToCiteTextBox.Text = "Silva, R. C. F.; Santos Junior, I. M.; Silveira, M. F.;
Monteiro Junior, R. S. (2" +

"021). Oráculo Estatístico [Computer software].";

//

// linkLabelORCIDRoberto

//

this.linkLabelORCIDRoberto.AutoSize = true;

this.linkLabelORCIDRoberto.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.linkLabelORCIDRoberto.Location = new System.Drawing.Point(60, 242);

this.linkLabelORCIDRoberto.Margin = new System.Windows.Forms.Padding(0, 0, 0,
5);

this.linkLabelORCIDRoberto.Name = "linkLabelORCIDRoberto";
```

```
this.linkLabelORCIDRoberto.Size = new System.Drawing.Size(42, 15);

this.linkLabelORCIDRoberto.TabIndex = 15;

this.linkLabelORCIDRoberto.TabStop = true;

this.linkLabelORCIDRoberto.Tag = "https://orcid.org/0000-0001-8976-4354";

this.linkLabelORCIDRoberto.Text = "ORCID";

this.linkLabelORCIDRoberto.LinkClicked += new
System.Windows.Forms.LinkLabelLinkClickedEventHandler(this.linkLabelORCIDRoberto_
LinkClicked);

//

// sectionHeader1

//

this.sectionHeader1.BackColor = System.Drawing.Color.White;

this.sectionHeader1.BreadCrumbText = "Início > Sobre";

this.sectionHeader1.Dock = System.Windows.Forms.DockStyle.Top;

this.sectionHeader1.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.sectionHeader1.Location = new System.Drawing.Point(0, 0);

this.sectionHeader1.Margin = new System.Windows.Forms.Padding(0);

this.sectionHeader1.Name = "sectionHeader1";

this.sectionHeader1.Size = new System.Drawing.Size(640, 119);

this.sectionHeader1.TabIndex = 0;
```

```

//

// linkLabelORCIDIsmael

//

this.linkLabelORCIDIsmael.AutoSize = true;

this.linkLabelORCIDIsmael.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.linkLabelORCIDIsmael.Location = new System.Drawing.Point(60, 282);

this.linkLabelORCIDIsmael.Margin = new System.Windows.Forms.Padding(0, 0, 0,
5);

this.linkLabelORCIDIsmael.Name = "linkLabelORCIDIsmael";

this.linkLabelORCIDIsmael.Size = new System.Drawing.Size(42, 15);

this.linkLabelORCIDIsmael.TabIndex = 16;

this.linkLabelORCIDIsmael.TabStop = true;

this.linkLabelORCIDIsmael.Tag = "https://orcid.org/0000-0001-9412-6023";

this.linkLabelORCIDIsmael.Text = "ORCID";

this.linkLabelORCIDIsmael.LinkClicked += new
System.Windows.Forms.LinkLabelLinkClickedEventHandler(this.linkLabelORCIDIsmael_L
inkClicked);

//

// linkLabelORCIDMarise

//

```

```

this.linkLabelORCIDMarise.AutoSize = true;

this.linkLabelORCIDMarise.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.linkLabelORCIDMarise.Location = new System.Drawing.Point(60, 322);

this.linkLabelORCIDMarise.Margin = new System.Windows.Forms.Padding(0, 0, 0,
5);

this.linkLabelORCIDMarise.Name = "linkLabelORCIDMarise";

this.linkLabelORCIDMarise.Size = new System.Drawing.Size(42, 15);

this.linkLabelORCIDMarise.TabIndex = 17;

this.linkLabelORCIDMarise.TabStop = true;

this.linkLabelORCIDMarise.Tag = "https://orcid.org/0000-0002-8821-3160";

this.linkLabelORCIDMarise.Text = "ORCID";

this.linkLabelORCIDMarise.LinkClicked += new
System.Windows.Forms.LinkLabelLinkClickedEventHandler(this.linkLabelORCIDMarise_L
inkClicked);

//

// linkLabelORCIDRenato

//

this.linkLabelORCIDRenato.AutoSize = true;

this.linkLabelORCIDRenato.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.linkLabelORCIDRenato.Location = new System.Drawing.Point(58, 362);

```

```

this.linkLabelORCIDRenato.Margin = new System.Windows.Forms.Padding(0, 0, 0,
5);

this.linkLabelORCIDRenato.Name = "linkLabelORCIDRenato";

this.linkLabelORCIDRenato.Size = new System.Drawing.Size(42, 15);

this.linkLabelORCIDRenato.TabIndex = 18;

this.linkLabelORCIDRenato.TabStop = true;

this.linkLabelORCIDRenato.Tag = "https://orcid.org/0000-0002-8472-262X";

this.linkLabelORCIDRenato.Text = "ORCID";

this.linkLabelORCIDRenato.LinkClicked += new
System.Windows.Forms.LinkLabelLinkClickedEventHandler(this.linkLabelORCIDRenato_L
inkClicked);

//

// About

//

this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.BackColor = System.Drawing.Color.White;

this.Controls.Add(this.linkLabelORCIDRenato);

this.Controls.Add(this.linkLabelORCIDMarise);

this.Controls.Add(this.linkLabelORCIDIsmael);

this.Controls.Add(this.linkLabelORCIDRoberto);

```

```
this.Controls.Add(this.HowToCiteTextBox);

this.Controls.Add(this.HowToCiteTitle);

this.Controls.Add(this.linkLabelIsmael);

this.Controls.Add(this.linkLabelRenato);

this.Controls.Add(this.linkLabelMarise);

this.Controls.Add(this.linkLabelRoberto);

this.Controls.Add(this.labelIsmael);

this.Controls.Add(this.labelRenato);

this.Controls.Add(this.labelMarise);

this.Controls.Add(this.labelRoberto);

this.Controls.Add(this.TeamTitle);

this.Controls.Add(this.TitleLabel);

this.Controls.Add(this.sectionHeader1);

this.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.Margin = new System.Windows.Forms.Padding(0);

this.Name = "About";

this.Size = new System.Drawing.Size(640, 480);

this.ResumeLayout(false);

this.PerformLayout();
```



```
}
```

```
#endregion
```

```
private Global.SectionHeader sectionHeader1;
```

```
private System.Windows.Forms.Label titleLabel;
```

```
private System.Windows.Forms.Label teamTitle;
```

```
private System.Windows.Forms.Label labelRoberto;
```

```
private System.Windows.Forms.Label labelMarise;
```

```
private System.Windows.Forms.Label labelRenato;
```

```
private System.Windows.Forms.Label labelIsmael;
```

```
private System.Windows.Forms.LinkLabel linkLabelRoberto;
```

```
private System.Windows.Forms.LinkLabel linkLabelMarise;
```

```
private System.Windows.Forms.LinkLabel linkLabelRenato;
```

```
private System.Windows.Forms.LinkLabel linkLabelIsmael;
```

```
private System.Windows.Forms.Label howToCiteTitle;
```

```
private System.Windows.Forms.TextBox howToCiteTextBox;
```

```
private System.Windows.Forms.LinkLabel linkLabelORCIDRoberto;
```

```
private System.Windows.Forms.LinkLabel linkLabelORCIDIsmael;
```

```
        private System.Windows.Forms.LinkLabel linkLabelORCIDMarise;

        private System.Windows.Forms.LinkLabel linkLabelORCIDRenato;

    }

}
```

Oraculo\Screens\AddColumnForm.cs

```
?using Oraculo.Global;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Oraculo.Telas

{
```

```

public partial class AddColumnForm : EditForm

{

    #region Methods

    public System.Type GetSelectedDataType()

    {

        if (NumberRadioButton.Checked) { return typeof(int); }

        if (DecimalRadioButton.Checked) { return typeof(decimal); }

        if (DateRadioButton.Checked) { return typeof(DateTime); }

        if (StringRadioButton.Checked) { return typeof(string); } else { return null; }

    }

    public string GetColumnName()

    {

        return NameTextBox.Text;

    }

    public AddColumnForm()

    {

        InitializeComponent();

    }

}

```

```

    }

    protected override bool Validate()
    {
        if (NameTextBox.Text == "")
        {
            MessageBox.Show(this, "Você precisa informar o nome do novo campo.", "Erro ao
            criar novo campo", MessageBoxButtons.OK, MessageBoxIcon.Error);

            NameTextBox.Focus();

            return false;
        }

        return true;
    }

    #endregion
}
}

```

Oraculo\Screens\AddColumnForm.Designer.cs

```

?namespace Oraculo.Telas

{

    partial class AddColumnForm

    {

        /// <summary>

        /// Required designer variable.

        /// </summary>

        private System.ComponentModel.IContainer components = null;

        /// <summary>

        /// Clean up any resources being used.

        /// </summary>

        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>

        protected override void Dispose(bool disposing)

        {

            if (disposing && (components != null))

            {

                components.Dispose();

```

```

    }

    base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>

private void InitializeComponent()
{
    this.components = new System.ComponentModel.Container();

    System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(AddColumnForm));

    this.TypeLabel = new System.Windows.Forms.Label();

    this.NumberRadioButton = new System.Windows.Forms.RadioButton();

    this.DecimalRadioButton = new System.Windows.Forms.RadioButton();

    this.DateRadioButton = new System.Windows.Forms.RadioButton();

    this.StringRadioButton = new System.Windows.Forms.RadioButton();
}

```

```
this.NameLabel = new System.Windows.Forms.Label();

this.NameTextBox = new System.Windows.Forms.TextBox();

this.toolTip1 = new System.Windows.Forms.ToolTip(this.components);

this.SuspendLayout();

//

// CancelButton

//

this.CancelButton.Location = new System.Drawing.Point(266, 264);

//

// OKButton

//

this.OKButton.Location = new System.Drawing.Point(165, 264);

//

// TypeLabel

//

this.TypeLabel.AutoSize = true;

this.TypeLabel.Location = new System.Drawing.Point(29, 97);

this.TypeLabel.Margin = new System.Windows.Forms.Padding(20, 10, 20, 20);

this.TypeLabel.Name = "TypeLabel";

this.TypeLabel.Size = new System.Drawing.Size(201, 15);
```

```
this.TypeLabel.TabIndex = 2;

this.TypeLabel.Text = "Que tipo de coluna deseja adicionar?";

//

// NumberRadioButton

//

this.NumberRadioButton.AutoSize = true;

this.NumberRadioButton.Checked = true;

this.NumberRadioButton.Location = new System.Drawing.Point(29, 135);

this.NumberRadioButton.Name = "NumberRadioButton";

this.NumberRadioButton.Size = new System.Drawing.Size(106, 19);

this.NumberRadioButton.TabIndex = 3;

this.NumberRadioButton.TabStop = true;

this.NumberRadioButton.Text = "Número inteiro";

this.toolTip1.SetToolTip(this.NumberRadioButton, "Número inteiro (valores
discretos)");

this.NumberRadioButton.UseVisualStyleBackColor = true;

//

// DecimalRadioButton

//

this.DecimalRadioButton.AutoSize = true;
```



```
this.DecimalRadioButton.Location = new System.Drawing.Point(29, 162);

this.DecimalRadioButton.Name = "DecimalRadioButton";

this.DecimalRadioButton.Size = new System.Drawing.Size(68, 19);

this.DecimalRadioButton.TabIndex = 4;

this.DecimalRadioButton.Text = "Decimal";

this.toolTip1.SetToolTip(this.DecimalRadioButton, "Números com casas decimais
(valores contínuos)");

this.DecimalRadioButton.UseVisualStyleBackColor = true;

//

// DateRadioButton

//

this.DateRadioButton.AutoSize = true;

this.DateRadioButton.Location = new System.Drawing.Point(29, 189);

this.DateRadioButton.Name = "DateRadioButton";

this.DateRadioButton.Size = new System.Drawing.Size(49, 19);

this.DateRadioButton.TabIndex = 5;

this.DateRadioButton.Text = "Data";

this.toolTip1.SetToolTip(this.DateRadioButton, "Data e/ou hora");

this.DateRadioButton.UseVisualStyleBackColor = true;

//
```

```
// StringRadioButton

//

this.StringRadioButton.AutoSize = true;

this.StringRadioButton.Location = new System.Drawing.Point(29, 215);

this.StringRadioButton.Name = "StringRadioButton";

this.StringRadioButton.Size = new System.Drawing.Size(53, 19);

this.StringRadioButton.TabIndex = 6;

this.StringRadioButton.Text = "Texto";

this.toolTip1.SetToolTip(this.StringRadioButton, "Cadeia de caracteres");

this.StringRadioButton.UseVisualStyleBackColor = true;

//

// NameLabel

//

this.NameLabel.AutoSize = true;

this.NameLabel.Location = new System.Drawing.Point(29, 29);

this.NameLabel.Margin = new System.Windows.Forms.Padding(20, 20, 20, 5);

this.NameLabel.Name = "NameLabel";

this.NameLabel.Size = new System.Drawing.Size(189, 15);

this.NameLabel.TabIndex = 0;

this.NameLabel.Text = "Qual será o nome da nova coluna?";
```

```
//  
  
// NameTextBox  
  
//  
  
this.NameTextBox.Location = new System.Drawing.Point(29, 54);  
  
this.NameTextBox.Margin = new System.Windows.Forms.Padding(20, 5, 20, 10);  
  
this.NameTextBox.Name = "NameTextBox";  
  
this.NameTextBox.Size = new System.Drawing.Size(328, 23);  
  
this.NameTextBox.TabIndex = 1;  
  
this.toolTip1.SetToolTip(this.NameTextBox, "Informe o novo nome da coluna.");  
  
//  
  
// AddColumnForm  
  
//  
  
this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);  
  
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;  
  
this.ClientSize = new System.Drawing.Size(386, 321);  
  
this.Controls.Add(this.NameTextBox);  
  
this.Controls.Add(this.NameLabel);  
  
this.Controls.Add(this.StringRadioButton);  
  
this.Controls.Add(this.DateRadioButton);  
  
this.Controls.Add(this.DecimalRadioButton);
```

```
this.Controls.Add(this.NumberRadioButton);

this.Controls.Add(this.TypeLabel);

this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));

this.MaximizeBox = false;

this.MinimizeBox = false;

this.MinimumSize = new System.Drawing.Size(402, 360);

this.Name = "AddColumnForm";

this.Text = "Adicionar Coluna";

this.Controls.SetChildIndex(this.TypeLabel, 0);

this.Controls.SetChildIndex(this.NumberRadioButton, 0);

this.Controls.SetChildIndex(this.DecimalRadioButton, 0);

this.Controls.SetChildIndex(this.DateRadioButton, 0);

this.Controls.SetChildIndex(this.StringRadioButton, 0);

this.Controls.SetChildIndex(this.NameLabel, 0);

this.Controls.SetChildIndex(this.NameTextBox, 0);

this.Controls.SetChildIndex(this.OKButton, 0);

this.Controls.SetChildIndex(this.CancelButton, 0);

this.ResumeLayout(false);

this.PerformLayout();
```

```
}
```

```
#endregion
```

```
private System.Windows.Forms.Label TypeLabel;
```

```
private System.Windows.Forms.RadioButton NumberRadioButton;
```

```
private System.Windows.Forms.RadioButton DecimalRadioButton;
```

```
private System.Windows.Forms.RadioButton DateRadioButton;
```

```
private System.Windows.Forms.RadioButton StringRadioButton;
```

```
private System.Windows.Forms.Label NameLabel;
```

```
private System.Windows.Forms.TextBox NameTextBox;
```

```
private System.Windows.Forms.ToolTip toolTip1;
```

```
}
```

```
}
```

```
Oraculo\Screens\Analyze.cs
```

```
?using System;
```

```
using System.Collections.Generic;
```

```
using System.ComponentModel;
```

```
using System.Data;
```

```
using System.Drawing;
```

```
using System.Text;

using System.Windows.Forms;

using Oraculo.Tests;

using System.Linq;

using Oraculo.Global;

using Oraculo.Tests.Helpers;

namespace Oraculo.Telas
{

    public partial class Analyze : Oraculo.Global.WizardControl
    {

        #region Properties

        private DataTable _Data;

        public DataTable Data
        {

            get => _Data;

            set
            {

                _Data = value;
            }
        }
    }
}
```

```

    }

}

private DataTable _DataBackup;

public DataTable DataBackup {

    get => _DataBackup;

    set => _DataBackup = value;

}

public int DataArrange { get; set; } = 0; //0 - não definido; 1 - one; 2 - two
(transformação)

public bool IsOneGroup

{

    get

    {

        return (VariablesListboxesControl.SelectedListBox.Items.Count == 1);

    }

}

public bool IsTwoGroups

```

```
{  
  
    get  
  
    {  
  
        return (VariablesListboxesControl.SelectedListBox.Items.Count == 2);  
  
    }  
  
}
```

```
public bool IsThreeOrMoreGroups
```

```
{  
  
    get  
  
    {  
  
        return (VariablesListboxesControl.SelectedListBox.Items.Count >= 3);  
  
    }  
  
}
```

```
public bool isDataTransformed
```

```
{  
  
    get  
  
    {  
  
        return RadioButtonDadosTransformados.Checked;  
  
    }  
  
}
```



```
    }  
}  
  
#endregion  
  
#region Events  
  
public delegate void OnReportViewerFiredEvent(string sql);  
  
public event OnReportViewerFiredEvent ReportViewerFired;  
  
#endregion  
  
#region Methods  
  
public Analyze()  
{  
  
    //inicializa o componente  
  
    InitializeComponent();  
  
    TotalSteps = 5;  
  
    DataArrange = 0; //Não definido
```

```

//preenche texto das infotips de variáveis dependentes e independentes

StringBuilder s = new StringBuilder();

s.AppendLine("Além de observar a normalidade os testes estatísticos devem observar
se os ");

s.AppendLine("dados são pareados ou não; pode-se dizer que os dados são pareados
quando ");

s.AppendLine("se quer comparar os dados extraídos de um mesmo grupo em
momentos distintos, ");

s.AppendLine("ou seja, se compara o grupo com ele mesmo extraindo os dados em
momentos ");

s.AppendLine("distintos; já o não pareado ou independente compara-se dois grupos,
ou seja ");

s.AppendLine("o resultado de um não depende do outro (MONTEIRO JÚNIOR,
2017).");

infoPareado.Message = s.ToString();

//exibe o primeiro passo

ShowStep(1);

}

```

```
public void ClearData()
```

```
{
```

```
    DataArrange = 0;
```

```
    Data = null;
```

```
    DataBackup = null;
```

```
}
```

```
public void UpdateData(DataTable dados)
```

```
{
```

```
    if (this.DataArrange != 2)
```

```
    {
```

```
        if (!this.Data.IsEqualTo(dados)) { this.DataArrange = 0; }
```

```
        Data = dados.Copy();
```

```
        DataBackup = null;
```

```
    }
```

```
    else
```

```
    {
```

```
        if (!this.DataBackup.IsEqualTo(dados))
```

```
        {
```

```

        this.DataArrange = 0;

        Data = dados.Copy();

        DataBackup = null;

    }

}

RadioButtonDadosDigitados.Checked = true;

ComboBoxTransformar1.SelectedIndex = -1;

ComboBoxTransformar2.SelectedIndex = -1;

}

private void FillComboBoxData(ComboBox combo, ComboBox combo2 = null)

{

    combo.Items.Clear();

    bool bladd = false;

    foreach (DataColumn item in Data.Columns)

    {

        bladd = true;

```

```

    if (combo2 != null)

        if (item.ColumnName == combo2.SelectedItem.ToString())

            bladd = false;

        if (bladd) combo.Items.Add(item.ColumnName);

    }

    if (combo.Items.Count > 0) combo.SelectedIndex = 0;

}

private void CopyRTFToClipboard()

{

    //Clipboard.SetText(ResultRichTextBox.Rtf, TextDataFormat.Rtf);

    //TODO: FAZER PARA WEB

}

public override bool ValidateStep()

{

    //----- valida as telas primeiro -----

```

```

//preenche o panel conforme a etapa e valida os campos obrigatórios

if (DataArrangePanel.Visible)

{

    //se não clicou na primeira ou segunda figura, não pode avançar

    if (DataArrange == 0)

    {

        MessageBox.Show(this, "Você precisa informar como os dados são organizados.
Clique na figura que representa como seus dados estão organizados.", "Erro ao analisar
dados", MessageBoxButtons.OK, MessageBoxIcon.Error);

        return false;

    }

} else if (TransformPanel.Visible)

{

    //verifica se selecionou as variáveis para transformar o campo

    if (RadioButtonDadosTransformados.Checked)

    {

        if (ComboBoxTransformar1.SelectedIndex == -1)

        {

            MessageBox.Show(this, "Você precisa informar a variável 1 que será o campo
a ser agrupado.", "Erro ao analisar dados", MessageBoxButtons.OK, MessageBoxIcon.Error);

```

```

        ComboBoxTransformar1.Focus();

        return false;
    }

    if (ComboBoxTransformar2.SelectedIndex == -1)
    {

        MessageBox.Show(this, "Você precisa informar a variável 2 que será o campo
        pelo qual a variável 1 será agrupada.", "Erro ao analisar dados", MessageBoxButtons.OK,
        MessageBoxIcon.Error);

        ComboBoxTransformar2.Focus();

        return false;
    }
}

} else if (VariablesPanel.Visible)
{

    //verifica se selecionou as variáveis

    if (VariablesListboxesControl.SelectedListBox.Items.Count == 0)
    {

        MessageBox.Show(this, "Você precisa informar pelo menos uma variável para a
        análise.", "Erro ao analisar dados", MessageBoxButtons.OK, MessageBoxIcon.Error);

        VariablesListboxesControl.Focus();
    }
}

```

```

        return false;
    }

    //se selecionou uma variável, verifica se o usuário preencheu a média

    if (VariablesListboxesControl.SelectedListBox.Items.Count == 1)
    {
        decimal mean;

        if (!Decimal.TryParse(MeanTextBox.Text, out mean))
        {
            MessageBox.Show(this, "Você precisa informar uma média válida para a
população.", "Erro ao analisar dados", MessageBoxButtons.OK, MessageBoxIcon.Error);

            MeanTextBox.Focus();

            return false;
        }
    }
}

//----- agora exige os resultados -----

--

//seções do relatório

```



```
ResultsTextSection relatorioTestesNormalidade = new  
ResultsTextSection("Normalidade");
```

```
ResultsTextSection relatorioTestesHipotese = new ResultsTextSection("Teste de  
hipótese");
```

```
ResultsTextSection relatorioTestesIgualdadeVariancias = new  
ResultsTextSection("Igualdade de variâncias");
```

```
ResultsTextSection relatorioTestesEsfericidade = new ResultsTextSection("Teste de  
esfericidade");
```

```
ResultsTextSection relatorioBonferroni = new ResultsTextSection("Teste de  
Bonferroni");
```

```
ResultsText report = new ResultsText("Relatório");
```

```
//tabelas
```

```
NormalityTestsResultsTable tabelaTestesNormalidade = new  
NormalityTestsResultsTable();
```

```
HypothesisTestsResultsTable tabelaTesteHipotese = new  
HypothesisTestsResultsTable();
```

```
EqualityOfVariancesTestsResultsTable tabelaTestesIgualdadeVariancias = new  
EqualityOfVariancesTestsResultsTable();
```

```
SphericityTestsResultsTable tabelaEsfericidade = new SphericityTestsResultsTable();
```

```
ResultsTable tabelaBonferroni = new ResultsTable("Teste de Bonferroni");
```

```
ResultsTables tabelas = new ResultsTables();
```

```

//obtem os resultados

if (VariablesPanel.Visible)

{

    //mostra o panel conforme a opção marcada

    if (IsOneGroup)

    {

        string varname =
VariablesListboxesControl.SelectedListBox.Items[0].ToString();

        //gera a tabela com a estatística descritiva

        DescriptiveTable dt = new DescriptiveTable();

        List<string> l = new List<string>();

        l.Add(varname);

        if (dt.Run(Data, l))

        {

            tabelas.Items.Add(dt.HTMLTable);

        }

        //faz a estatística para uma variável

```

```

OneGroupTestResultTable t = new OneGroupTestResultTable();

ResultsTextSection s = new ResultsTextSection("");

TTest tt = new TTest();

if (tt.Run(Data, varname, Double.Parse(MeanTextBox.Text)))
{
    s.Text.AddRange(tt.Result);

    t.AddRow(varname, tt.PValue, tt.Stat, tt.CriticalValue);
}

else
{
    s.Text.Add("Não foi possível executar o Teste T. Houve um erro interno.");
}

tabelas.Items.Add(t);

report.Items.Add(s);
}

else
{
    //cria as variáveis para processamento do dataset

```

```

int nvar = VariablesListboxesControl.SelectedListBox.Items.Count; //número de
variáveis

bool isAllNormal = true; //todas as variáveis são normais?

bool[] isNormal = new bool[nvar]; //vetor que diz para cada variável se é normal

bool isPaired = PairedRadioButton.Checked; //é pareado?

bool isTwoGroups = (nvar == 2); //são dois grupos?

int n = Data.Rows.Count; //tamanho dos grupos

//nomes das variáveis selecionadas pelo usuário

List<string> varsel = new List<string>();

foreach (string item in VariablesListboxesControl.SelectedListBox.Items)

{

    varsel.Add(item);

}

//gera a tabela com a estatística descritiva

DescriptiveTable dt = new DescriptiveTable();

if (dt.Run(Data, varsel))

{

    tabelas.Items.Add(dt.HTMLTable);

```

```
}
```

```
//-----executa os testes de normalidade-----
```

```
relatorioTestesNormalidade.Text.Add("Utiliza-se o Teste de Shapiro-Wilk  
quando o n é menor ou igual a 50 e o Teste de Kolmogorov-Smirnov quando o n é maior que  
50.");
```

```
relatorioTestesNormalidade.Text.Add("Quando os dados são independentes se  
testa o grupo individualmente, quando os dados são pareados se faz o teste das diferenças  
entre os grupos.");
```

```
relatorioTestesNormalidade.Text.Add("A distribuição dos dados será considerada  
normal se o p-valor for igual ou superior a 0,05.");
```

```
double[] pvalue = new double[varsel.Count];
```

```
string[] normaltest = new string[varsel.Count];
```

```
if (isPaired && nvar == 2)
```

```
{
```

```
    //remove as colunas que não são numéricas
```

```
    List<string> varselnumeric = new List<string>();
```

```
    foreach(string v in varsel)
```

```
    {
```

```

        if (Oraculo.Tests.StatisticalTest.IsNumeric(Data.Columns[v]))
        {
            varselnumeric.Add(v);
        }
    }

    //calcula a diferença entre os grupos (se pareadas)

    DataTable DiferencaEntreGruposData = new DataTable();

    IEnumerable<int[]> combs = Combinatorics.Combinations(2,
varselnumeric.Count);

    Array.Resize(ref pvalue, combs.Count()); //aumenta o array já que agora
guarda as combinações

    Array.Resize(ref normaltest, combs.Count());

    Array.Resize(ref isNormal, combs.Count());

    foreach (int[] c in combs)
    {
        //cria a nova coluna

        string nomecoluna = varselnumeric[c[0] - 1] + " - " + varselnumeric[c[1] -
1];

```

```

DataColumn novacoluna = new DataColumn(nomecoluna, typeof(double));

DiferencaEntreGruposData.Columns.Add(novacoluna);

//obtem o índice das colunas

int c1 = Data.Columns[varselnumeric[c[0] - 1]].Ordinal;

int c2 = Data.Columns[varselnumeric[c[1] - 1]].Ordinal;

//preenche com as diferenças

int rdif = 0;

for (int r = 0; r < Data.Rows.Count; r++)

{

    //não calcula a diferença se um dos valores for nulo

    if (!Data.Rows[r][c1].IsNullOrEquivalent() &&
!Data.Rows[r][c2].IsNullOrEquivalent())

    {

        if (rdif <= DiferencaEntreGruposData.Rows.Count)

        {

            DiferencaEntreGruposData.Rows.Add();

            rdif++;

        }

    }

```

```

        double dif = Convert.ToDouble(Data.Rows[r][c1]) -
Convert.ToDouble(Data.Rows[r][c2]);

        DiferencaEntreGruposData.Rows[rdif -
1][DiferencaEntreGruposData.Columns[nomecoluna].Ordinal] = dif;

    }

}

}

//percorre cada coluna de combinação para fazer o teste de normalidade
for (int i = 0; i < DiferencaEntreGruposData.Columns.Count; i++)
{

    n = DiferencaEntreGruposData.CountNumericValuesAt(i);

    if (n <= 50)
    {

        //executa shapiro para cada variável se n <= 50

        ShapiroTest s = new ShapiroTest();

        if (s.Run(DiferencaEntreGruposData,
DiferencaEntreGruposData.Columns[i].ColumnName))
        {

```



```

        relatorioTestesNormalidade.Text.AddRange(s.Result);

        isNormal[i] = s.IsNormal;

        pvalue[i] = s.PValue;

        normaltest[i] = "Shapiro-Wilk";
    }

    else

    {

        relatorioTestesNormalidade.Text.Add(string.Format("Não foi possível
executar o Teste de Shapiro-Wilk. Houve um erro interno ao processar a diferença entre os
grupos '{0}'.", varsel[i]));

    }

}

else

{

    //executa kolmogorov para variável 1 se n > 50

    KolmogorovSmirnovTest ks = new KolmogorovSmirnovTest();

    if (ks.Run(DiferencaEntreGruposData,
DiferencaEntreGruposData.Columns[i].ColumnName))

    {

        relatorioTestesNormalidade.Text.AddRange(ks.Result);

        isNormal[i] = ks.IsNormal;
    }
}

```

```

        pvalue[i] = ks.PValue;

        normaltest[i] = "Kolmogorov-Smirnov";

    }

    else

    {

        relatorioTestesNormalidade.Text.Add(string.Format("Não foi possível
executar o Teste de Kolmogorov-Smirnov. Houve um erro interno ao processar a diferença
entre os grupos '{0}'.", varsel[i]));

    }

}

//inclui a linha na tabela

tabelaTestesNormalidade.AddRow(DiferencaEntreGruposData.Columns[i].ColumnName,
DiferencaEntreGruposData.CountNumericValuesAt(i), normaltest[i], pvalue[i], isNormal[i]);

//verifica se todas as variáveis são normais

if (!isNormal[i]) { isAllNormal = false; }

}

}

else

{

```

```

//percorre cada variável selecionada para fazer o teste de normalidade

for (int i = 0; i < varsel.Count; i++)

{

    if (n <= 50)

    {

        //executa shapiro para cada variável se n <= 50

        ShapiroTest s = new ShapiroTest();

        if (s.Run(Data, varsel[i]))

        {

            relatorioTestesNormalidade.Text.AddRange(s.Result);

            isNormal[i] = s.IsNormal;

            pvalue[i] = s.PValue;

            normaltest[i] = "Shapiro-Wilk";

        }

        else

        {

            relatorioTestesNormalidade.Text.Add(string.Format("Não foi possível
executar o Teste de Shapiro-Wilk. Houve um erro interno ao processar a variável '{0}'.",
varsel[i]));

        }

    }

}

```

```

else

{

//executa kolmogorov para variável 1 se n > 50

KolmogorovSmirnovTest ks = new KolmogorovSmirnovTest();

if (ks.Run(Data, varsel[i]))

{

relatorioTestesNormalidade.Text.AddRange(ks.Result);

isNormal[i] = ks.IsNormal;

pvalue[i] = ks.PValue;

normaltest[i] = "Kolmogorov-Smirnov";

}

else

{

relatorioTestesNormalidade.Text.Add(string.Format("Não foi possível
executar o Teste de Kolmogorov-Smirnov. Houve um erro interno ao processar a variável
'{0}'.", varsel[i]));

}

}

//inclui a linha na tabela

tabelaTestesNormalidade.AddRow(varsel[i],
Data.CountNumericValuesAt(varsel[i]), normaltest[i], pvalue[i], isNormal[i]);

```

```

//verifica se todas as variáveis são normais

if (!isNormal[i]) { isAllNormal = false; }

}

}

//constrói a tabela de teste de normalidade

tabelas.Items.Add(tabelaTestesNormalidade);

//-----executa os testes de hipótese-----

//é normal (p > 0.05)

if (isAllNormal)

{

//é pareado?

if (isPaired)

{

//dois grupos?

if (isTwoGroups)

{

```

```

//executa o teste t para dados pareados -> t.test (... , paired = TRUE)

PairedTTest ptt = new PairedTTest();

if (ptt.Run(Data, varsel[0], varsel[1]))

{

    relatorioTestesHipotese.Text.AddRange(ptt.Result);

    tabelaTesteHipotese.AddRow("Teste T pareado", ptt.PValue, ptt.Stat,
ptt.CriticalValue);

}

else

{

    relatorioTestesHipotese.Text.Add("Não foi possível executar o Teste T
para dados pareados. Houve um erro interno.");

}

}

else

{

    //executa o teste ANOVA para medidas repetidas repeated measures
anova

    RepeatedMeasuresAnovaTest anova = new
RepeatedMeasuresAnovaTest();

    bool exibirAnova = false;

```

```

bool calcularBonferroni = false;

if (anova.Run(Data, varsel))
{
    //testa a esfericidade com o teste de mauchly
    Oraculo.Tests.MauchlyTest m = new Oraculo.Tests.MauchlyTest();

    if (m.Run(Data, varsel))
    {
        relatorioTestesEsfericidade.Text.AddRange(m.Result);
        tabelaEsfericidade.AddRow("Teste de Mauchly", m.PValue);

        if (m.AreSpherical)
        {
            //há esfericidade: retorna os resultados do teste ANOVA para
medidas repetidas

            exhibirAnova = true;

            if (!anova.AreEqual) calcularBonferroni = true;
        }
        else

```

```

{

    //não há esfericidade: faz a correção EGG-EHF

    Oraculo.Tests.GreenhouseGeisserEpsilonTest gge = new
Oraculo.Tests.GreenhouseGeisserEpsilonTest();

    if (gge.Run(Data, varsel, ANOVATests.RepeatedMeasures))

    {

        //não há diferença significativa

        if (gge.AreEqual)

        {

            //retorna os resultados do teste ANOVA para medidas
repetidas (sem correção mesmo)

            exhibirAnova = true;

        }

        else

        {

            //faz ajuste EHF

            Oraculo.Tests.HuynhFeldtEpsilonTest hfe = new
Oraculo.Tests.HuynhFeldtEpsilonTest();

            if (hfe.Run(Data, varsel, ANOVATests.RepeatedMeasures,
gge.GGE))

            {

```



```

//há diferença significativa?

if (hfe.AreEqual)

{

//retorna os resultados do teste ANOVA para medidas
repetidas (sem correção mesmo)

    exibirAnova = true;

}

else

{

//retorna os resultados do teste ANOVA para medidas
repetidas (com a correção do HFE)

    hfe.CriticalValue = anova.CriticalValue;

    relatorioTestesHipotese.Text.AddRange(hfe.Result);

    tabelaTesteHipotese.AddRow("ANOVA para medidas
repetidas ajustado nos graus de liberdade (HFE)", hfe.PValue, hfe.Stat, hfe.CriticalValue);

    calcularBonferroni = true;

}

}

else

{

```

```
relatorioTestesHipotese.Text.Add("Não foi possível  
executar o Teste ANOVA para medidas repetidas com a correção do Epsilon de Huynh e  
Feldt. Houve um erro interno.");
```

```
}
```

```
}
```

```
}
```

```
else
```

```
{
```

```
relatorioTestesHipotese.Text.Add("Não foi possível executar o  
Teste ANOVA para medidas repetidas com a correção do Epsilon de Greenhouse e Geisser.  
Houve um erro interno.");
```

```
}
```

```
}
```

```
//exibe ANOVA se há esfericidade e a diferença não foi significativa
```

```
if (exibirAnova)
```

```
{
```

```
relatorioTestesHipotese.Text.AddRange(anova.Result);
```

```
tabelaTesteHipotese.AddRow("ANOVA para medidas repetidas",  
anova.PValue, anova.Stat, anova.CriticalValue);
```

```
}
```

```

//calcula bonferroni se há diferença significativa

if (calcularBonferroni)

{

    //faz bonferroni

    Oraculo.Tests.BonferroniTest b = new
Oraculo.Tests.BonferroniTest();

    if (b.Run(Data, varse1))

    {

        //relatorioBonferroni.Text.AddRange(b.Result);

        relatorioBonferroni.Text.AddRange(b.Result);

        tabelaBonferroni.Data = b.ResultsTable.Data;

    }

    else

    {

        relatorioBonferroni.Text.Add("Não foi possível executar o
Teste de Bonferroni. Houve um erro interno.");

    }

}

}

else

{

```

```
relatorioTestesHipotese.Text.Add("Não foi possível executar o Teste  
de Mauchly. Houve um erro interno.");
```

```
}
```

```
}
```

```
else
```

```
{
```

```
relatorioTestesHipotese.Text.Add("Não foi possível executar o Teste  
ANOVA para medidas repetidas. Houve um erro interno.");
```

```
}
```

```
}
```

```
}
```

```
else
```

```
{
```

```
//dois grupos?
```

```
if (isTwoGroups)
```

```
{
```

```
//faz o teste de Levene
```

```
LeveneTest l = new LeveneTest();
```

```
if (l.Run(Data, varsel))
```

```
{
```

```
relatorioTestesIgualdadeVariancias.Text.AddRange(l.Result);
```

```

tabelaTestesIgualdadeVariancias.AddRow("Teste de Levene",
l.PValue);

//é homogêneo?

if (l.AreEqual)

{

//teste t independente

TTest nptt = new TTest();

if (nptt.Run(Data, varsel[0], varsel[1]))

{

relatorioTestesHipotese.Text.AddRange(nptt.Result);

tabelaTesteHipotese.AddRow("Teste T independente",
nptt.PValue, nptt.Stat, nptt.CriticalValue);

}

else

{

relatorioTestesHipotese.Text.Add("Não foi possível executar o
Teste T para dados não pareados. Houve um erro interno.");

}

}

else

```

```

{
    //faz o Teste T para variâncias diferentes pressupondo diferença de
variâncias

    TwoSampleTTestTest t = new TwoSampleTTestTest();

    if (t.Run(Data, varsel[0], varsel[1]))

    {

        relatorioTestesHipotese.Text.AddRange(t.Result);

        tabelaTesteHipotese.AddRow("Teste T para variâncias
diferentes", t.PValue, t.Stat, t.CriticalValue);

    }

    else

    {

        relatorioTestesHipotese.Text.Add("Não foi possível executar o
Teste T para variâncias diferentes. Houve um erro interno.");

    }

}

}

else

{

    relatorioTestesIgualdadeVariancias.Text.Add("Não foi possível
executar o Teste de Levene. Houve um erro interno.");

```

```

    }

}

else

{

    //faz o teste de Levene

    LeveneTest l = new LeveneTest();

    if (l.Run(Data, varsel))

    {

        relatorioTestesIgualdadeVariancias.Text.AddRange(l.Result);

        tabelaTestesIgualdadeVariancias.AddRow("Teste de Levene",

l.PValue);

        //é homogêneo?

        if (l.AreEqual)

        {

            //faz anova

            //executa o teste ANOVA (One Way ANOVA)

            Oraculo.Tests.OneWayAnovaTest anova = new

Oraculo.Tests.OneWayAnovaTest();

            if (anova.Run(Data, varsel))

```

```

    {

        relatorioTestesHipotese.Text.AddRange(anova.Result);

        tabelaTesteHipotese.AddRow("ANOVA", anova.PValue,
anova.Stat, anova.CriticalValue);

    }

    if (!anova.AreEqual)

    {

        //faz bonferroni

        Oraculo.Tests.BonferroniTest b = new
Oraculo.Tests.BonferroniTest();

        if (b.Run(Data, varsel))

        {

            //relatorioBonferroni.Text.AddRange(b.Result);

            relatorioBonferroni.Text.AddRange(b.Result);

            tabelaBonferroni.Data = b.ResultsTable.Data;

        }

        else

        {

            relatorioBonferroni.Text.Add("Não foi possível executar o
Teste de Bonferroni. Houve um erro interno.");

        }

    }

```



```

    }

}

else

{

    relatorioTestesHipotese.Text.Add("Não foi possível executar o
Teste ANOVA. Houve um erro interno.");

}

}

else

{

    //executa o Teste T de Welch se não é igual

    WelchTest w = new WelchTest();

    if (w.Run(Data, varsel))

    {

        relatorioTestesHipotese.Text.AddRange(w.Result);

        tabelaTesteHipotese.AddRow("Teste de Teste T de Welch",
w.PValue, w.Stat, w.CriticalValue);

    }

    if (!w.AreEqual)

    {

        //faz bonferroni

```

```

Oraculo.Tests.BonferroniTest b = new
Oraculo.Tests.BonferroniTest();

    if (b.Run(Data, vassel))

        {

            //relatorioBonferroni.Text.AddRange(b.Result);

            relatorioBonferroni.Text.AddRange(b.Result);

            tabelaBonferroni.Data = b.ResultsTable.Data;

        }

    else

        {

            relatorioBonferroni.Text.Add("Não foi possível executar o
Teste de Bonferroni. Houve um erro interno.");

        }

    }

}

else

    {

        relatorioTestesHipotese.Text.Add("Não foi possível executar o
Teste de Welch. Houve um erro interno.");

    }

}

```

```

    }

    else

    {

        relatorioTestesIgualdadeVariancias.Text.Add("Não foi possível
executar o Teste de Levene. Houve um erro interno.");

    }

}

}

}

else //não é normal

{

    //é pareado?

    if (isPaired)

    {

        //dois grupos?

        if (isTwoGroups)

        {

            //executa o teste de Wilcoxon

            WilcoxonTest wt = new WilcoxonTest();

            if (wt.Run(Data, varsel[0], varsel[1]))

```

```

    {

        relatorioTestesHipotese.Text.AddRange(wt.Result);

        tabelaTesteHipotese.ShowCriticalValue = false;

        tabelaTesteHipotese.AddRow("Teste de Wilcoxon", wt.PValue,
wt.Stat, wt.CriticalValue);

    }

    else

    {

        relatorioTestesHipotese.Text.Add("Não foi possível executar o Teste
de Wilcoxon. Houve um erro interno.");

    }

}

else

{

    //executa o teste de friedman

    FriedmanTest f = new FriedmanTest();

    if (f.Run(Data, varsel))

    {

        relatorioTestesHipotese.Text.AddRange(f.Result);

        tabelaTesteHipotese.ShowCriticalValue = false;

```

```
tabelaTesteHipotese.AddRow("Teste de Friedman", f.PValue, f.Stat,  
f.CriticalValue);
```

```
if (!f.AreEqual)  
{  
    //faz bonferroni  
  
    Oraculo.Tests.BonferroniTest b = new  
Oraculo.Tests.BonferroniTest();  
  
    if (b.Run(Data, varsel))  
    {  
        //relatorioBonferroni.Text.AddRange(b.Result);  
  
        relatorioBonferroni.Text.AddRange(b.Result);  
  
        tabelaBonferroni.Data = b.ResultsTable.Data;  
    }  
  
    else  
    {  
        relatorioBonferroni.Text.Add("Não foi possível executar o Teste  
de Bonferroni. Houve um erro interno.");  
    }  
}  
}
```

```

else

{

    relatorioTestesHipotese.Text.Add("Não foi possível executar o Teste
de Friedman. Houve um erro interno.");

}

}

}

else

{

    //dois grupos?

    if (isTwoGroups)

    {

        //executa o teste de Mann-Whitney-Wilcoxon

        ManWhitneyTest mt = new ManWhitneyTest();

        if (mt.Run(Data, varsel[0], varsel[1]))

        {

            relatorioTestesHipotese.Text.AddRange(mt.Result);

            tabelaTesteHipotese.ShowCriticalValue = false;

            tabelaTesteHipotese.AddRow("Teste de Mann-Whitney", mt.PValue,
mt.Stat, mt.CriticalValue);

        }

    }

}

```

```

else

{

    relatorioTestesHipotese.Text.Add("Não foi possível executar o Teste
de Mann-Whitney. Houve um erro interno.");

}

}

else

{

    //executa o teste de Kruskal-Wallis

    KruskalWallisTest kwt = new KruskalWallisTest();

    if (kwt.Run(Data, varsel))

    {

        relatorioTestesHipotese.Text.AddRange(kwt.Result);

        tabelaTesteHipotese.ShowCriticalValue = false;

        tabelaTesteHipotese.AddRow("Teste de Kruskal-Wallis", kwt.PValue,
kwt.Stat, kwt.CriticalValue);

        if (!kwt.AreEqual)

        {

            //faz bonferroni

```

```

        Oraculo.Tests.BonferroniTest b = new
Oraculo.Tests.BonferroniTest();

        if (b.Run(Data, varsel))

        {

            //relatorioBonferroni.Text.AddRange(b.Result);

            relatorioBonferroni.Text.AddRange(b.Result);

            tabelaBonferroni.Data = b.ResultsTable.Data;

        }

        else

        {

            relatorioBonferroni.Text.Add("Não foi possível executar o Teste
de Bonferroni. Houve um erro interno.");

        }

    }

}

else

{

    relatorioTestesHipotese.Text.Add("Não foi possível executar o Teste
de Kruskal-Wallis. Houve um erro interno.");

}

}

```



```
    }  
}  
  
//monta a tabela de post hoc  
  
tabelas.Items.Add(tabelaTestesIgualdadeVariancias);  
  
tabelas.Items.Add(tabelaEsfericidade);  
  
tabelas.Items.Add(tabelaTesteHipotese);  
  
tabelas.Items.Add(tabelaBonferroni);  
  
}
```

```
//monta o texto de análise  
  
report.Items.Add(relatorioTestesNormalidade);  
  
report.Items.Add(relatorioTestesIgualdadeVariancias);  
  
report.Items.Add(relatorioTestesEsfericidade);  
  
report.Items.Add(relatorioTestesHipotese);  
  
report.Items.Add(relatorioBonferroni);
```

```
//verifica se há algum resultado para mostrar e exibe  
  
if (!report.IsNullOrEmpty())  
  
{
```

```

        string html = "<h1>Resultado da análise</h1>";

        html += tabelas.ToHTML();

        html += report.ToHTML();

        ReportViewerFired.Invoke(html);

    }

    else

    {

        ReportViewerFired.Invoke("<p>Houve erro ao processar a solicitação</p>");

        return false;

    }

}

return true;

}

public override void ShowStep()

{

    //esconde todos os panels

    DataArrangePanel.Visible = false;

```

```
TransformPanel.Visible = false;

VariablesPanel.Visible = false;

ResultPanel.Visible = false;

ClipboardButton.Visible = false;

//exibe os panels conforme as opções marcadas

DataArrangePanel.Visible = (Step == 1);

if (DataArrange == 1)
{
    VariablesPanel.Visible = (Step == 2);

    ResultPanel.Visible = (Step == 3);
}

else
{
    TransformPanel.Visible = (Step == 2);

    VariablesPanel.Visible = (Step == 3);

    ResultPanel.Visible = (Step == 4);
}
```

```

//se há dados, preenche os combos e lists

if (Data != null)

{

    if (VariablesPanel.Visible)

    {

VariablesListboxesControl.FillListBoxFromData(VariablesListboxesControl.UnselectedListB
ox, VariablesListboxesControl.SelectedListBox, Data, "");

    }

}

base.ShowStep();

PreviousButton.Visible = !DataArrangePanel.Visible;

NextButton.Visible = !DataArrangePanel.Visible;

if (VariablesPanel.Visible && DataArrange == 1 &&
VariablesListboxesControl.SelectedListBox.SelectedItems.Count >= 1) NextButton.Text =
"Concluir";

if (ResultPanel.Visible) NextButton.Enabled = false;

}

```

```

private void TransformData()

{

    //restaura o backup e substitui pelos dados originais

    if (DataBackup != null) Data = DataBackup.Copy();

    //busca os nomes dos dois campos: o campo1 é agrupado pelo campo 2 (nome das
colunas)

    string campo1 = ComboBoxTransformar1.SelectedItem.ToString();

    string campo2 = ComboBoxTransformar2.SelectedItem.ToString();

    //limpa a tabela

    DataTable dtNovoData = new DataTable();

    //agrupa os valores do primeiro campo para identificar as colunas

    DataTable dtColunas = Data.AsEnumerable()

        .GroupBy(r => new { Col1 = r[campo2] })

        .Select(g => g.OrderBy(r => r[campo2]).First())

        .CopyToDataTable();

    //cria as colunas da nova tabela

```

```

foreach (DataRow linha in dtColunas.Rows)

{

    string newColName = linha[campo2].ToString();

    DataColumnCollection columns = dtColunas.Columns;

    if (!columns.Contains(newColName))

    {

        Type tipo = dtColunas.Columns[campo1].DataType;

        dtNovoData.Columns.Add(newColName, tipo);

    }

}

//preenche as linhas

foreach (DataRow linha in Data.Rows)

{

    DataRow linhaModificada = null;

    bool newRow = false;

    //verifica se já tem uma linha ou se precisa inserir uma nova

    if (dtNovoData.Rows.Count > 0)

    {

```

```

foreach (DataRow linhaExistente in dtNovoData.Rows)

{

    if (linhaExistente[linha[campo2].ToString()].ToString().IsNullOrEquivalent())

    {

        linhaModificada = linhaExistente;

        break;

    }

}

if (linhaModificada == null)

{

    linhaModificada = dtNovoData.NewRow();

    newRow = true;

}

//preenche o valor na coluna certa

linhaModificada[linha[campo2].ToString()] = linha[campo1];

//insere ou atualiza a coluna

```

```
        if (newRow) dtNovoData.Rows.Add(linhaModificada);
    }

    //faz o backup e substitui pelos novos dados

    if (DataBackup == null) DataBackup = Data.Copy();

    Data = dtNovoData;

    //preenche o datagrid

    DataGrid.DataSource = Data;
}

public void EnableTransformFields(bool value)
{
    LabelTransformar1.Enabled = value;

    ComboBoxTransformar1.Enabled = value;

    LabelTransformar2.Enabled = value;

    ComboBoxTransformar2.Enabled = value;

    if (value)
    {
```



```

        //preenche o primeiro combobox

        FillComboBoxData(ComboBoxTransformar1);

    }

    else

    {

        //volta aos dados originais e os mostra no grid

        if (DataBackup != null) Data = DataBackup;

        DataGrid.DataSource = this.Data;

    }

}

#endregion

private void ClipboardButton_Click(object sender, EventArgs e)

{

    CopyRTFToClipboard();

}

private void RadioButtonDadosDigitados_CheckedChanged(object sender, EventArgs e)

{

```

```

        EnableTransformFields(false);

    }

    private void RadioButtonDadosTransformados_CheckedChanged(object sender,
EventArgs e)
    {

        //habilita novamente os campos para recriar a tabela modificada

        EnableTransformFields(true);

    }

    private void ComboBoxTransformar1_SelectedIndexChanged(object sender, EventArgs
e)
    {

        //se não está no modo tabela transformada

        if (!RadioButtonDadosTransformados.Checked) return;

        //volta os dados originais

        if (DataBackup != null) Data = DataBackup.Copy();

        FillComboBoxData(ComboBoxTransformar2, ComboBoxTransformar1);

    }

```

```

private void ComboBoxTransformar2_SelectedIndexChanged(object sender, EventArgs
e)
{
    //se não está no modo tabela transformada

    if (!RadioButtonDadosTransformados.Checked) return;

    //se não selecionou os dois campos, sai sem fazer modificações

    if ((ComboBoxTransformar1.SelectedIndex == -1) &&
(ComboBoxTransformar2.SelectedIndex == -1)) return;

    //gera novo datatable com campo 1 agrupado por campo 2

    TransformData();
}

private void TransformPanel_VisibleChanged(object sender, EventArgs e)

{

    if (TransformPanel.Visible)

        if (RadioButtonDadosDigitados.Checked)

            EnableTransformFields(false);

}

```

```
private void PictureBoxGroup1_Click(object sender, EventArgs e)
```

```
{
```

```
    this.DataArrange = 1;
```

```
    ShowStep(2);
```

```
}
```

```
private void PictureBoxGroup2_Click(object sender, EventArgs e)
```

```
{
```

```
    this.DataArrange = 2;
```

```
    ShowStep(2);
```

```
}
```

```
private void VariablesListboxesControl_SelectedListBoxItemsChanged()(object  
sender, EventArgs e)
```

```
{
```

```
    PairedGroupBox.Visible = (VariablesListboxesControl.SelectedListBox.Items.Count  
> 1);
```

```
    MeanLabel.Visible = (VariablesListboxesControl.SelectedListBox.Items.Count == 1);
```

```
    MeanTextBox.Visible = (VariablesListboxesControl.SelectedListBox.Items.Count ==  
1);
```

```
    }  
  }  
}
```

Oraculo\Screens\Analyze.Designer.cs

```
?namespace Oraculo.Telas
```

```
{
```

```
    partial class Analyze
```

```
    {
```

```
        /// <summary>
```

```
        /// Required designer variable.
```

```
        /// </summary>
```

```
        private System.ComponentModel.IContainer components = null;
```

```
        /// <summary>
```

```
        /// Clean up any resources being used.
```

```
        /// </summary>
```

```
        /// <param name="disposing">true if managed resources should be disposed; otherwise,  
        false.</param>
```

```

protected override void Dispose(bool disposing)

{

    if (disposing && (components != null))

    {

        components.Dispose();

    }

    base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

    this.components = new System.ComponentModel.Container();

    System.ComponentModel.ComponentResourceManager resources = new

System.ComponentModel.ComponentResourceManager(typeof(Analyze));

```

```
this.DataArrangePanel = new System.Windows.Forms.Panel();

this.DataArrangeOnePictureBox2 = new System.Windows.Forms.PictureBox();

this.DataArrangeOnePictureBox = new System.Windows.Forms.PictureBox();

this.DataArrangeLabel2 = new System.Windows.Forms.Label();

this.DataArrangeLabel = new System.Windows.Forms.Label();

this.VariablesPanel = new System.Windows.Forms.Panel();

this.MeanTextBox = new System.Windows.Forms.TextBox();

this.label1 = new System.Windows.Forms.Label();

this.MeanLabel = new System.Windows.Forms.Label();

this.PairedGroupBox = new System.Windows.Forms.GroupBox();

this.infoPareado = new Oraculo.Global.Info();

this.PairedRadioButton = new System.Windows.Forms.RadioButton();

this.NotPairedRadioButton = new System.Windows.Forms.RadioButton();

this.VariablesListboxesControl = new
Oraculo.Global.MoveBetwenListboxesControl();

this.ResultPanel = new System.Windows.Forms.Panel();

this.ResultWebBrowser = new System.Windows.Forms.WebBrowser();

this.ResultLabel = new System.Windows.Forms.Label();

this.ClipboardButton = new System.Windows.Forms.Button();

this.toolTip1 = new System.Windows.Forms.ToolTip(this.components);
```

```
this.TransformPanel = new System.Windows.Forms.Panel();

this.ComboBoxTransformar2 = new System.Windows.Forms.ComboBox();

this.LabelTransformar2 = new System.Windows.Forms.Label();

this.ComboBoxTransformar1 = new System.Windows.Forms.ComboBox();

this.LabelTransformar1 = new System.Windows.Forms.Label();

this.DataGrid = new System.Windows.Forms.DataGridView();

this.LabelPreviewDados = new System.Windows.Forms.Label();

this.RadioButtonDadosTransformados = new System.Windows.Forms.RadioButton();

this.RadioButtonDadosDigitados = new System.Windows.Forms.RadioButton();

this.LabelTransform = new System.Windows.Forms.Label();

this.ButtonPanel.SuspendLayout();

this.DataArrangePanel.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.DataArrangeOnePictureBox2)).BeginInit(
);

((System.ComponentModel.ISupportInitialize)(this.DataArrangeOnePictureBox)).BeginInit();

this.VariablesPanel.SuspendLayout();

this.PairedGroupBox.SuspendLayout();

this.ResultPanel.SuspendLayout();

this.TransformPanel.SuspendLayout();
```



```
((System.ComponentModel.ISupportInitialize)(this.DataGrid)).BeginInit();

this.SuspendLayout();

//

// sectionHeader1

//

this.sectionHeader1.BreadCrumbText = "Início > Analisar";

this.sectionHeader1.Size = new System.Drawing.Size(838, 119);

//

// TitleLabel

//

this.TitleLabel.Size = new System.Drawing.Size(838, 69);

this.TitleLabel.Text = "Analisar";

//

// ButtonPanel

//

this.ButtonPanel.Controls.Add(this.ClipboardButton);

this.ButtonPanel.Location = new System.Drawing.Point(0, 567);

this.ButtonPanel.Margin = new System.Windows.Forms.Padding(23);

this.ButtonPanel.Size = new System.Drawing.Size(838, 69);

this.ButtonPanel.Controls.SetChildIndex(this.PreviousButton, 0);
```

```
this.ButtonPanel.Controls.SetChildIndex(this.NextButton, 0);

this.ButtonPanel.Controls.SetChildIndex(this.ClipboardButton, 0);

//

// NextButton

//

this.NextButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;

this.NextButton.Location = new System.Drawing.Point(524, 21);

this.NextButton.Margin = new System.Windows.Forms.Padding(7, 27, 27, 27);

//

// PreviousButton

//

this.PreviousButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;

this.PreviousButton.Location = new System.Drawing.Point(437, 20);

this.PreviousButton.Margin = new System.Windows.Forms.Padding(27, 27, 0, 27);

//

// DataArrangePanel

//

this.DataArrangePanel.AutoScroll = true;

this.DataArrangePanel.Controls.Add(this.DataArrangeOnePictureBox2);

this.DataArrangePanel.Controls.Add(this.DataArrangeOnePictureBox);
```

```
this.DataArrangePanel.Controls.Add(this.DataArrangeLabel2);

this.DataArrangePanel.Controls.Add(this.DataArrangeLabel);

this.DataArrangePanel.Dock = System.Windows.Forms.DockStyle.Fill;

this.DataArrangePanel.Location = new System.Drawing.Point(0, 188);

this.DataArrangePanel.Name = "DataArrangePanel";

this.DataArrangePanel.Padding = new System.Windows.Forms.Padding(7);

this.DataArrangePanel.Size = new System.Drawing.Size(838, 379);

this.DataArrangePanel.TabIndex = 0;

//

// DataArrangeOnePictureBox2

//

this.DataArrangeOnePictureBox2.Anchor =
System.Windows.Forms.AnchorStyles.Top;

this.DataArrangeOnePictureBox2.Image =
global::Oraculo.Properties.Resources.DataByLines;

this.DataArrangeOnePictureBox2.Location = new System.Drawing.Point(428, 70);

this.DataArrangeOnePictureBox2.MaximumSize = new System.Drawing.Size(307,
195);

this.DataArrangeOnePictureBox2.Name = "DataArrangeOnePictureBox2";

this.DataArrangeOnePictureBox2.Size = new System.Drawing.Size(307, 195);
```

```

        this.DataArrangeOnePictureBox2.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.Zoom;

        this.DataArrangeOnePictureBox2.TabIndex = 5;

        this.DataArrangeOnePictureBox2.TabStop = false;

        this.DataArrangeOnePictureBox2.Click += new
System.EventHandler(this.PictureBoxGroup2_Click);

        //

        // DataArrangeOnePictureBox

        //

        this.DataArrangeOnePictureBox.Anchor =
System.Windows.Forms.AnchorStyles.Top;

        this.DataArrangeOnePictureBox.Image =
global::Oraculo.Properties.Resources.DataByColumns;

        this.DataArrangeOnePictureBox.Location = new System.Drawing.Point(104, 70);

        this.DataArrangeOnePictureBox.MaximumSize = new System.Drawing.Size(307,
195);

        this.DataArrangeOnePictureBox.Name = "DataArrangeOnePictureBox";

        this.DataArrangeOnePictureBox.Size = new System.Drawing.Size(307, 195);

        this.DataArrangeOnePictureBox.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.Zoom;

        this.DataArrangeOnePictureBox.TabIndex = 4;

        this.DataArrangeOnePictureBox.TabStop = false;

```

```

        this.DataArrangeOnePictureBox.Click += new
System.EventHandler(this.PictureBoxGroup1_Click);

//

// DataArrangeLabel2

//

this.DataArrangeLabel2.Anchor = System.Windows.Forms.AnchorStyles.Top;

this.DataArrangeLabel2.AutoSize = true;

this.DataArrangeLabel2.Location = new System.Drawing.Point(303, 41);

this.DataArrangeLabel2.Margin = new System.Windows.Forms.Padding(20);

this.DataArrangeLabel2.Name = "DataArrangeLabel2";

this.DataArrangeLabel2.Size = new System.Drawing.Size(233, 15);

this.DataArrangeLabel2.TabIndex = 3;

this.DataArrangeLabel2.Text = "Clique na figura que representa seus dados";

//

// DataArrangeLabel

//

this.DataArrangeLabel.Anchor = System.Windows.Forms.AnchorStyles.Top;

this.DataArrangeLabel.AutoSize = true;

this.DataArrangeLabel.Font = new System.Drawing.Font("Segoe UI", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.DataArrangeLabel.Location = new System.Drawing.Point(274, 0);

```

```
this.DataArrangeLabel.Margin = new System.Windows.Forms.Padding(20);

this.DataArrangeLabel.Name = "DataArrangeLabel";

this.DataArrangeLabel.Size = new System.Drawing.Size(290, 21);

this.DataArrangeLabel.TabIndex = 0;

this.DataArrangeLabel.Text = "Como estão organizados os seus dados?";

//

// VariablesPanel

//

this.VariablesPanel.AutoScroll = true;

this.VariablesPanel.Controls.Add(this.MeanTextBox);

this.VariablesPanel.Controls.Add(this.label1);

this.VariablesPanel.Controls.Add(this.MeanLabel);

this.VariablesPanel.Controls.Add(this.PairedGroupBox);

this.VariablesPanel.Controls.Add(this.VariablesListboxesControl);

this.VariablesPanel.Dock = System.Windows.Forms.DockStyle.Fill;

this.VariablesPanel.Location = new System.Drawing.Point(0, 188);

this.VariablesPanel.Margin = new System.Windows.Forms.Padding(0);

this.VariablesPanel.Name = "VariablesPanel";

this.VariablesPanel.Size = new System.Drawing.Size(838, 379);

this.VariablesPanel.TabIndex = 0;
```

```
//  
  
// MeanTextBox  
  
//  
  
this.MeanTextBox.Location = new System.Drawing.Point(31, 265);  
  
this.MeanTextBox.Name = "MeanTextBox";  
  
this.MeanTextBox.Size = new System.Drawing.Size(116, 23);  
  
this.MeanTextBox.TabIndex = 18;  
  
//  
  
// label1  
  
//  
  
this.label1.AutoSize = true;  
  
this.label1.Font = new System.Drawing.Font("Segoe UI", 12F,  
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));  
  
this.label1.Location = new System.Drawing.Point(20, 0);  
  
this.label1.Name = "label1";  
  
this.label1.Size = new System.Drawing.Size(210, 21);  
  
this.label1.TabIndex = 19;  
  
this.label1.Text = "Que variáveis deseja utilizar?";  
  
//  
  
// MeanLabel
```

```

//

this.MeanLabel.AutoSize = true;

this.MeanLabel.Location = new System.Drawing.Point(28, 241);

this.MeanLabel.Name = "MeanLabel";

this.MeanLabel.Size = new System.Drawing.Size(222, 15);

this.MeanLabel.TabIndex = 17;

this.MeanLabel.Text = "Média que se deseja fazer a comparação:";

//

// PairedGroupBox

//

this.PairedGroupBox.Controls.Add(this.infoPareado);

this.PairedGroupBox.Controls.Add(this.PairedRadioButton);

this.PairedGroupBox.Controls.Add(this.NotPairedRadioButton);

this.PairedGroupBox.Location = new System.Drawing.Point(24, 241);

this.PairedGroupBox.Name = "PairedGroupBox";

this.PairedGroupBox.Size = new System.Drawing.Size(344, 106);

this.PairedGroupBox.TabIndex = 16;

this.PairedGroupBox.TabStop = false;

this.PairedGroupBox.Text = "Os dados são:";

```



```
        this.toolTip1.SetToolTip(this.PairedGroupBox, "Informe se as amostras são de dados  
pareados ou se não são pareadas");
```

```
//
```

```
// infoPareado
```

```
//
```

```
        this.infoPareado.BackgroundImage =  
        ((System.Drawing.Image)(resources.GetObject("infoPareado.BackgroundImage")));
```

```
        this.infoPareado.BackgroundImageLayout =  
        System.Windows.Forms.ImageLayout.Center;
```

```
        this.infoPareado.Location = new System.Drawing.Point(93, -3);
```

```
        this.infoPareado.Message = "";
```

```
        this.infoPareado.Name = "infoPareado";
```

```
        this.infoPareado.Size = new System.Drawing.Size(24, 24);
```

```
        this.infoPareado.TabIndex = 2;
```

```
        this.infoPareado.TabStop = false;
```

```
//
```

```
// PairedRadioButton
```

```
//
```

```
        this.PairedRadioButton.AutoSize = true;
```

```
        this.PairedRadioButton.Checked = true;
```

```
        this.PairedRadioButton.Location = new System.Drawing.Point(34, 36);
```

```
this.PairedRadioButton.Name = "PairedRadioButton";

this.PairedRadioButton.Size = new System.Drawing.Size(153, 19);

this.PairedRadioButton.TabIndex = 0;

this.PairedRadioButton.TabStop = true;

this.PairedRadioButton.Text = "Pareados (Dependentes)";

this.PairedRadioButton.UseVisualStyleBackColor = true;

//

// NotPairedRadioButton

//

this.NotPairedRadioButton.AutoSize = true;

this.NotPairedRadioButton.Location = new System.Drawing.Point(34, 62);

this.NotPairedRadioButton.Name = "NotPairedRadioButton";

this.NotPairedRadioButton.Size = new System.Drawing.Size(187, 19);

this.NotPairedRadioButton.TabIndex = 1;

this.NotPairedRadioButton.TabStop = true;

this.NotPairedRadioButton.Text = "N\u00e3o pareados (Independentes)";

this.NotPairedRadioButton.UseVisualStyleBackColor = true;

//

// VariablesListboxesControl

//
```

```

        this.VariablesListboxesControl.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)

        | System.Windows.Forms.AnchorStyles.Right)));

        this.VariablesListboxesControl.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

        this.VariablesListboxesControl.Location = new System.Drawing.Point(20, 41);

        this.VariablesListboxesControl.Margin = new System.Windows.Forms.Padding(20);

        this.VariablesListboxesControl.Name = "VariablesListboxesControl";

        this.VariablesListboxesControl.SelectedCaption = "Variável que se deseja utilizar:";

        this.VariablesListboxesControl.Size = new System.Drawing.Size(798, 177);

        this.VariablesListboxesControl.TabIndex = 15;

        this.toolTip1.SetToolTip(this.VariablesListboxesControl, "Selecione uma ou mais
variáveis de teste");

        this.VariablesListboxesControl.UnselectedCaption = "Variáveis importadas:";

        this.VariablesListboxesControl.SelectedListBoxItemsChanged += new
Oraculo.Global.ListBoxItemsChangedEvent(this.VariablesListboxesControl_SelectedListBox
ItemsChanged);

        //

        // ResultPanel

        //

        this.ResultPanel.Controls.Add(this.ResultWebBrowser);

```

```

this.ResultPanel.Controls.Add(this.ResultLabel);

this.ResultPanel.Dock = System.Windows.Forms.DockStyle.Fill;

this.ResultPanel.Location = new System.Drawing.Point(0, 188);

this.ResultPanel.Margin = new System.Windows.Forms.Padding(0);

this.ResultPanel.Name = "ResultPanel";

this.ResultPanel.Size = new System.Drawing.Size(838, 379);

this.ResultPanel.TabIndex = 3;

//

// ResultWebBrowser

//

this.ResultWebBrowser.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)

| System.Windows.Forms.AnchorStyles.Left)

| System.Windows.Forms.AnchorStyles.Right))));

this.ResultWebBrowser.Location = new System.Drawing.Point(0, 41);

this.ResultWebBrowser.Margin = new System.Windows.Forms.Padding(0);

this.ResultWebBrowser.MinimumSize = new System.Drawing.Size(23, 23);

this.ResultWebBrowser.Name = "ResultWebBrowser";

this.ResultWebBrowser.Size = new System.Drawing.Size(838, 338);

this.ResultWebBrowser.TabIndex = 3;

```

```

//

// ResultLabel

//

this.ResultLabel.AutoSize = true;

this.ResultLabel.Font = new System.Drawing.Font("Segoe UI", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.ResultLabel.Location = new System.Drawing.Point(20, 0);

this.ResultLabel.Name = "ResultLabel";

this.ResultLabel.Size = new System.Drawing.Size(79, 21);

this.ResultLabel.TabIndex = 0;

this.ResultLabel.Text = "Resultado";

//

// ClipboardButton

//

this.ClipboardButton.Image = global::Oraculo.Properties.Resources.clipboard;

this.ClipboardButton.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.ClipboardButton.Location = new System.Drawing.Point(10, 20);

this.ClipboardButton.Name = "ClipboardButton";

this.ClipboardButton.Size = new System.Drawing.Size(162, 30);

this.ClipboardButton.TabIndex = 2;

```

```
this.ClipboardButton.Text = "Copiar Texto (Ctrl + C)";

this.ClipboardButton.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

this.toolTip1.SetToolTip(this.ClipboardButton, "Clique para copiar o resultado do
teste para a área de trabalho (Ctrl + C). Você " +

"pode colar o resultado em outro aplicativo (Ctrl + V).");

this.ClipboardButton.UseVisualStyleBackColor = true;

this.ClipboardButton.Click += new
System.EventHandler(this.ClipboardButton_Click);

//

// TransformPanel

//

this.TransformPanel.AutoScroll = true;

this.TransformPanel.Controls.Add(this.ComboBoxTransformar2);

this.TransformPanel.Controls.Add(this.LabelTransformar2);

this.TransformPanel.Controls.Add(this.ComboBoxTransformar1);

this.TransformPanel.Controls.Add(this.LabelTransformar1);

this.TransformPanel.Controls.Add(this.DataGrid);

this.TransformPanel.Controls.Add(this.LabelPreviewDados);

this.TransformPanel.Controls.Add(this.RadioButtonDadosTransformados);

this.TransformPanel.Controls.Add(this.RadioButtonDadosDigitados);

this.TransformPanel.Controls.Add(this.LabelTransform);
```

```

this.TransformPanel.Dock = System.Windows.Forms.DockStyle.Fill;

this.TransformPanel.Location = new System.Drawing.Point(0, 188);

this.TransformPanel.Margin = new System.Windows.Forms.Padding(0);

this.TransformPanel.Name = "TransformPanel";

this.TransformPanel.Size = new System.Drawing.Size(838, 379);

this.TransformPanel.TabIndex = 0;

this.TransformPanel.Visible = false;

this.TransformPanel.VisibleChanged += new
System.EventHandler(this.TransformPanel_VisibleChanged);

//

// ComboBoxTransformar2

//

this.ComboBoxTransformar2.Enabled = false;

this.ComboBoxTransformar2.FormattingEnabled = true;

this.ComboBoxTransformar2.Location = new System.Drawing.Point(428, 103);

this.ComboBoxTransformar2.Name = "ComboBoxTransformar2";

this.ComboBoxTransformar2.Size = new System.Drawing.Size(268, 23);

this.ComboBoxTransformar2.TabIndex = 6;

this.ComboBoxTransformar2.SelectedIndexChanged += new
System.EventHandler(this.ComboBoxTransformar2_SelectedIndexChanged);

//

```

```
// LabelTransformar2

//

this.LabelTransformar2.AutoSize = true;

this.LabelTransformar2.Enabled = false;

this.LabelTransformar2.Location = new System.Drawing.Point(395, 107);

this.LabelTransformar2.Name = "LabelTransformar2";

this.LabelTransformar2.Size = new System.Drawing.Size(25, 15);

this.LabelTransformar2.TabIndex = 5;

this.LabelTransformar2.Text = "por";

//

// ComboBoxTransformar1

//

this.ComboBoxTransformar1.Enabled = false;

this.ComboBoxTransformar1.FormattingEnabled = true;

this.ComboBoxTransformar1.Location = new System.Drawing.Point(119, 103);

this.ComboBoxTransformar1.Name = "ComboBoxTransformar1";

this.ComboBoxTransformar1.Size = new System.Drawing.Size(268, 23);

this.ComboBoxTransformar1.TabIndex = 4;

this.ComboBoxTransformar1.SelectedIndexChanged += new
System.EventHandler(this.ComboBoxTransformar1_SelectedIndexChanged);
```



```

//

// LabelTransformar1

//

this.LabelTransformar1.AutoSize = true;

this.LabelTransformar1.Enabled = false;

this.LabelTransformar1.Location = new System.Drawing.Point(59, 107);

this.LabelTransformar1.Name = "LabelTransformar1";

this.LabelTransformar1.Size = new System.Drawing.Size(50, 15);

this.LabelTransformar1.TabIndex = 3;

this.LabelTransformar1.Text = "Agrupar";

//

// DataGrid

//

this.DataGrid.AllowUserToAddRows = false;

this.DataGrid.AllowUserToDeleteRows = false;

this.DataGrid.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right))));

```

```

        this.DataGrid.AutoSizeColumnsMode =
System.Windows.Forms.DataGridViewAutoSizeColumnsMode.DisplayedCells;

        this.DataGrid.BackgroundColor = System.Drawing.Color.Silver;

        this.DataGrid.BorderStyle = System.Windows.Forms.BorderStyle.None;

        this.DataGrid.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;

        this.DataGrid.Location = new System.Drawing.Point(27, 172);

        this.DataGrid.Margin = new System.Windows.Forms.Padding(20, 0, 20, 20);

        this.DataGrid.Name = "DataGrid";

        this.DataGrid.ReadOnly = true;

        this.DataGrid.RowHeadersWidth = 25;

        this.DataGrid.Size = new System.Drawing.Size(784, 181);

        this.DataGrid.TabIndex = 8;

        //

        // LabelPreviewDados

        //

        this.LabelPreviewDados.AutoSize = true;

        this.LabelPreviewDados.Location = new System.Drawing.Point(27, 149);

        this.LabelPreviewDados.Margin = new System.Windows.Forms.Padding(20, 20, 5, 5);

        this.LabelPreviewDados.Name = "LabelPreviewDados";

        this.LabelPreviewDados.Size = new System.Drawing.Size(95, 15);

```

```
this.LabelPreviewDados.TabIndex = 7;

this.LabelPreviewDados.Text = "Pré-visualização:";

//

// RadioButtonDadosTransformados

//

this.RadioButtonDadosTransformados.AutoSize = true;

this.RadioButtonDadosTransformados.Location = new System.Drawing.Point(29, 70);

this.RadioButtonDadosTransformados.Margin = new
System.Windows.Forms.Padding(20);

this.RadioButtonDadosTransformados.Name = "RadioButtonDadosTransformados";

this.RadioButtonDadosTransformados.Size = new System.Drawing.Size(138, 19);

this.RadioButtonDadosTransformados.TabIndex = 2;

this.RadioButtonDadosTransformados.Text = "Dados transformados";

this.RadioButtonDadosTransformados.UseVisualStyleBackColor = true;

this.RadioButtonDadosTransformados.CheckedChanged += new
System.EventHandler(this.RadioButtonDadosTransformados_CheckedChanged);

//

// RadioButtonDadosDigitados

//

this.RadioButtonDadosDigitados.AutoSize = true;

this.RadioButtonDadosDigitados.Checked = true;
```

```

this.RadioButtonDadosDigitados.Location = new System.Drawing.Point(29, 41);

this.RadioButtonDadosDigitados.Margin = new System.Windows.Forms.Padding(20);

this.RadioButtonDadosDigitados.Name = "RadioButtonDadosDigitados";

this.RadioButtonDadosDigitados.Size = new System.Drawing.Size(214, 19);

this.RadioButtonDadosDigitados.TabIndex = 1;

this.RadioButtonDadosDigitados.TabStop = true;

this.RadioButtonDadosDigitados.Text = "Dados da tabela, tal como digitados";

this.RadioButtonDadosDigitados.UseVisualStyleBackColor = true;

this.RadioButtonDadosDigitados.CheckedChanged += new
System.EventHandler(this.RadioButtonDadosDigitados_CheckedChanged);

//

// LabelTransform

//

this.LabelTransform.AutoSize = true;

this.LabelTransform.Font = new System.Drawing.Font("Segoe UI", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.LabelTransform.Location = new System.Drawing.Point(20, 0);

this.LabelTransform.Margin = new System.Windows.Forms.Padding(20);

this.LabelTransform.Name = "LabelTransform";

this.LabelTransform.Size = new System.Drawing.Size(191, 21);

this.LabelTransform.TabIndex = 0;

```

```
this.LabelTransform.Text = "Que dados deseja utilizar?";

//

// Analyze

//

this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);

this.Controls.Add(this.DataArrangePanel);

this.Controls.Add(this.VariablesPanel);

this.Controls.Add(this.TransformPanel);

this.Controls.Add(this.ResultPanel);

this.Name = "Analyze";

this.Size = new System.Drawing.Size(838, 636);

this.Title = "Analisar";

this.Controls.SetChildIndex(this.sectionHeader1, 0);

this.Controls.SetChildIndex(this.TitleLabel, 0);

this.Controls.SetChildIndex(this.ButtonPanel, 0);

this.Controls.SetChildIndex(this.ResultPanel, 0);

this.Controls.SetChildIndex(this.TransformPanel, 0);

this.Controls.SetChildIndex(this.VariablesPanel, 0);

this.Controls.SetChildIndex(this.DataArrangePanel, 0);

this.ButtonPanel.ResumeLayout(false);
```

```
        this.DataArrangePanel.ResumeLayout(false);

        this.DataArrangePanel.PerformLayout();

        ((System.ComponentModel.ISupportInitialize)(this.DataArrangeOnePictureBox2)).EndInit();

        ((System.ComponentModel.ISupportInitialize)(this.DataArrangeOnePictureBox)).EndInit();

        this.VariablesPanel.ResumeLayout(false);

        this.VariablesPanel.PerformLayout();

        this.PairedGroupBox.ResumeLayout(false);

        this.PairedGroupBox.PerformLayout();

        this.ResultPanel.ResumeLayout(false);

        this.ResultPanel.PerformLayout();

        this.TransformPanel.ResumeLayout(false);

        this.TransformPanel.PerformLayout();

        ((System.ComponentModel.ISupportInitialize)(this.DataGrid)).EndInit();

        this.ResumeLayout(false);

    }

#endregion
```

```
private System.Windows.Forms.Panel DataArrangePanel;

private System.Windows.Forms.Label DataArrangeLabel;

private System.Windows.Forms.Panel VariablesPanel;

private System.Windows.Forms.GroupBox PairedGroupBox;

private System.Windows.Forms.RadioButton PairedRadioButton;

private System.Windows.Forms.RadioButton NotPairedRadioButton;

private Global.MoveBetwenListboxesControl VariablesListboxesControl;

private System.Windows.Forms.Panel ResultPanel;

private System.Windows.Forms.Label ResultLabel;

private System.Windows.Forms.Button ClipboardButton;

private System.Windows.Forms.ToolTip toolTip1;

private Global.Info infoPareado;

private System.Windows.Forms.Panel TransformPanel;

private System.Windows.Forms.Label LabelPreviewDados;

private System.Windows.Forms.RadioButton RadioButtonDadosTransformados;

private System.Windows.Forms.RadioButton RadioButtonDadosDigitados;

private System.Windows.Forms.Label LabelTransform;

private System.Windows.Forms.Label LabelTransformar1;

private System.Windows.Forms.DataGridView DataGrid;

private System.Windows.Forms.ComboBox ComboBoxTransformar2;
```

```
private System.Windows.Forms.Label LabelTransformar2;

private System.Windows.Forms.ComboBox ComboBoxTransformar1;

private System.Windows.Forms.PictureBox DataArrangeOnePictureBox2;

private System.Windows.Forms.PictureBox DataArrangeOnePictureBox;

private System.Windows.Forms.Label DataArrangeLabel2;

private System.Windows.Forms.TextBox MeanTextBox;

private System.Windows.Forms.Label MeanLabel;

private System.Windows.Forms.WebBrowser ResultWebBrowser;

private System.Windows.Forms.Label label1;

}

}
```

Oraculo\Screens\CSVFileImportWizardControl.cs

```
?using Oraculo.Global;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;
```



```

using System.Drawing;

using System.Globalization;

using System.IO;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using ExcelDataReader;

namespace Oraculo.Telas
{

    public delegate void EventHandler(DataTable data);

    public partial class CSVFileImportWizardControl : Oraculo.Global.WizardControl
    {

        #region Properties

        public String FilePath { get; set; } = "";

        public string SelectedSheet { get; set; } = "";

        public bool ShowColumnTitles { get; set; } = false;

        public DataTable Data { get; set; }

```

```
#endregion
```

```
#region Events
```

```
public event EventHandler AfterImportData;
```

```
#endregion
```

```
#region Métodos
```

```
public CSVFileImportWizardControl()
```

```
{
```

```
    InitializeComponent();
```

```
    SelectFilePanel.AllowDrop = true;
```

```
    SelectFilePanel.DragDrop += new DragEventHandler(SelectFileTextBox_DragDrop);
```

```
    SelectFilePanel.DragEnter += new  
DragEventHandler(SelectFileTextBox_DragEnter);
```

```
    SelectFilePanel.DragOver += new DragEventHandler(SelectFileTextBox_DragOver);
```

```

    TotalSteps = 2;

    ShowStep(1);
}

//importa os dados

private Boolean ImportData()
{
    Data = GetDataTableFromCSV(FilePath, ShowColumnTitles);

    Data = ConvertDataTableColumnTypes(Data);

    return (Data.Rows.Count > 0);
}

//converte as colunas da datatable conforme os tipos de dados

private DataTable ConvertDataTableColumnTypes(DataTable dt)
{
    //copia a estrutura da tabela para uma tabela clonada

    DataTable newdt = dt.Clone();

```

//percorre para cada coluna da tabela cada linha e para cada valor tenta obter o tipo e atribuí-lo como tipo da coluna

```
foreach (DataColumn col in dt.Columns)

{

    Type atual = null;

    foreach (DataRow row in dt.Rows)

    {

        newdt.Columns[col.ColumnName].DataType = GetCellType(row[col], atual);

        atual = newdt.Columns[col.ColumnName].DataType;

    }

}

foreach (DataRow row in dt.Rows)

{

    newdt.ImportRow(row);

}

return newdt;

}
```

//obtem os dados do csv para um dataset

```
private DataTable GetDataTableFromCSV(string path, bool isFirstRowHeader)

{

    using (var stream = File.Open(path.TrimEnd(), FileMode.Open, FileAccess.Read))

    {

        using (var reader = ExcelReaderFactory.CreateCsvReader(stream))

        {

            var result = reader.AsDataSet(new ExcelDataSetConfiguration()

            {

                UseColumnDataType = true,

                ConfigureDataTable = (tableReader) => new ExcelDataTableConfiguration()

                {

                    EmptyColumnNamePrefix = "Coluna",

                    UseHeaderRow = isFirstRowHeader,

                    //ReadHeaderRow = (rowReader) => { rowReader.Read(); },

                    //FilterRow = (rowReader) => { return true; },

                    //FilterColumn = (rowReader, columnIndex) => { return true; }

                }

            });

        }

    }

};
```

```

        foreach (DataTable dt in result.Tables)

            return dt;

    }

}

return null;

}

//retorna true se o dado é um campo numérico

public bool IsNumeric(DataColumn col)

{

    if (col == null) return false;

    //Make this const

    var numericTypes = new[] { typeof(Byte), typeof(Decimal), typeof(Double),

    typeof(Int16), typeof(Int32), typeof(Int64), typeof(SByte),

    typeof(Single), typeof(UInt16), typeof(UInt32), typeof(UInt64)};

    return numericTypes.Contains(col.DataType);

}

```

//obtem tipo de dados de uma célula da datatable conforme o seu conteúdo

```
public Type GetCellType(object value, Type lasttype = null)
```

```
{
```

```
    if (lasttype == typeof(string))
```

```
    {
```

```
        return typeof(String);
```

```
    }
```

```
    else
```

```
    {
```

```
        Type theType = lasttype;
```

```
        string valueStr = value.ToString();
```

```
        DateTime parsedDateTime;
```

```
        Int32 parsedInt32;
```

```
        double parsedDouble;
```

```
        bool parsedBool;
```

```
        // We try to parse the value using all standard datetime formats
```

```
        // (custom formats are not supported unless you want to modify the
```

```
        // DateFormatStrings object.).
```

```

        if (DateTime.TryParseExact(valueStr,
CultureInfo.CurrentUICulture.DateTimeFormat.ShortDatePattern,
CultureInfo.CurrentUICulture, DateTimeStyles.None, out parsedDateTime))
    {
        theType = typeof(DateTime);
    }

    else if (DateTime.TryParseExact(valueStr,
CultureInfo.CurrentUICulture.DateTimeFormat.LongDatePattern,
CultureInfo.CurrentUICulture, DateTimeStyles.None, out parsedDateTime))
    {
        theType = typeof(DateTime);
    }

    else if (DateTime.TryParseExact(valueStr,
CultureInfo.CurrentUICulture.DateTimeFormat.ShortTimePattern,
CultureInfo.CurrentUICulture, DateTimeStyles.None, out parsedDateTime))
    {
        theType = typeof(DateTime);
    }

    else if (Int32.TryParse(valueStr, out parsedInt32))
    {
        // we only want to change the type if it's not already a double
        if (theType == null || (theType != null && theType.Name != "Double"))

```



```
    {  
        theType = typeof(Int32);  
    }  
}  
else if (double.TryParse(valueStr, out parsedDouble))  
  
    {  
        theType = typeof(Double);  
    }  
else if (bool.TryParse(valueStr, out parsedBool))  
  
    {  
        theType = typeof(Boolean);  
    }  
else  
  
    {  
        theType = typeof(String);  
    }  
  
return theType;  
}
```

```

}

public override bool ValidateStep()

{

    //preenche o panel conforme a etapa

    switch (Step)

    {

        case 1:

            //solicita ao usuário o arquivo que deseja importar

            if (FilePath == "")

            {

                MessageBox.Show(this, "Você precisa informar qual é o arquivo que deseja importar.", "Erro ao importar", MessageBoxButtons.OK, MessageBoxIcon.Error);

                SelectFileButton.Focus();

                return false;

            }

            else

            {

                if (Path.GetExtension(FilePath) != ".csv" && Path.GetExtension(FilePath) != ".txt")

                {

```

```
        MessageBox.Show("Você não informou um arquivo texto (*.csv; *.txt).  
Informe um novo arquivo e tente novamente.", "Erro ao importar arquivo texto",  
        MessageBoxButtons.OK, MessageBoxIcon.Error);
```

```
        SelectFileButton.Focus();
```

```
        return false;
```

```
    }
```

```
        SelectFileTextBox.Text = FilePath;
```

```
    }
```

```
        //exibe mensagem para confirmar se a primeira linha tem títulos dos campos
```

```
        ShowColumnTitles = (MessageBox.Show(this, "A primeira linha contém os  
títulos dos campos?", "Títulos das colunas", MessageBoxButtons.YesNo,  
        MessageBoxIcon.Question) == DialogResult.Yes);
```

```
        //exibe tela com mensagem para usuário aguardar a importação
```

```
        SelectFilePanel.Visible = false;
```

```
        ImportingPanel.Visible = true;
```

```
        //importa os dados
```

```
        if (!ImportData())
```

```

    {

        MessageBox.Show(this, "Não foi possível importar os dados ou o arquivo está
vazio. Verifique o arquivo informado e tente novamente.", "Erro ao importar",
MessageBoxButtons.OK, MessageBoxIcon.Error);

        return false;

    }

else

{

    //exibe os dados importados para o usuário confirmar se está tudo OK.

    DataSubtitleLabel.Text = String.Format("Foram localizadas {0} coluna(s) e
{1} linha(s). Confira se os dados foram importados corretamente:", Data.Columns.Count,
Data.Rows.Count);

    DataGrid.DataSource = Data;

}

break;

case 2:

    if (MessageBox.Show(this, "Os dados importados estão corretos?", "Confirmação
dos dados", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No)

    {

        Step = 1;

        return false;

    }

```

```
        break;

    }

    return true;

}

public override void ShowStep()

{

    SelectFilePanel.Visible = (Step == 1);

    ImportingPanel.Visible = false;

    DataPanel.Visible = (Step == 2);

    base.ShowStep();

}

public override bool CloseWizard(bool cancel = false)

{

    if (cancel)

    {

        ShowStep(1);

    }

    else
```

```
{  
  
    AfterImportData.Invoke(Data);  
  
}  
  
return base.CloseWizard(cancel);  
  
}  
  
#endregion  
  
private void SelectFileButton_Click(object sender, EventArgs e)  
  
{  
  
    //abre a caixa de diálogo de sistema para localizar uma arquivo csv  
  
    OpenFileDialog tela = new OpenFileDialog();  
  
    if (FilePath != "")  
  
    {  
  
        tela.FileName = FilePath;  
  
    }  
  
    else  
  
    {
```

```

        tela.InitialDirectory =
System.Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);

    }

    tela.Filter = "Arquivos texto (*.txt)|*.txt|Arquivos CSV (*.csv)|*.csv|Todos os
arquivos|*.*";

    tela.DefaultExt = "csv";

    tela.CheckFileExists = true;

    tela.CheckPathExists = true;

    if (tela.ShowDialog() == DialogResult.OK)

    {

        FilePath = tela.FileName;

        SelectFileTextBox.Text = FilePath;

    }

}

private void SelectFileTextBox_DragDrop(object sender, DragEventArgs e)

{

    if (e.Data.GetDataPresent(DataFormats.FileDrop, false) == true)

    {

        e.Effect = DragDropEffects.All;

```

```

    }
}

private void SelectFileTextBox_DragEnter(object sender, DragEventArgs e)
{
    string[] file = (string[])e.Data.GetData(DataFormats.FileDrop);

    if (file != null && file.Length != 0)
    {
        SelectFileTextBox.Text = file[0];

        FilePath = file[0];
    }
}

```

```

private void SelectFileTextBox_DragOver(object sender, DragEventArgs e)
{
    if (e.Data.GetDataPresent(DataFormats.FileDrop))
    {
        e.Effect = DragDropEffects.Copy;
    }

    else

```



```

    {
        e.Effect = DragDropEffects.None;
    }
}

private void SelectFileTextBox_TextChanged(object sender, EventArgs e)
{
    if (SelectFileTextBox.Text != "Clique no botão localizar ou arraste o arquivo aqui!")
        SelectFileTextBox.ForeColor = System.Drawing.SystemColors.WindowText;
    else
        SelectFileTextBox.ForeColor = Color.Gray;
}
}
}
}

```

Oraculo\Screens\CSVFileImportWizardControl.Designer.cs

```

?namespace Oraculo.Telas
{
    partial class CSVFileImportWizardControl

```

```

{

    /// <summary>

    /// Required designer variable.

    /// </summary>

    private System.ComponentModel.IContainer components = null;

    /// <summary>

    /// Clean up any resources being used.

    /// </summary>

    /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>

    protected override void Dispose(bool disposing)
    {

        if (disposing && (components != null))
        {

            components.Dispose();

        }

        base.Dispose(disposing);

    }

```

```
#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

    this.ImportingPanel = new System.Windows.Forms.Panel();

    this.ImportingLabel = new System.Windows.Forms.Label();

    this.DataPanel = new System.Windows.Forms.Panel();

    this.DataSubtitleLabel = new System.Windows.Forms.Label();

    this.DataTitleLabel = new System.Windows.Forms.Label();

    this.DataGrid = new System.Windows.Forms.DataGridView();

    this.SelectFilePanel = new System.Windows.Forms.Panel();

    this.SelectFileButton = new System.Windows.Forms.Button();

    this.SelectFileTextBox = new System.Windows.Forms.TextBox();

    this.SelectFileLabel = new System.Windows.Forms.Label();

    this.ButtonPanel.SuspendLayout();

    this.ImportingPanel.SuspendLayout();
```

```
this.DataPanel.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.DataGrid)).BeginInit();

this.SelectFilePanel.SuspendLayout();

this.SuspendLayout();

//

// sectionHeader1

//

this.sectionHeader1.BreadCrumbText = "Início > Dados > Importar";

//

// TitleLabel

//

this.TitleLabel.Text = "Importar arquivo texto (*.txt; *.csv)";

//

// NextButton

//

this.NextButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;

//

// PreviousButton

//

this.PreviousButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;
```

```

//

// ImportingPanel

//

this.ImportingPanel.Controls.Add(this.ImportingLabel);

this.ImportingPanel.Dock = System.Windows.Forms.DockStyle.Fill;

this.ImportingPanel.Location = new System.Drawing.Point(0, 188);

this.ImportingPanel.Name = "ImportingPanel";

this.ImportingPanel.Size = new System.Drawing.Size(640, 223);

this.ImportingPanel.TabIndex = 4;

//

// ImportingLabel

//

this.ImportingLabel.Anchor = System.Windows.Forms.AnchorStyles.None;

this.ImportingLabel.AutoSize = true;

this.ImportingLabel.Font = new System.Drawing.Font("Segoe UI", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.ImportingLabel.Location = new System.Drawing.Point(205, 101);

this.ImportingLabel.Name = "ImportingLabel";

this.ImportingLabel.Size = new System.Drawing.Size(230, 21);

this.ImportingLabel.TabIndex = 0;

```

```
this.ImportingLabel.Text = "Aguarde... importando os dados";

//

// DataPanel

//

this.DataPanel.Controls.Add(this.DataSubtitleLabel);

this.DataPanel.Controls.Add(this.DataTitleLabel);

this.DataPanel.Controls.Add(this.DataGrid);

this.DataPanel.Dock = System.Windows.Forms.DockStyle.Fill;

this.DataPanel.Location = new System.Drawing.Point(0, 188);

this.DataPanel.Margin = new System.Windows.Forms.Padding(0);

this.DataPanel.Name = "DataPanel";

this.DataPanel.Size = new System.Drawing.Size(640, 223);

this.DataPanel.TabIndex = 8;

//

// DataSubtitleLabel

//

this.DataSubtitleLabel.AutoSize = true;

this.DataSubtitleLabel.Location = new System.Drawing.Point(21, 26);

this.DataSubtitleLabel.Margin = new System.Windows.Forms.Padding(20, 0, 0, 20);

this.DataSubtitleLabel.Name = "DataSubtitleLabel";
```

```

this.DataSubtitleLabel.Size = new System.Drawing.Size(507, 15);

this.DataSubtitleLabel.TabIndex = 7;

this.DataSubtitleLabel.Text = "Foram localizados {0} campos e {1} casos. Confira se
os dados foram importados co" +

"rretamente:";

//

// DataTitleLabel

//

this.DataTitleLabel.AutoSize = true;

this.DataTitleLabel.Font = new System.Drawing.Font("Segoe UI", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.DataTitleLabel.Location = new System.Drawing.Point(19, 0);

this.DataTitleLabel.Margin = new System.Windows.Forms.Padding(20, 0, 20, 5);

this.DataTitleLabel.Name = "DataTitleLabel";

this.DataTitleLabel.Size = new System.Drawing.Size(164, 21);

this.DataTitleLabel.TabIndex = 6;

this.DataTitleLabel.Text = "Importação concluída!";

//

// DataGrid

//

this.DataGrid.AllowUserToAddRows = false;

```

```

this.DataGrid.AllowUserToDeleteRows = false;

this.DataGrid.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)

| System.Windows.Forms.AnchorStyles.Left)

| System.Windows.Forms.AnchorStyles.Right))));

this.DataGrid.AutoSizeColumnsMode =
System.Windows.Forms.DataGridViewAutoSizeColumnsMode.DisplayedCells;

this.DataGrid.BackgroundColor = System.Drawing.Color.Silver;

this.DataGrid.BorderStyle = System.Windows.Forms.BorderStyle.None;

this.DataGrid.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;

this.DataGrid.Location = new System.Drawing.Point(24, 61);

this.DataGrid.Margin = new System.Windows.Forms.Padding(20, 0, 20, 20);

this.DataGrid.Name = "DataGrid";

this.DataGrid.ReadOnly = true;

this.DataGrid.RowHeadersWidth = 25;

this.DataGrid.Size = new System.Drawing.Size(596, 136);

this.DataGrid.TabIndex = 2;

//

// SelectFilePanel

```



```

//

this.SelectFilePanel.Controls.Add(this.SelectFileButton);

this.SelectFilePanel.Controls.Add(this.SelectFileTextBox);

this.SelectFilePanel.Controls.Add(this.SelectFileLabel);

this.SelectFilePanel.Dock = System.Windows.Forms.DockStyle.Fill;

this.SelectFilePanel.Location = new System.Drawing.Point(0, 188);

this.SelectFilePanel.Margin = new System.Windows.Forms.Padding(0);

this.SelectFilePanel.Name = "SelectFilePanel";

this.SelectFilePanel.Size = new System.Drawing.Size(640, 223);

this.SelectFilePanel.TabIndex = 9;

//

// SelectFileButton

//

this.SelectFileButton.BackColor = System.Drawing.Color.Gainsboro;

this.SelectFileButton.FlatAppearance.BorderColor =
System.Drawing.Color.Gainsboro;

this.SelectFileButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.SelectFileButton.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.SelectFileButton.Location = new System.Drawing.Point(408, 21);

this.SelectFileButton.Margin = new System.Windows.Forms.Padding(20);

```

```
this.SelectFileButton.Name = "SelectFileButton";

this.SelectFileButton.Size = new System.Drawing.Size(99, 28);

this.SelectFileButton.TabIndex = 2;

this.SelectFileButton.Text = "Localizar...";

this.SelectFileButton.UseVisualStyleBackColor = false;

this.SelectFileButton.Click += new
System.EventHandler(this.SelectFileButton_Click);

//

// SelectFileTextBox

//

this.SelectFileTextBox.AllowDrop = true;

this.SelectFileTextBox.BackColor = System.Drawing.SystemColors.ButtonFace;

this.SelectFileTextBox.ForeColor = System.Drawing.Color.Gray;

this.SelectFileTextBox.Location = new System.Drawing.Point(24, 25);

this.SelectFileTextBox.Margin = new System.Windows.Forms.Padding(20, 0, 20, 20);

this.SelectFileTextBox.Name = "SelectFileTextBox";

this.SelectFileTextBox.ReadOnly = true;

this.SelectFileTextBox.Size = new System.Drawing.Size(375, 23);

this.SelectFileTextBox.TabIndex = 1;

this.SelectFileTextBox.Text = "Clique no botão localizar ou arraste o arquivo aqui!";
```

```

        this.SelectFileTextBox.TextChanged += new
System.EventHandler(this.SelectFileTextBox_TextChanged);

        this.SelectFileTextBox.DragDrop += new
System.Windows.Forms.DragEventHandler(this.SelectFileTextBox_DragDrop);

        this.SelectFileTextBox.DragEnter += new
System.Windows.Forms.DragEventHandler(this.SelectFileTextBox_DragEnter);

        this.SelectFileTextBox.DragOver += new
System.Windows.Forms.DragEventHandler(this.SelectFileTextBox_DragOver);

//

// SelectFileLabel

//

this.SelectFileLabel.AutoSize = true;

this.SelectFileLabel.Location = new System.Drawing.Point(20, 0);

this.SelectFileLabel.Margin = new System.Windows.Forms.Padding(20, 0, 20, 5);

this.SelectFileLabel.Name = "SelectFileLabel";

this.SelectFileLabel.Size = new System.Drawing.Size(285, 15);

this.SelectFileLabel.TabIndex = 0;

this.SelectFileLabel.Text = "Selecione o arquivo que contém os dados desejados:";

//

// CSVFileImportWizardControl

//

```

```
this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);

this.Controls.Add(this.DataPanel);

this.Controls.Add(this.SelectFilePanel);

this.Controls.Add(this.ImportingPanel);

this.Name = "CSVFileImportWizardControl";

this.Title = "Importar arquivo texto (*.txt; *.csv)";

this.Controls.SetChildIndex(this.sectionHeader1, 0);

this.Controls.SetChildIndex(this.TitleLabel, 0);

this.Controls.SetChildIndex(this.ButtonPanel, 0);

this.Controls.SetChildIndex(this.ImportingPanel, 0);

this.Controls.SetChildIndex(this.SelectFilePanel, 0);

this.Controls.SetChildIndex(this.DataPanel, 0);

this.ButtonPanel.ResumeLayout(false);

this.ImportingPanel.ResumeLayout(false);

this.ImportingPanel.PerformLayout();

this.DataPanel.ResumeLayout(false);

this.DataPanel.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.DataGrid)).EndInit();

this.SelectFilePanel.ResumeLayout(false);

this.SelectFilePanel.PerformLayout();
```

```
        this.ResumeLayout(false);

    }

#endregion

private System.Windows.Forms.Panel ImportingPanel;

private System.Windows.Forms.Label ImportingLabel;

private System.Windows.Forms.Panel DataPanel;

private System.Windows.Forms.Label DataSubtitleLabel;

private System.Windows.Forms.Label DataTitleLabel;

private System.Windows.Forms.DataGridView DataGrid;

private System.Windows.Forms.Panel SelectFilePanel;

private System.Windows.Forms.Button SelectFileButton;

private System.Windows.Forms.TextBox SelectFileTextBox;

private System.Windows.Forms.Label SelectFileLabel;

    }

}
```

Oraculo\Screens\DataView.cs

```
?using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Oraculo.Telas
{
    public partial class DataView : UserControl
    {
        #region Properties

        public DataTable Data
        {
            get
            {

```

```
        return (DataTable)this.DataGrid.DataSource;
    }

    set
    {
        this.DataGrid.DataSource = value;

        RefreshTotal();

        //RefreshDataGrid();
    }
}
```

```
public Boolean HasData
```

```
{
    get
    {
        return (this.DataGrid.DataSource != null);
    }
}
```

```
private int _SelectedColumn;
```

```
public int SelectedColumn
```

```
{  
  
    get  
  
    {  
  
        return _SelectedColumn;  
  
    }  
  
    set  
  
    {  
  
        _SelectedColumn = value;  
  
  
        //habilita/desabilita o botão para remover coluna  
  
        RemoveColumnButton.Enabled = (value > -1);  
  
    }  
  
}
```

#endregion

#region Methods

public DataView()

```
{
```



```
InitializeComponent();

//cria tooltip para os botões

ToolTip tt = new ToolTip();

tt.AutoPopDelay = 5000;

tt.InitialDelay = 1000;

tt.ReshowDelay = 500;

tt.ShowAlways = false;

//atribui a tooltip para os botões

tt.SetToolTip(AddColumnButton, "Adicionar Coluna");

tt.SetToolTip(RemoveColumnButton, "Remover Coluna");

tt.SetToolTip(RemoveLinesButton, "Remover linha(s)");

}

private void RefreshTotal()

{

    if (!HasData)

    {
```

```

        TotalLabel.Text = "Nenhum dado.";

    }

    else

    {

        TotalLabel.Text = String.Format("Total: {1} linhas e {0} colunas.",
Data.Columns.Count, Data.Rows.Count);

    }

    AnalyzeButton.Enabled = (HasData);

}

//mostra a caixa de texto para editar o cabeçalho da coluna no grid

private void ShowTextBoxHeader(int index)

{

    //guarda o texto atual e o index da coluna na caixa de texto

    textBoxHeader.Text = DataGrid.Columns[index].HeaderText;

    this.SelectedColumn = index;

    //posiciona a caixa de texto sobre o header do datagridview

    textBoxHeader.Top = DataGrid.Top + DataGrid.Margin.Top - 1;

```

```

int esquerda = DataGrid.Left;

for (int i = 0; i < index; i++)

    esquerda += DataGrid.Columns[i].Width;

textBoxHeader.Left = esquerda + DataGrid.RowHeadersWidth;

textBoxHeader.Width = DataGrid.Columns[index].Width + DataGrid.Margin.Left - 2;

textBoxHeader.Height = DataGrid.Columns[index].HeaderCell.Size.Height;

//desabilita o datagridview

DataGrid.Enabled = false;

//exibe a caixa de texto e dá foco para ela

textBoxHeader.Visible = true;

textBoxHeader.Focus();

}

//fecha a caixa de texto para editar o cabeçalho da coluna do grid e atualiza o dataset

private void ChangeColumnHeaderText()

{

    //se o usuário preencheu algo, troca o texto

```

```

        if (textBoxHeader.Text != "" && textBoxHeader.Text !=
DataGridView.Columns[this.SelectedColumn].HeaderText)

    {

        DataGridView.Columns[this.SelectedColumn].HeaderText = textBoxHeader.Text;

        Data.Columns[this.SelectedColumn].ColumnName = textBoxHeader.Text;

    }

//reabilita a edição no dataview

DataGridView.Enabled = true;

//oculta a caixa de texto

textBoxHeader.Visible = false;

//seleciona a primeira célula da coluna

UnselectDataGridColumn(SelectedColumn);

}

//adiciona um anova coluna no dataset e no datagrid

private void AddColumn()

{

```

```

AddColumnForm frm = new AddColumnForm();

frm.ShowInsertMode();

if (frm.HasModified)
{
    if (!HasData) { Data = new DataTable(); }

    //copia a estrutura da tabela para uma tabela clonada

    DataTable newdt = Data.Clone();

    newdt.Columns.Add(frm.GetColumnName(), frm.GetSelectedDataType());

    foreach (DataRow row in Data.Rows)

        newdt.ImportRow(row);

    Data = newdt;

    //deseleciona a coluna e move o foco para a nova coluna

    UnselectDataGridColumn(SelectedColumn);

    DataGrid.CurrentCell = DataGrid.Rows[0].Cells[DataGrid.ColumnCount - 1];

    //atualiza o total

    RefreshTotal();

```

```

    }
}

//remove a coluna do dataset e do datagrid

private void RemoveColumn()
{
    if (!HasData || SelectedColumn == -1) { return; }

    string coluna = DataGrid.Columns[SelectedColumn].HeaderText;

    if (MessageBox.Show(this, String.Format("Você deseja realmente excluir a coluna
'{0}'?", coluna), "Confirmação da exclusão", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
    {
        //copia a estrutura da tabela para uma tabela clonada

        DataGridViewCell currentCell = DataGrid.CurrentCell;

        DataTable newdt = Data.Clone();

        newdt.Columns.RemoveAt(SelectedColumn);

        foreach (DataRow row in Data.Rows)

            if (row != Data.Rows[SelectedColumn])

                newdt.ImportRow(row);

        Data = newdt;
    }
}

```

```

        RefreshTotal();

    }

}

//remove a linha do dataset e do datagrid

private void RemoveLines()

{

    if (DataGrid.SelectedRows.Count == 0) return;

    string mensagem;

    if (DataGrid.SelectedRows.Count == 1)

        mensagem = "Você deseja realmente excluir a linha selecionada?";

    else

        mensagem = "Você deseja realmente excluir as linhas selecionadas?";

    if (MessageBox.Show(this, mensagem, "Confirmação da exclusão",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

    {

        DataGrid.SuspendLayout();

```

```

//percorre cada item do datagrid e o remove na tabela clonada

foreach (DataGridViewRow item in DataGrid.SelectedRows)

{

    if (!item.IsNewRow)

    {

        Data.Rows.RemoveAt(item.Index);

    }

}

DataGrid.ResumeLayout();

}

}

```

//faz com que a coluna inteira seja selecionada, caso não esteja

```
private void SelectDataGridColumn(int index)
```

```

{

    if (DataGrid.RowCount > 0)

    {

        //seleciona todas as células da coluna

```



```

    DataGrid.SuspendLayout();

    DataGrid.ClearSelection();

    for (int r = 0; r < DataGrid.RowCount; r++)

        DataGrid[index, r].Selected = true;

    DataGrid.PerformLayout();

    //atualiza a coluna selecionada

    SelectedColumn = index;

}

}

//tira a seleção da coluna toda e seleciona a primeira célula de uma coluna

private void UnselectDataGridColumn(int index)

{

    if (index == -1) return;

    if (DataGrid.RowCount > 0)

    {

```

```
//retira a seleção das células

DataGrid.SuspendLayout();

DataGrid.ClearSelection();

DataGrid[index, 0].Selected = true;

DataGrid.PerformLayout();

//retira a seleção da coluna

SelectedColumn = -1;

}

}

#endregion

#region Events

public event EventHandler AfterClickAnalyze;

#endregion
```

```
private void AddColumnButton_Click(object sender, EventArgs e)
```

```
{
```

```
    AddColumn();
```

```
}
```

```
private void RemoveColumnButton_Click(object sender, EventArgs e)
```

```
{
```

```
    RemoveColumn();
```

```
}
```

```
private void DataGrid_ColumnRemoved(object sender,  
DataGridViewColumnEventArgs e)
```

```
{
```

```
    RefreshTotal();
```

```
}
```

```
private void DataGrid_ColumnAdded(object sender, DataGridViewColumnEventArgs e)
```

```
{
```

```
    //desabilita a ordenação
```

```
e.Column.SortMode = DataGridViewColumnSortMode.NotSortable;
```

```
//atualiza o total
```

```
RefreshTotal();
```

```
}
```

```
private void DataGrid_RowsAdded(object sender, DataGridViewRowsAddedEventArgs  
e)
```

```
{
```

```
RefreshTotal();
```

```
}
```

```
private void DataGrid_RowsRemoved(object sender,  
DataGridViewRowsRemovedEventArgs e)
```

```
{
```

```
RefreshTotal();
```

```
}
```

```
private void AnalyzeButton_Click(object sender, EventArgs e)
```

```
{
```

```
if (HasData) { AfterClickAnalyze.Invoke(Data); }
```

```
}
```

```
private void DataGrid_ColumnHeaderMouseDoubleClick(object sender,  
DataGridViewCellEventArgs e)
```

```
{
```

```
    //seleciona a coluna toda
```

```
    SelectDataGridColumn(e.ColumnIndex);
```

```
    //exibe a caixa de texto
```

```
    ShowTextBoxHeader(e.ColumnIndex);
```

```
}
```

```
private void textBoxHeader_Leave(object sender, EventArgs e)
```

```
{
```

```
    //se perder o foco, atualiza o grid
```

```
    ChangeColumnHeaderText();
```

```
}
```

```
private void textBoxHeader_KeyPress(object sender, KeyPressEventArgs e)
```

```
{
```

```

//se pressionou ENTER ou ESC, atualiza o grid

if ((e.KeyChar == (char)Keys.Return) || (e.KeyChar == (char)Keys.Escape))

    ChangeColumnHeaderText();

}

private void adicionarColunaToolStripMenuItem_Click(object sender, EventArgs e)

{

    AddColumn();

}

private void excluirColunaToolStripMenuItem_Click(object sender, EventArgs e)

{

    RemoveColumn();

}

private void renomearToolStripMenuItem_Click(object sender, EventArgs e)

{

    //seleciona a coluna toda

    SelectDataGridColumn(this.SelectedColumn);

```

```

//exibe a caixa de texto

ShowTextBoxHeader(this.SelectedColumn);

}

private void DataGrid_CellMouseDown(object sender,
DataGridViewCellMouseEventArgs e)

{

//não faz nada se clicou na célula -1 -1 do cabeçalho

if (e.RowIndex == -1 && e.ColumnIndex == -1 && e.Button ==
System.Windows.Forms.MouseButtons.Right)

return;

//faz com que a coluna inteira seja selecionada, caso não esteja

if (e.RowIndex == -1 && e.ColumnIndex > -1)

{

SelectDataGridColumn(e.ColumnIndex);

return;

}

//se clicou em uma célula qualquer que não seja a do cabeçalho, não há coluna
selecionada mais

```

```

if (e.RowIndex > -1 && e.Button == System.Windows.Forms.MouseButtons.Left)

{

    SelectedColumn = -1;

}

/*

//faz com que a célula clicada se torne a célula selecionada

if (e.ColumnIndex != -1 && e.RowIndex != -1 && e.Button ==
System.Windows.Forms.MouseButtons.Right)

{

    DataGridViewCell c = (sender as DataGridView)[e.ColumnIndex, e.RowIndex];

    if (!c.Selected)

    {

        c.DataGridView.ClearSelection();

        c.DataGridView.CurrentCell = c;

        c.Selected = true;

    }

}

*/

}

```



```

private void DataGrid_KeyDown(object sender, KeyEventArgs e)
{
    //permite que os atalhos de teclado acionem o menu de contexto
    if ((e.KeyCode == Keys.F10 && e.Shift) || e.KeyCode == Keys.Apps)
    {
        e.SuppressKeyPress = true;

        DataGridViewCell currentCell = (sender as DataGridView).CurrentCell;

        if (currentCell != null)
        {
            ContextMenuStrip cms = currentCell.ContextMenuStrip;

            if (cms != null)
            {
                Rectangle r =
currentCell.DataGridView.GetCellDisplayRectangle(currentCell.ColumnIndex,
currentCell.RowIndex, false);

                Point p = new Point(r.X + r.Width, r.Y + r.Height);

                cms.Show(currentCell.DataGridView, p);
            }
        }
    }
}

```

```

//se pressionar delete em uma coluna selecionada, remove

if (e.KeyCode == Keys.Delete && this.SelectedColumn > -1)

{

    e.SuppressKeyPress = true;

    RemoveColumn();

}

//se pressionar delete em uma (ou mais) linhas selecionadas, remove

if (e.KeyCode == Keys.Delete && DataGridView.SelectedRows.Count > 0)

{

    e.SuppressKeyPress = true;

    RemoveLines();

}

}

private void DataGridView_CellContextMenuStripNeeded(object sender,
DataGridViewCellContextMenuStripNeededEventArgs e)

{

    //exibe o menu de contexto para o cabeçalho se a célula clicada é a do cabeçalho

```

```

if (e.RowIndex == -1 && e.ColumnIndex > -1)

{

    this.SelectedColumn = e.ColumnIndex;

    DataGrid.ContextMenuStrip = contextMenuColuna;

}

//exibe o menu de contexto para a linha selecionada

else if (e.RowIndex > -1 && e.ColumnIndex == -1)

{

    DataGrid.ContextMenuStrip = contextMenuLinha;

}

else

{

    this.SelectedColumn = -1;

    DataGrid.ContextMenuStrip = null;

    DataGrid.ContextMenuStrip = null;

}

}

private void DataGrid_DataError(object sender, DataGridViewDataErrorEventArgs e)

{

```

```

Type t = Data.Columns[e.ColumnIndex].DataType;

if (t == typeof(int) || t == typeof(decimal) || t == typeof(double))

{

    MessageBox.Show(this, "Você tentou informar um número inválido. Informe um
número válido e tente novamente", "Dados inválidos", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);

}

else if (t == typeof(DateTime))

{

    MessageBox.Show(this, "Você tentou informar uma data inválida. Informe uma
data válida e tente novamente", "Dados inválidos", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);

}

else

{

    MessageBox.Show(this, "Você tentou informar um valor inválido. Informe uma
valor válido e tente novamente", "Dados inválidos", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);

}

}

```

```
private void ExcluirLinhaToolStripMenuItem_Click(object sender, EventArgs e)

{

    RemoveLines();

}
```

```
private void RemoveLineButton_Click(object sender, EventArgs e)

{

    RemoveLines();

}
```

```
private void DataGrid_SelectionChanged(object sender, EventArgs e)

{

    //habilita o botão para remover linhas se pelo menos uma foi selecionada

    RemoveLinesButton.Enabled = (DataGrid.SelectedRows.Count > 0);

}
```

```
private void DataView_SizeChanged(object sender, EventArgs e)

{

    //atualiza o datagrid

    //RefreshDataGrid();

}
```

```

    }

    private void DataGrid_ColumnDisplayIndexChanged(object sender,
DataGridViewColumnEventArgs e)
    {
        this.DataGrid.Refresh();
    }

    public void RefreshDataGrid()
    {
        this.Refresh();
    }
}
}
}

```

Oraculo\Screens\DataView.Designer.cs

```
?namespace Oraculo.Telas
```

```

{
    partial class DataView

```

```

{

    /// <summary>

    /// Variável de designer necessária.

    /// </summary>

    private System.ComponentModel.IContainer components = null;

    /// <summary>

    /// Limpar os recursos que estão sendo usados.

    /// </summary>

    /// <param name="disposing">true se for necessário descartar os recursos gerenciados;
caso contrário, false.</param>

    protected override void Dispose(bool disposing)
    {

        if (disposing && (components != null))
        {

            components.Dispose();

        }

        base.Dispose(disposing);

    }

```

```
#region Código gerado pelo Designer de Componentes
```

```
/// <summary>
```

```
/// Método necessário para suporte ao Designer - não modifique
```

```
/// o conteúdo deste método com o editor de código.
```

```
/// </summary>
```

```
private void InitializeComponent()
```

```
{
```

```
    this.components = new System.ComponentModel.Container();
```

```
    this.textBoxHeader = new System.Windows.Forms.TextBox();
```

```
    this.contextMenuColuna = new  
System.Windows.Forms.ContextMenuStrip(this.components);
```

```
    this.adicionarColunaToolStripMenuItem = new  
System.Windows.Forms.ToolStripItem();
```

```
    this.excluirColunaToolStripMenuItem = new  
System.Windows.Forms.ToolStripItem();
```

```
    this.toolStripSeparator1 = new System.Windows.Forms.ToolStripSeparator();
```

```
    this.renomearToolStripMenuItem = new  
System.Windows.Forms.ToolStripItem();
```

```
    this.contextMenuLinha = new  
System.Windows.Forms.ContextMenuStrip(this.components);
```



```

        this.ExcluirLinhaToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();

        this.TitleLabel = new System.Windows.Forms.Label();

        this.TotalLabel = new System.Windows.Forms.Label();

        this.AnalyzeButton = new System.Windows.Forms.Button();

        this.RemoveLinesButton = new System.Windows.Forms.Button();

        this.RemoveColumnButton = new System.Windows.Forms.Button();

        this.AddColumnButton = new System.Windows.Forms.Button();

        this.sectionHeader1 = new Oraculo.Global.SectionHeader();

        this.DataGrid = new System.Windows.Forms.DataGridView();

        this.contextMenuColuna.SuspendLayout();

        this.contextMenuLinha.SuspendLayout();

        ((System.ComponentModel.ISupportInitialize)(this.DataGrid)).BeginInit();

        this.SuspendLayout();

        //

        // textBoxHeader

        //

        this.textBoxHeader.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

        this.textBoxHeader.Location = new System.Drawing.Point(227, 253);

        this.textBoxHeader.Multiline = true;

```

```

this.textBoxHeader.Name = "textBoxHeader";

this.textBoxHeader.Size = new System.Drawing.Size(144, 20);

this.textBoxHeader.TabIndex = 3;

this.textBoxHeader.Visible = false;

this.textBoxHeader.KeyPress += new
System.Windows.Forms.KeyPressEventHandler(this.textBoxHeader_KeyPress);

this.textBoxHeader.Leave += new System.EventHandler(this.textBoxHeader_Leave);

//

// contextMenuColuna

//

this.contextMenuColuna.Items.AddRange(new
System.Windows.Forms.ToolStripItem[] {

this.adicionarColunaToolStripMenuItem,

this.excluirColunaToolStripMenuItem,

this.toolStripSeparator1,

this.renomearToolStripMenuItem });

this.contextMenuColuna.Name = "contextMenuColuna";

this.contextMenuColuna.Size = new System.Drawing.Size(174, 76);

//

// adicionarColunaToolStripMenuItem

//

```

```

        this.adicionarColunaToolStripMenuItem.Image =
global::Oraculo.Properties.Resources.add;

        this.adicionarColunaToolStripMenuItem.Name =
"adicionarColunaToolStripMenuItem";

        this.adicionarColunaToolStripMenuItem.Size = new System.Drawing.Size(173, 22);

        this.adicionarColunaToolStripMenuItem.Text = "Adicionar coluna...";

        this.adicionarColunaToolStripMenuItem.Click += new
System.EventHandler(this.adicionarColunaToolStripMenuItem_Click);

//

// excluirColunaToolStripMenuItem

//

        this.excluirColunaToolStripMenuItem.Image =
global::Oraculo.Properties.Resources.delete;

        this.excluirColunaToolStripMenuItem.Name = "excluirColunaToolStripMenuItem";

        this.excluirColunaToolStripMenuItem.Size = new System.Drawing.Size(173, 22);

        this.excluirColunaToolStripMenuItem.Text = "Excluir coluna";

        this.excluirColunaToolStripMenuItem.Click += new
System.EventHandler(this.excluirColunaToolStripMenuItem_Click);

//

// toolStripSeparator1

//

        this.toolStripSeparator1.Name = "toolStripSeparator1";

```

```

this.toolStripSeparator1.Size = new System.Drawing.Size(170, 6);

//

// renomearToolStripMenuItem

//

this.renomearToolStripMenuItem.Name = "renomearToolStripMenuItem";

this.renomearToolStripMenuItem.Size = new System.Drawing.Size(173, 22);

this.renomearToolStripMenuItem.Text = "Renomear";

this.renomearToolStripMenuItem.Click += new
System.EventHandler(this.renomearToolStripMenuItem_Click);

//

// contextMenuLinha

//

this.contextMenuLinha.Items.AddRange(new
System.Windows.Forms.ToolStripItem[] {

this.ExcluirLinhaToolStripMenuItem});

this.contextMenuLinha.Name = "contextMenuColuna";

this.contextMenuLinha.Size = new System.Drawing.Size(139, 26);

//

// ExcluirLinhaToolStripMenuItem

//

```

```

        this.ExcluirLinhaToolStripMenuItem.Image =
global::Oraculo.Properties.Resources.delete;

        this.ExcluirLinhaToolStripMenuItem.Name = "ExcluirLinhaToolStripMenuItem";

        this.ExcluirLinhaToolStripMenuItem.Size = new System.Drawing.Size(138, 22);

        this.ExcluirLinhaToolStripMenuItem.Text = "Excluir linha";

        this.ExcluirLinhaToolStripMenuItem.Click += new
System.EventHandler(this.ExcluirLinhaToolStripMenuItem_Click);

        //

        // titleLabel

        //

        this.titleLabel.AutoSize = true;

        this.titleLabel.Font = new System.Drawing.Font("Segoe UI", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

        this.titleLabel.Location = new System.Drawing.Point(20, 139);

        this.titleLabel.Margin = new System.Windows.Forms.Padding(20);

        this.titleLabel.Name = "titleLabel";

        this.titleLabel.Size = new System.Drawing.Size(114, 25);

        this.titleLabel.TabIndex = 4;

        this.titleLabel.Text = "Meus dados";

        //

        // TotalLabel

```

```

//

this.TotalLabel.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Left)

| System.Windows.Forms.AnchorStyles.Right)));

this.TotalLabel.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.TotalLabel.Location = new System.Drawing.Point(293, 435);

this.TotalLabel.Margin = new System.Windows.Forms.Padding(10);

this.TotalLabel.Name = "TotalLabel";

this.TotalLabel.Size = new System.Drawing.Size(330, 28);

this.TotalLabel.TabIndex = 5;

this.TotalLabel.Text = "Total:";

this.TotalLabel.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

//

// AnalyzeButton

//

this.AnalyzeButton.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Left)));

this.AnalyzeButton.BackColor = System.Drawing.Color.Gainsboro;

this.AnalyzeButton.Enabled = false;

```

```

this.AnalyzeButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;

this.AnalyzeButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.AnalyzeButton.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.AnalyzeButton.Image = global::Oraculo.Properties.Resources.analysis_flat;

this.AnalyzeButton.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.AnalyzeButton.Location = new System.Drawing.Point(20, 435);

this.AnalyzeButton.Margin = new System.Windows.Forms.Padding(10);

this.AnalyzeButton.Name = "AnalyzeButton";

this.AnalyzeButton.Size = new System.Drawing.Size(86, 28);

this.AnalyzeButton.TabIndex = 4;

this.AnalyzeButton.Text = "Analisar...";

this.AnalyzeButton.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

this.AnalyzeButton.UseVisualStyleBackColor = false;

this.AnalyzeButton.Click += new System.EventHandler(this.AnalyzeButton_Click);

//

// RemoveLinesButton

//

this.RemoveLinesButton.BackColor = System.Drawing.Color.Gainsboro;

this.RemoveLinesButton.Enabled = false;

```

```

        this.RemoveLinesButton.FlatAppearance.BorderColor =
System.Drawing.Color.Gainsboro;

        this.RemoveLinesButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

        this.RemoveLinesButton.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

        this.RemoveLinesButton.Image =
global::Oraculo.Properties.Resources.delete_row_flat;

        this.RemoveLinesButton.ImageAlign =
System.Drawing.ContentAlignment.MiddleLeft;

        this.RemoveLinesButton.Location = new System.Drawing.Point(301, 194);

        this.RemoveLinesButton.Margin = new System.Windows.Forms.Padding(10);

        this.RemoveLinesButton.Name = "RemoveLinesButton";

        this.RemoveLinesButton.Size = new System.Drawing.Size(130, 28);

        this.RemoveLinesButton.TabIndex = 4;

        this.RemoveLinesButton.Text = "Remover Linha(s)";

        this.RemoveLinesButton.TextAlign =
System.Drawing.ContentAlignment.MiddleRight;

        this.RemoveLinesButton.UseVisualStyleBackColor = false;

        this.RemoveLinesButton.Click += new
System.EventHandler(this.RemoveLineButton_Click);

//

// RemoveColumnButton

```



```
//  
  
this.RemoveColumnButton.BackColor = System.Drawing.Color.Gainsboro;  
  
this.RemoveColumnButton.Enabled = false;  
  
this.RemoveColumnButton.FlatAppearance.BorderColor =  
System.Drawing.Color.Gainsboro;  
  
this.RemoveColumnButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;  
  
this.RemoveColumnButton.Font = new System.Drawing.Font("Segoe UI", 9F,  
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));  
  
this.RemoveColumnButton.Image =  
global::Oraculo.Properties.Resources.delete_column_flat;  
  
this.RemoveColumnButton.ImageAlign =  
System.Drawing.ContentAlignment.MiddleLeft;  
  
this.RemoveColumnButton.Location = new System.Drawing.Point(166, 194);  
  
this.RemoveColumnButton.Margin = new System.Windows.Forms.Padding(10, 10, 0,  
10);  
  
this.RemoveColumnButton.Name = "RemoveColumnButton";  
  
this.RemoveColumnButton.Size = new System.Drawing.Size(125, 28);  
  
this.RemoveColumnButton.TabIndex = 3;  
  
this.RemoveColumnButton.Text = "Remover Coluna";  
  
this.RemoveColumnButton.TextAlign =  
System.Drawing.ContentAlignment.MiddleRight;  
  
this.RemoveColumnButton.UseVisualStyleBackColor = false;
```

```

        this.RemoveColumnButton.Click += new
System.EventHandler(this.RemoveColumnButton_Click);

//

// AddColumnButton

//

this.AddColumnButton.BackColor = System.Drawing.Color.Gainsboro;

this.AddColumnButton.FlatAppearance.BorderColor =
System.Drawing.Color.Gainsboro;

this.AddColumnButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.AddColumnButton.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.AddColumnButton.Image =
global::Oraculo.Properties.Resources.add_column_flat;

this.AddColumnButton.ImageAlign =
System.Drawing.ContentAlignment.MiddleLeft;

this.AddColumnButton.Location = new System.Drawing.Point(20, 194);

this.AddColumnButton.Margin = new System.Windows.Forms.Padding(10, 10, 0,
10);

this.AddColumnButton.Name = "AddColumnButton";

this.AddColumnButton.Size = new System.Drawing.Size(136, 28);

this.AddColumnButton.TabIndex = 2;

this.AddColumnButton.Text = "Adicionar Coluna...";

```

```

        this.AddColumnButton.TextAlign =
System.Drawing.ContentAlignment.MiddleRight;

        this.AddColumnButton.UseVisualStyleBackColor = false;

        this.AddColumnButton.Click += new
System.EventHandler(this.AddColumnButton_Click);

//

// sectionHeader1

//

this.sectionHeader1.BackColor = System.Drawing.Color.White;

this.sectionHeader1.BreadCrumbText = "Início > Dados";

this.sectionHeader1.Dock = System.Windows.Forms.DockStyle.Top;

this.sectionHeader1.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.sectionHeader1.Location = new System.Drawing.Point(0, 0);

this.sectionHeader1.Margin = new System.Windows.Forms.Padding(0);

this.sectionHeader1.Name = "sectionHeader1";

this.sectionHeader1.Size = new System.Drawing.Size(640, 119);

this.sectionHeader1.TabIndex = 6;

//

// DataGrid

//

```

```

        this.DataGrid.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)

        | System.Windows.Forms.AnchorStyles.Left)

        | System.Windows.Forms.AnchorStyles.Right))));

        this.DataGrid.BackgroundColor = System.Drawing.Color.Silver;

        this.DataGrid.BorderStyle = System.Windows.Forms.BorderStyle.None;

        this.DataGrid.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;

        this.DataGrid.Location = new System.Drawing.Point(20, 232);

        this.DataGrid.Margin = new System.Windows.Forms.Padding(20, 0, 20, 0);

        this.DataGrid.Name = "DataGrid";

        this.DataGrid.RowHeadersWidth = 25;

        this.DataGrid.Size = new System.Drawing.Size(600, 193);

        this.DataGrid.TabIndex = 7;

        this.DataGrid.CellContextMenuStripNeeded += new
System.Windows.Forms.DataGridViewCellContextMenuStripNeededEventHandler(this.Data
Grid_CellContextMenuStripNeeded);

        this.DataGrid.CellMouseDown += new
System.Windows.Forms.DataGridViewCellMouseEventHandler(this.DataGrid_CellMouseDo
wn);

        this.DataGrid.ColumnAdded += new
System.Windows.Forms.DataGridViewColumnEventHandler(this.DataGrid_ColumnAdded);

```

```

        this.DataGrid.ColumnDisplayIndexChanged += new
System.Windows.Forms.DataGridViewColumnEventHandler(this.DataGrid_ColumnDisplayI
ndexChanged);

        this.DataGrid.ColumnHeaderMouseDoubleClick += new
System.Windows.Forms.DataGridViewCellMouseEventHandler(this.DataGrid_ColumnHead
erMouseDoubleClick);

        this.DataGrid.ColumnRemoved += new
System.Windows.Forms.DataGridViewColumnEventHandler(this.DataGrid_ColumnRemove
d);

        this.DataGrid.DataError += new
System.Windows.Forms.DataGridViewDataErrorEventHandler(this.DataGrid_DataError);

        this.DataGrid.RowsAdded += new
System.Windows.Forms.DataGridViewRowsAddedEventHandler(this.DataGrid_RowsAdded
);

        this.DataGrid.RowsRemoved += new
System.Windows.Forms.DataGridViewRowsRemovedEventHandler(this.DataGrid_RowsRe
moved);

        this.DataGrid.SelectionChanged += new
System.EventHandler(this.DataGrid_SelectionChanged);

        this.DataGrid.KeyDown += new
System.Windows.Forms.KeyEventHandler(this.DataGrid_KeyDown);

        //

        // DataView

        //

        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

```

```
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.AutoScroll = true;

this.BackColor = System.Drawing.Color.White;

this.Controls.Add(this.sectionHeader1);

this.Controls.Add(this.TotalLabel);

this.Controls.Add(this.textBoxHeader);

this.Controls.Add(this.AnalyzeButton);

this.Controls.Add(this.RemoveLinesButton);

this.Controls.Add(this.TitleLabel);

this.Controls.Add(this.RemoveColumnButton);

this.Controls.Add(this.AddColumnButton);

this.Controls.Add(this.DataGrid);

this.Font = new System.Drawing.Font("Segoe UI", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.Margin = new System.Windows.Forms.Padding(0);

this.Name = "DataView";

this.Size = new System.Drawing.Size(640, 480);

this.SizeChanged += new System.EventHandler(this.DataView_SizeChanged);

this.contextMenuColuna.ResumeLayout(false);

this.contextMenuLinha.ResumeLayout(false);
```

```

        ((System.ComponentModel.ISupportInitialize)(this.DataGrid)).EndInit();

        this.ResumeLayout(false);

        this.PerformLayout();

    }

#endregion

private System.Windows.Forms.Button AddColumnButton;

private System.Windows.Forms.Button RemoveColumnButton;

private System.Windows.Forms.Button AnalyzeButton;

private System.Windows.Forms.TextBox textBoxHeader;

private System.Windows.Forms.ContextMenuStrip contextMenuColuna;

private System.Windows.Forms.ToolStripMenuItem
adicionarColunaToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem excluirColunaToolStripMenuItem;

private System.Windows.Forms.ToolStripSeparator toolStripSeparator1;

private System.Windows.Forms.ToolStripMenuItem renomearToolStripMenuItem;

private System.Windows.Forms.ContextMenuStrip contextMenuLinha;

private System.Windows.Forms.ToolStripMenuItem ExcluirLinhaToolStripMenuItem;

private System.Windows.Forms.Button RemoveLinesButton;

```

```
private System.Windows.Forms.Label TitleLabel;

private System.Windows.Forms.Label TotalLabel;

private Global.SectionHeader sectionHeader1;

private System.Windows.Forms.DataGridView DataGrid;

}

}
```

Oraculo\Screens\ExcelFileImportWizardControl.cs

```
?using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.IO;

using System.Text;

using System.Windows.Forms;

using ExcelDataReader;
```



```
namespace Oraculo.Telas

{

    public partial class ExcelFileImportWizardControl : Oraculo.Global.WizardControl

    {

        #region Properties

        public String FilePath { get; set; } = "";

        public string SelectedSheet { get; set; } = "";

        public bool ShowColumnTitles { get; set; } = false;

        public DataTable Data { get; set; }

        #endregion

        #region Events

        public event EventHandler AfterImportData;

        #endregion

    }

}
```

```
#region Métodos
```

```
public ExcelFileImportWizardControl()
```

```
{
```

```
    InitializeComponent();
```

```
    SelectFilePanel.AllowDrop = true;
```

```
    SelectFilePanel.DragDrop += new DragEventHandler(SelectFileTextBox_DragDrop);
```

```
    SelectFilePanel.DragEnter += new  
DragEventHandler(SelectFileTextBox_DragEnter);
```

```
    SelectFilePanel.DragOver += new DragEventHandler(SelectFileTextBox_DragOver);
```

```
    TotalSteps = 3;
```

```
    ShowStep(1);
```

```
}
```

```
public static List<string> GetSheetList(string path)
```

```
{
```

```
    List<string> sheets = new List<string>();
```

```

try
{
    using (var stream = File.Open(path.TrimEnd(), FileMode.Open, FileAccess.Read))
    {
        using (var reader = ExcelReaderFactory.CreateReader(stream))
        {
            var result = reader.AsDataSet();

            foreach (DataTable sheet in result.Tables)
            {
                sheets.Add(sheet.TableName);
            }
        }
    }
}

catch (ExcelDataReader.Exceptions.HeaderException)
{
    MessageBox.Show("Você não informou um arquivo do Excel (*.xls; *.xlsx).  
Informe um novo arquivo e tente novamente.", "Erro ao importar planilha do Excel",  
MessageBoxButtons.OK, MessageBoxIcon.Error);
}

```

```

catch (System.IO.IOException)

{

    MessageBox.Show("O arquivo do Excel (*.xls; *.xlsx) está aberto. Feche o arquivo
primeiro e tente novamente.", "Erro ao importar planilha do Excel", MessageBoxButtons.OK,
MessageBoxIcon.Error);

}

return sheets;

}

private DataTable GetDataTableFromExcel(string path, string sheet, bool
isFirstRowHeader)

{

    using (var stream = File.Open(path.TrimEnd(), FileMode.Open, FileAccess.Read))

    {

        using (var reader = ExcelReaderFactory.CreateReader(stream))

        {

            var result = reader.AsDataSet(new ExcelDataSetConfiguration()

            {

                UseColumnDataType = true,

                FilterSheet = (tableReader, sheetIndex) => true,

```

```

ConfigureDataTable = (tableReader) => new ExcelDataTableConfiguration()

{

    EmptyColumnNamePrefix = "Coluna",

    UseHeaderRow = isFirstRowHeader

    //ReadHeaderRow = (rowReader) => { rowReader.Read(); }

}

});

foreach (DataTable dt in result.Tables)

{

    if (dt.TableName == sheet)

    {

        return dt;

    }

}

}

return null;

}

```

```

private void FillSheetsListBox()

{

    //limpa a lista de planilhas e preenche

    SelectSheetListBox.Items.Clear();

    foreach (string item in GetSheetList(FilePath))

    {

        SelectSheetListBox.Items.Add(item);

    }

    //se só tem um item, seleciona

    if (SelectSheetListBox.Items.Count == 1) SelectSheetListBox.SelectedIndex = 0;

}

private Boolean ImportData()

{

    Data = GetDataTableFromExcel(FilePath, SelectedSheet, ShowColumnTitles);

    return (Data.Rows.Count > 0);

}

```

```

public override bool ValidateStep()

{

    //preenche o panel conforme a etapa

    switch (Step)

    {

        case 1:

            //solicita ao usuário o arquivo que deseja importar

            if (FilePath == "")

            {

                MessageBox.Show(this, "Você precisa informar qual é a planilha que deseja
importar.", "Erro ao importar planilha do Excel", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);

                SelectFileButton.Focus();

                return false;

            }

            else

            {

                SelectFileTextBox.Text = FilePath;

            }

        //preenche a lista de planilhas

```

```

FillSheetsListBox();

//se não tem planilhas, não exibe a próxima tela

if (SelectSheetListBox.Items.Count == 0)

{

    //MessageBox.Show(this, "A pasta de trabalho do Excel selecionada não contém planilhas. Selecione outro arquivo e tente novamente.", "Erro ao abrir arquivo do Excel (*.xls;*.xlsx)", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

    return false;

}

break;

case 2:

//verifica se alguma planilha foi selecionada

if (SelectSheetListBox.SelectedIndex == -1)

{

    MessageBox.Show(this, "Você não escolheu a planilha. Selecione uma planilha na lista e tente novamente.", "Planilha para importar", MessageBoxButtons.OK, MessageBoxIcon.Error);

    return false;

}

```



```

//armazena a planilha selecionada

SelectedSheet = SelectSheetListBox.SelectedItem.ToString();

//exibe mensagem para confirmar se a primeira linha tem títulos dos campos

ShowColumnTitles = (MessageBox.Show(this, "A primeira linha contém os
títulos dos campos?", "Títulos das colunas", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes);

//exibe tela com mensagem para usuário aguardar a importação

SelectFilePanel.Visible = false;

ImportingPanel.Visible = true;

//importa os dados

if (!ImportData())

{

    MessageBox.Show(this, "Não foi possível importar os dados. O arquivo está
vazio ou corrompido ou a planilha está vazia. Verifique o arquivo informado e tente
novamente.", "Erro ao importar pasta de trabalho do Excel", MessageBoxButtons.OK,
MessageBoxIcon.Error);

    return false;

}

```

```

else

{

    //exibe os dados importados para o usuário confirmar se está tudo OK.

    DataSubtitleLabel.Text = String.Format("Foram localizadas {0} coluna(s) e
{1} linha(s). Confira se os dados foram importados corretamente:", Data.Columns.Count,
Data.Rows.Count);

    DataGrid.DataSource = Data;

}

break;

case 3:

    if (MessageBox.Show(this, "Os dados importados estão corretos?", "Confirmação
dos dados", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No)

    {

        Step = 1;

        return false;

    }

    break;

}

return true;

}

```

```
public override void ShowStep()

{

    SelectFilePanel.Visible = (Step == 1);

    SelectSheetPanel.Visible = (Step == 2);

    ImportingPanel.Visible = false;

    DataPanel.Visible = (Step == 3);

    base.ShowStep();

}

public override bool CloseWizard(bool cancel = false)

{

    if (cancel)

    {

        ShowStep(1);

    }

    else

    {

        AfterImportData.Invoke(Data);

    }

}
```

```

        return base.CloseWizard(cancel);
    }

#endregion

private void SelectFileButton_Click(object sender, EventArgs e)
{
    //abre a caixa de diálogo de sistema para localizar uma arquivo csv

    OpenFileDialog tela = new OpenFileDialog();

    if (FilePath != "")
    {
        tela.FileName = FilePath;
    }
    else
    {
        tela.InitialDirectory =
System.Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
    }

    tela.Filter = "Pastas de trabalho do Microsoft Excel|*.xls;*.xlsx|Todos os
arquivos|*.*";
}

```

```

tela.DefaultExt = "xls";

tela.CheckFileExists = true;

tela.CheckPathExists = true;

if (tela.ShowDialog() == DialogResult.OK)
{
    FilePath = tela.FileName;

    SelectFileTextBox.Text = FilePath;
}
}

private void SelectFileTextBox_DragDrop(object sender, DragEventArgs e)
{
    if (e.Data.GetDataPresent(DataFormats.FileDrop, false) == true)
    {
        e.Effect = DragDropEffects.All;
    }
}

private void SelectFileTextBox_DragEnter(object sender, DragEventArgs e)

```

```

{

    string[] file = (string[])e.Data.GetData(DataFormats.FileDrop);

    if (file != null && file.Length != 0)

    {

        SelectFileTextBox.Text = file[0];

        FilePath = file[0];

    }

}

private void SelectFileTextBox_DragOver(object sender, DragEventArgs e)

{

    if (e.Data.GetDataPresent(DataFormats.FileDrop))

    {

        e.Effect = DragDropEffects.Copy;

    }

    else

    {

        e.Effect = DragDropEffects.None;

    }

}

```

```

private void SelectFileTextBox_TextChanged(object sender, EventArgs e)

{

    if (SelectFileTextBox.Text != "Clique no botão localizar ou arraste o arquivo aqui!")

        SelectFileTextBox.ForeColor = System.Drawing.SystemColors.WindowText;

    else

        SelectFileTextBox.ForeColor = Color.Gray;

}

}

}

```

Oraculo\Screens\ExcelFileImportWizardControl.Designer.cs

```

?namespace Oraculo.Telas

{

    partial class ExcelFileImportWizardControl

    {

        /// <summary>

        /// Required designer variable.

        /// </summary>

        private System.ComponentModel.IContainer components = null;

```

```
/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>

protected override void Dispose(bool disposing)

{

    if (disposing && (components != null))

    {

        components.Dispose();

    }

    base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.
```



```
/// </summary>
```

```
private void InitializeComponent()
```

```
{
```

```
    this.SelectFilePanel = new System.Windows.Forms.Panel();
```

```
    this.SelectFileButton = new System.Windows.Forms.Button();
```

```
    this.SelectFileTextBox = new System.Windows.Forms.TextBox();
```

```
    this.SelectFileLabel = new System.Windows.Forms.Label();
```

```
    this.SelectSheetPanel = new System.Windows.Forms.Panel();
```

```
    this.SelectSheetListBox = new System.Windows.Forms.ListBox();
```

```
    this.SelectSheetLabel = new System.Windows.Forms.Label();
```

```
    this.ImportingPanel = new System.Windows.Forms.Panel();
```

```
    this.ImportingLabel = new System.Windows.Forms.Label();
```

```
    this.DataPanel = new System.Windows.Forms.Panel();
```

```
    this.DataSubtitleLabel = new System.Windows.Forms.Label();
```

```
    this.DataTitleLabel = new System.Windows.Forms.Label();
```

```
    this.DataGrid = new System.Windows.Forms.DataGridView();
```

```
    this.ButtonPanel.SuspendLayout();
```

```
    this.SelectFilePanel.SuspendLayout();
```

```
    this.SelectSheetPanel.SuspendLayout();
```

```
    this.ImportingPanel.SuspendLayout();
```

```
this.DataPanel.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.DataGrid)).BeginInit();

this.SuspendLayout();

//

// sectionHeader1

//

this.sectionHeader1.BreadCrumbText = "Início > Dados > Importar";

//

// TitleLabel

//

this.TitleLabel.Text = "Importar arquivos do Excel (*.xlsx)";

//

// ButtonPanel

//

this.ButtonPanel.Margin = new System.Windows.Forms.Padding(23);

//

// NextButton

//

this.NextButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;

this.NextButton.Location = new System.Drawing.Point(316, 21);
```

```
this.NextButton.Margin = new System.Windows.Forms.Padding(7, 27, 27, 27);

//

// PreviousButton

//

this.PreviousButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;

this.PreviousButton.Location = new System.Drawing.Point(218, 21);

this.PreviousButton.Margin = new System.Windows.Forms.Padding(27, 27, 0, 27);

//

// SelectFilePanel

//

this.SelectFilePanel.Controls.Add(this.SelectFileButton);

this.SelectFilePanel.Controls.Add(this.SelectFileTextBox);

this.SelectFilePanel.Controls.Add(this.SelectFileLabel);

this.SelectFilePanel.Dock = System.Windows.Forms.DockStyle.Fill;

this.SelectFilePanel.Location = new System.Drawing.Point(0, 188);

this.SelectFilePanel.Margin = new System.Windows.Forms.Padding(0);

this.SelectFilePanel.Name = "SelectFilePanel";

this.SelectFilePanel.Size = new System.Drawing.Size(640, 223);

this.SelectFilePanel.TabIndex = 10;

//
```

```

// SelectFileButton

//

this.SelectFileButton.BackColor = System.Drawing.Color.Gainsboro;

this.SelectFileButton.FlatAppearance.BorderColor =
System.Drawing.Color.Gainsboro;

this.SelectFileButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.SelectFileButton.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.SelectFileButton.Location = new System.Drawing.Point(407, 19);

this.SelectFileButton.Margin = new System.Windows.Forms.Padding(3, 5, 3, 5);

this.SelectFileButton.Name = "SelectFileButton";

this.SelectFileButton.Size = new System.Drawing.Size(103, 29);

this.SelectFileButton.TabIndex = 2;

this.SelectFileButton.Text = "Localizar...";

this.SelectFileButton.UseVisualStyleBackColor = false;

this.SelectFileButton.Click += new
System.EventHandler(this.SelectFileButton_Click);

//

// SelectFileTextBox

//

this.SelectFileTextBox.AllowDrop = true;

this.SelectFileTextBox.BackColor = System.Drawing.SystemColors.ButtonFace;

```

```
this.SelectFileTextBox.ForeColor = System.Drawing.Color.Gray;

this.SelectFileTextBox.Location = new System.Drawing.Point(24, 23);

this.SelectFileTextBox.Margin = new System.Windows.Forms.Padding(20, 0, 5, 20);

this.SelectFileTextBox.Name = "SelectFileTextBox";

this.SelectFileTextBox.ReadOnly = true;

this.SelectFileTextBox.Size = new System.Drawing.Size(375, 23);

this.SelectFileTextBox.TabIndex = 1;

this.SelectFileTextBox.Text = "Clique no botão localizar ou arraste o arquivo aqui!";

this.SelectFileTextBox.TextChanged += new
System.EventHandler(this.SelectFileTextBox_TextChanged);

this.SelectFileTextBox.DragDrop += new
System.Windows.Forms.DragEventHandler(this.SelectFileTextBox_DragDrop);

this.SelectFileTextBox.DragEnter += new
System.Windows.Forms.DragEventHandler(this.SelectFileTextBox_DragEnter);

this.SelectFileTextBox.DragOver += new
System.Windows.Forms.DragEventHandler(this.SelectFileTextBox_DragOver);

//

// SelectFileLabel

//

this.SelectFileLabel.AutoSize = true;

this.SelectFileLabel.Location = new System.Drawing.Point(20, 0);
```

```
this.SelectFileLabel.Margin = new System.Windows.Forms.Padding(20, 0, 20, 20);

this.SelectFileLabel.Name = "SelectFileLabel";

this.SelectFileLabel.Size = new System.Drawing.Size(285, 15);

this.SelectFileLabel.TabIndex = 0;

this.SelectFileLabel.Text = "Selecione o arquivo que contém os dados desejados:";

//

// SelectSheetPanel

//

this.SelectSheetPanel.Controls.Add(this.SelectSheetListBox);

this.SelectSheetPanel.Controls.Add(this.SelectSheetLabel);

this.SelectSheetPanel.Dock = System.Windows.Forms.DockStyle.Fill;

this.SelectSheetPanel.Location = new System.Drawing.Point(0, 188);

this.SelectSheetPanel.Margin = new System.Windows.Forms.Padding(0);

this.SelectSheetPanel.Name = "SelectSheetPanel";

this.SelectSheetPanel.Size = new System.Drawing.Size(640, 223);

this.SelectSheetPanel.TabIndex = 11;

//

// SelectSheetListBox

//
```

```

        this.SelectSheetListBox.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)

        | System.Windows.Forms.AnchorStyles.Right)));

        this.SelectSheetListBox.FormattingEnabled = true;

        this.SelectSheetListBox.ItemHeight = 15;

        this.SelectSheetListBox.Location = new System.Drawing.Point(24, 25);

        this.SelectSheetListBox.Margin = new System.Windows.Forms.Padding(20);

        this.SelectSheetListBox.Name = "SelectSheetListBox";

        this.SelectSheetListBox.ScrollAlwaysVisible = true;

        this.SelectSheetListBox.Size = new System.Drawing.Size(596, 169);

        this.SelectSheetListBox.TabIndex = 1;

        //

        // SelectSheetLabel

        //

        this.SelectSheetLabel.AutoSize = true;

        this.SelectSheetLabel.Location = new System.Drawing.Point(20, 0);

        this.SelectSheetLabel.Margin = new System.Windows.Forms.Padding(20, 0, 20, 20);

        this.SelectSheetLabel.Name = "SelectSheetLabel";

        this.SelectSheetLabel.Size = new System.Drawing.Size(163, 15);

        this.SelectSheetLabel.TabIndex = 0;

```

```

this.SelectSheetLabel.Text = "Selecione a planilha desejada:";

//

// ImportingPanel

//

this.ImportingPanel.Controls.Add(this.ImportingLabel);

this.ImportingPanel.Dock = System.Windows.Forms.DockStyle.Fill;

this.ImportingPanel.Location = new System.Drawing.Point(0, 188);

this.ImportingPanel.Name = "ImportingPanel";

this.ImportingPanel.Size = new System.Drawing.Size(640, 223);

this.ImportingPanel.TabIndex = 12;

//

// ImportingLabel

//

this.ImportingLabel.Anchor = System.Windows.Forms.AnchorStyles.None;

this.ImportingLabel.AutoSize = true;

this.ImportingLabel.Font = new System.Drawing.Font("Segoe UI", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.ImportingLabel.Location = new System.Drawing.Point(205, 101);

this.ImportingLabel.Name = "ImportingLabel";

this.ImportingLabel.Size = new System.Drawing.Size(230, 21);

```



```
this.ImportingLabel.TabIndex = 0;

this.ImportingLabel.Text = "Aguarde... importando os dados";

//

// DataPanel

//

this.DataPanel.Controls.Add(this.DataSubtitleLabel);

this.DataPanel.Controls.Add(this.DataTitleLabel);

this.DataPanel.Controls.Add(this.DataGrid);

this.DataPanel.Dock = System.Windows.Forms.DockStyle.Fill;

this.DataPanel.Location = new System.Drawing.Point(0, 188);

this.DataPanel.Margin = new System.Windows.Forms.Padding(0);

this.DataPanel.Name = "DataPanel";

this.DataPanel.Size = new System.Drawing.Size(640, 223);

this.DataPanel.TabIndex = 13;

//

// DataSubtitleLabel

//

this.DataSubtitleLabel.AutoSize = true;

this.DataSubtitleLabel.Location = new System.Drawing.Point(21, 26);

this.DataSubtitleLabel.Margin = new System.Windows.Forms.Padding(20, 0, 20, 20);
```

```

this.DataSubtitleLabel.Name = "DataSubtitleLabel";

this.DataSubtitleLabel.Size = new System.Drawing.Size(507, 15);

this.DataSubtitleLabel.TabIndex = 7;

this.DataSubtitleLabel.Text = "Foram localizados {0} campos e {1} casos. Confira se
os dados foram importados co" +

"rretamente:";

//

// DataTitleLabel

//

this.DataTitleLabel.AutoSize = true;

this.DataTitleLabel.Font = new System.Drawing.Font("Segoe UI", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.DataTitleLabel.Location = new System.Drawing.Point(20, 0);

this.DataTitleLabel.Margin = new System.Windows.Forms.Padding(20, 0, 20, 5);

this.DataTitleLabel.Name = "DataTitleLabel";

this.DataTitleLabel.Size = new System.Drawing.Size(164, 21);

this.DataTitleLabel.TabIndex = 6;

this.DataTitleLabel.Text = "Importação concluída!";

//

// DataGrid

//

```

```

this.DataGrid.AllowUserToAddRows = false;

this.DataGrid.AllowUserToDeleteRows = false;

this.DataGrid.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)

| System.Windows.Forms.AnchorStyles.Left)

| System.Windows.Forms.AnchorStyles.Right))));

this.DataGrid.AutoSizeColumnsMode =
System.Windows.Forms.DataGridViewAutoSizeColumnsMode.DisplayedCells;

this.DataGrid.BackgroundColor = System.Drawing.Color.Silver;

this.DataGrid.BorderStyle = System.Windows.Forms.BorderStyle.None;

this.DataGrid.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;

this.DataGrid.Location = new System.Drawing.Point(20, 56);

this.DataGrid.Margin = new System.Windows.Forms.Padding(20);

this.DataGrid.Name = "DataGrid";

this.DataGrid.ReadOnly = true;

this.DataGrid.RowHeadersWidth = 25;

this.DataGrid.Size = new System.Drawing.Size(600, 147);

this.DataGrid.TabIndex = 2;

//

```

```
// ExcelFileImportWizardControl

//

this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);

this.Controls.Add(this.SelectSheetPanel);

this.Controls.Add(this.DataPanel);

this.Controls.Add(this.SelectFilePanel);

this.Controls.Add(this.ImportingPanel);

this.Name = "ExcelFileImportWizardControl";

this.Title = "Importar arquivos do Excel (*.xlsx)";

this.Controls.SetChildIndex(this.sectionHeader1, 0);

this.Controls.SetChildIndex(this.TitleLabel, 0);

this.Controls.SetChildIndex(this.ButtonPanel, 0);

this.Controls.SetChildIndex(this.ImportingPanel, 0);

this.Controls.SetChildIndex(this.SelectFilePanel, 0);

this.Controls.SetChildIndex(this.DataPanel, 0);

this.Controls.SetChildIndex(this.SelectSheetPanel, 0);

this.ButtonPanel.ResumeLayout(false);

this.SelectFilePanel.ResumeLayout(false);

this.SelectFilePanel.PerformLayout();

this.SelectSheetPanel.ResumeLayout(false);
```

```
    this.SelectSheetPanel.PerformLayout();

    this.ImportingPanel.ResumeLayout(false);

    this.ImportingPanel.PerformLayout();

    this.DataPanel.ResumeLayout(false);

    this.DataPanel.PerformLayout();

    ((System.ComponentModel.ISupportInitialize)(this.DataGrid)).EndInit();

    this.ResumeLayout(false);

}

```

```
#endregion

```

```
private System.Windows.Forms.Panel SelectFilePanel;

private System.Windows.Forms.Button SelectFileButton;

private System.Windows.Forms.TextBox SelectFileTextBox;

private System.Windows.Forms.Label SelectFileLabel;

private System.Windows.Forms.Panel SelectSheetPanel;

private System.Windows.Forms.ListBox SelectSheetListBox;

private System.Windows.Forms.Label SelectSheetLabel;

private System.Windows.Forms.Panel ImportingPanel;

```

```
private System.Windows.Forms.Label ImportingLabel;

private System.Windows.Forms.Panel DataPanel;

private System.Windows.Forms.Label DataSubtitleLabel;

private System.Windows.Forms.Label DataTitleLabel;

private System.Windows.Forms.DataGridView DataGrid;

}

}
```

Oraculo\Screens\LoginForm.cs

```
?using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;
```

```

using Oraculo.Global;

namespace Oraculo.Screens
{
    public partial class LoginForm : Form
    {
        private bool Validar()
        {
            if (String.IsNullOrEmpty(textBoxUsuario.Text))
            {
                MessageBox.Show(this, "Você não informou o usuário.", "Erro",
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

                textBoxUsuario.Focus();

                return false;
            }

            if (String.IsNullOrEmpty(textBoxSenha.Text))
            {
                MessageBox.Show(this, "Você não informou a senha.", "Erro",
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

                textBoxSenha.Focus();
            }
        }
    }
}

```

```

        return false;

    }

    return true;

}

//private bool Login(string usuario, string senha)

//{

//    MySqlConnectionStringBuilder stringConexao = new
MySqlConnectionStringBuilder();

//    stringConexao.Server = "108.179.192.39";

//    stringConexao.Port = 3306;

//    stringConexao.UserID = "oracu573_usuario";

//    stringConexao.Password = "_+W9$D@b0-(6";

//    stringConexao.Database = "oracu573_oraculo";

//    MySqlConnection conexao = new MySqlConnection(stringConexao.ToString());

//    try

//    {

```



```

//     string sql = string.Format("SELECT COUNT(*) AS total FROM
oracu573_oraculo.usuarios WHERE(nome = '{0}') AND (senha = '{1}']", usuario, senha);

//     MySqlCommand comando = new MySqlCommand(sql, conexao);

//     //Comando SQL

//     //comando.CommandText = "SELECT COUNT(*) AS total FROM
oracu573_oraculo.usuarios WHERE(nome = '@nome') AND (senha = '@senha)";

//     //comando.Parameters.AddWithValue("@nome", usuario);

//     //comando.Parameters.AddWithValue("@senha", senha);

//     //Conectar com o banco

//     //comando.Connection = conn;

//     conexao.Open();

//     //Executar Comando

//     var resultado = comando.ExecuteScalar();

//     int res = resultado != null ? int.Parse(resultado.ToString()) : 0;

//     if (res > 0)

```

```

//    {

//        return true;

//    }

//    else

//    {

//        MessageBox.Show("Usuário ou Senha inválidos", "Oráculo Estatístico",
MessageBoxButtons.OK, MessageBoxIcon.Warning);

//        return false;

//    }

// }

// catch (MySqlException er)

// {

//     MessageBox.Show("Não foi possível acessar o banco de dados. Verifique sua
conexão com a Internet." + er, "Erro de acesso ao banco de dados", MessageBoxButtons.OK,
MessageBoxIcon.Error);

//     return false;

// }

// finally

// {

//     conexao.Close();

//     conexao.Dispose();

```

```
// }  
  
//}  
  
public LoginForm()  
  
{  
  
    InitializeComponent();  
  
}  
  
private void OKButton_Click(object sender, EventArgs e)  
  
{  
  
    //valida se preencheu usuário e senha  
  
    if (Validar() == true)  
  
    {  
  
        //tenta logar ao banco de dados remoto  
  
        if (User.Login(textBoxUsuario.Text, textBoxSenha.Text))  
  
        {  
  
            this.DialogResult = DialogResult.OK;  
  
        }  
  
    }  
  
}  
  
}
```

```
private void CancelButton_Click(object sender, EventArgs e)

{

    this.DialogResult = DialogResult.Cancel;

}

}

}
```

Oraculo\Screens\LoginForm.Designer.cs

?

```
namespace Oraculo.Screens
```

```
{
```

```
    partial class LoginForm
```

```
    {
```

```
        /// <summary>
```

```
        /// Required designer variable.
```

```
        /// </summary>
```

```
        private System.ComponentModel.IContainer components = null;
```

```
/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>

protected override void Dispose(bool disposing)

{

    if (disposing && (components != null))

    {

        components.Dispose();

    }

    base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>
```

```

private void InitializeComponent()

{

    this.CancelButton = new System.Windows.Forms.Button();

    this.OKButton = new System.Windows.Forms.Button();

    this.labelUsuario = new System.Windows.Forms.Label();

    this.textBoxUsuario = new System.Windows.Forms.TextBox();

    this.textBoxSenha = new System.Windows.Forms.TextBox();

    this.labelSenha = new System.Windows.Forms.Label();

    this.pictureBox1 = new System.Windows.Forms.PictureBox();

    ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();

    this.SuspendLayout();

    //

    // CancelButton

    //

    this.CancelButton.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));

    this.CancelButton.BackColor = System.Drawing.Color.Gainsboro;

    this.CancelButton.DialogResult = System.Windows.Forms.DialogResult.Cancel;

    this.CancelButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;

    this.CancelButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

```

```

this.CancelButton.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.CancelButton.Location = new System.Drawing.Point(266, 149);

this.CancelButton.Margin = new System.Windows.Forms.Padding(10, 20, 20, 20);

this.CancelButton.Name = "CancelButton";

this.CancelButton.Size = new System.Drawing.Size(91, 28);

this.CancelButton.TabIndex = 5;

this.CancelButton.Text = "Cancelar";

this.CancelButton.UseVisualStyleBackColor = false;

this.CancelButton.Click += new System.EventHandler(this.CancelButton_Click);

//

// OKButton

//

this.OKButton.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));

this.OKButton.BackColor = System.Drawing.Color.Gainsboro;

this.OKButton.FlatAppearance.BorderColor = System.Drawing.Color.Gainsboro;

this.OKButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.OKButton.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.OKButton.Location = new System.Drawing.Point(165, 149);

this.OKButton.Margin = new System.Windows.Forms.Padding(20, 20, 0, 20);

```

```
this.OKButton.Name = "OKButton";

this.OKButton.Size = new System.Drawing.Size(91, 28);

this.OKButton.TabIndex = 4;

this.OKButton.Text = "OK";

this.OKButton.UseVisualStyleBackColor = false;

this.OKButton.Click += new System.EventHandler(this.OKButton_Click);

//

// labelUsuario

//

this.labelUsuario.AutoSize = true;

this.labelUsuario.Location = new System.Drawing.Point(162, 24);

this.labelUsuario.Name = "labelUsuario";

this.labelUsuario.Size = new System.Drawing.Size(50, 15);

this.labelUsuario.TabIndex = 0;

this.labelUsuario.Text = "Usuário:";

//

// textBoxUsuario

//

this.textBoxUsuario.Location = new System.Drawing.Point(165, 42);

this.textBoxUsuario.MaxLength = 20;
```



```
this.textBoxUsuario.Name = "textBoxUsuario";

this.textBoxUsuario.Size = new System.Drawing.Size(192, 23);

this.textBoxUsuario.TabIndex = 0;

//

// textBoxSenha

//

this.textBoxSenha.Location = new System.Drawing.Point(165, 92);

this.textBoxSenha.MaxLength = 20;

this.textBoxSenha.Name = "textBoxSenha";

this.textBoxSenha.PasswordChar = '*';

this.textBoxSenha.Size = new System.Drawing.Size(192, 23);

this.textBoxSenha.TabIndex = 3;

//

// labelSenha

//

this.labelSenha.AutoSize = true;

this.labelSenha.Location = new System.Drawing.Point(162, 74);

this.labelSenha.Name = "labelSenha";

this.labelSenha.Size = new System.Drawing.Size(42, 15);

this.labelSenha.TabIndex = 2;
```

```
this.labelSenha.Text = "Senha:";

//

// pictureBox1

//

this.pictureBox1.Image = global::Oraculo.Properties.Resources.logo;

this.pictureBox1.Location = new System.Drawing.Point(12, 14);

this.pictureBox1.Name = "pictureBox1";

this.pictureBox1.Size = new System.Drawing.Size(125, 130);

this.pictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;

this.pictureBox1.TabIndex = 17;

this.pictureBox1.TabStop = false;

//

// LoginForm

//

this.AcceptButton = this.OKButton;

this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.BackColor = System.Drawing.Color.WhiteSmoke;

this.ClientSize = new System.Drawing.Size(386, 206);

this.Controls.Add(this.pictureBox1);
```

```
this.Controls.Add(this.textBoxSenha);

this.Controls.Add(this.labelSenha);

this.Controls.Add(this.textBoxUsuario);

this.Controls.Add(this.labelUsuario);

this.Controls.Add(this.CancelButton);

this.Controls.Add(this.OKButton);

this.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog;

this.MaximizeBox = false;

this.MinimizeBox = false;

this.Name = "LoginForm";

this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;

this.Text = "Oráculo Estatístico";

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();

this.ResumeLayout(false);

this.PerformLayout();

}
```

```
#endregion
```

```
protected System.Windows.Forms.Button CancelButton;
```

```
protected System.Windows.Forms.Button OKButton;
```

```
private System.Windows.Forms.Label labelUsuario;
```

```
private System.Windows.Forms.TextBox textBoxUsuario;
```

```
private System.Windows.Forms.TextBox textBoxSenha;
```

```
private System.Windows.Forms.Label labelSenha;
```

```
private System.Windows.Forms.PictureBox pictureBox1;
```

```
}
```

```
}
```

```
Oraculo\Screens\MainForm.cs
```

```
?using System;
```

```
using System.Collections.Generic;
```

```
using System.ComponentModel;
```

```
using System.Data;
```

```
using System.Drawing;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;

using System.Windows.Forms;

using Oraculo.Telas;

using System.Data.Common;

using Oraculo.Global;

using Oraculo.Screens;

namespace Oraculo

{

    public partial class MainForm : Form

    {

        #region Fields

        private FormWindowState LastWindowState = FormWindowState.Normal;

        #endregion

        #region Properties

        public DataTable Data
```

```
{  
  
    get  
  
    {  
  
        return this.MainFormDataGridView.Data;  
  
    }  
  
    set  
  
    {  
  
        this.MainFormDataGridView.Data = value;  
  
    }  
  
}
```

```
public bool HasData
```

```
{  
  
    get  
  
    {  
  
        return MainFormDataGridView.HasData;  
  
    }  
  
}
```

```
#endregion
```

#region Methods

```
private void PlayYouTubeVideo(WebBrowser browser, string video)
```

```
{
```

```
    string html = "<html><head>";
```

```
    html += "<meta content='IE=Edge' http-equiv='X-UA-Compatible'/>";
```

```
    //html += "<iframe id='video' src= 'https://www.youtube.com/embed/{0}' width='{1}'  
height='{2}' frameborder='0' allowfullscreen='false' allowtransparency='true'  
style='background: #000000;'></iframe>";
```

```
    html += "<iframe id='video' src= 'https://www.youtube.com/embed/{0}'  
frameborder='0' allowfullscreen='false' allowtransparency='true' style='background:  
#000000;'></iframe>";
```

```
    html += "<body style='background-color:#000000; margin:0px;'></html>";
```

```
    browser.ScriptErrorsSuppressed = true;
```

```
    browser.ScrollBarsEnabled = false;
```

```
    browser.DocumentText = string.Format(html, video, browser.Width, browser.Height);
```

```
}
```

```
private void PlayYouTubeVideoFullScreen(string video)
{
    MediaPlayer tela = new MediaPlayer();

    tela.PlayYouTubeVideo(tela.WebBrowserVideo, video);

    tela.ShowDialog(this);
}
```

```
private void ShowModule(int module = 0)
{
    //oculta todos

    HomePanel.Visible = false;

    ImportPanel.Visible = false;

    CSVWizard.Visible = false;

    ExcelWizard.Visible = false;

    MainFormDataView.Visible = false;

    AnalyzeData.Visible = false;

    about1.Visible = false;

    reportViewer1.Visible = false;

    //exibe conforme o módulo escolhido no treeview
```



```
switch (module)
{
    case 0: //Apresentação

        HomePanel.Visible = true;

        break;

    case 1: //Dados

        if (!this.HasData)
        {

            //importar

            ImportPanel.Visible = true;

            //PlayYouTubeVideo(WebBrowserImportPanel, "c_jpWdoeQzw");

            //PlayYouTubeVideoFullScreen("c_jpWdoeQzw");

            break;

        }

        else

        {

            //abre para visualizar/digitar os dados

            MainFormDataView.Visible = true;

            break;

        }
}
```

case 2: //importar

ImportPanel.Visible = true;

//PlayYouTubeVideo(WebBrowserImportPanel, "c_jpWdoeQzw");

//PlayYouTubeVideoFullScreen("c_jpWdoeQzw");

break;

case 3: //csv

CSVWizard.Visible = true;

CSVWizard.ShowStep(1);

break;

case 4: //xls

ExcelWizard.Visible = true;

ExcelWizard.ShowStep(1);

break;

case 5: //digitar

MainFormDataView.Visible = true;

break;

case 6: //analisar

if (!this.HasData)

{

//clica no menu no item dados

```
sideNavigationMenu1.SelectMenuItemByName("Data");

break;

}

else

{

    //abre tela para analisar

    AnalyzeData.UpdateData(MainFormDataView.Data);

    switch (AnalyzeData.DataArrange)

    {

        case 0: //não definido

            AnalyzeData.ShowStep(1);

            break;

        case 1: //sem transformação

            AnalyzeData.ShowStep(2);

            break;

        case 2: //com transformação

            AnalyzeData.ShowStep(3);

            break;

    }

}
```

```
AnalyzeData.Visible = true;

break;

}

case 7: //relatório

if (!this.HasData)

{

//clica no menu no item dados

sideNavigationMenu1.SelectMenuItemByName("Data");

break;

}

else

{

//abre tela para analisar;

reportViewer1.Visible = true;

break;

}

case 8: //about

about1.Visible = true;

break;
```

```

    }
}

public MainForm()
{
    InitializeComponent();

    ShowModule(0);

    //User.CheckLogin();
}

#endregion

private void TelaPrincipalForm_Load(object sender, EventArgs e)
{
    sideNavigationMenu1.SelectMenuItemByName("Home");
}

private void ImportDataButton_Click(object sender, EventArgs e)
{
    sideNavigationMenu1.SelectMenuItemByName("Data");
}

```

```
}
```

```
private void TypeDataButton_Click(object sender, EventArgs e)
```

```
{
```

```
    ShowModule(5);
```

```
}
```

```
private void CSVWizard_AfterImportData(DataTable data)
```

```
{
```

```
    this.Data = data;
```

```
    ShowModule(1);
```

```
}
```

```
private void ExcelWizard_AfterImportData(DataTable data)
```

```
{
```

```
    this.Data = data;
```

```
    this.AnalyzeData.ClearData();
```

```
    ShowModule(1);
```

```
}
```

```
private void ImportFromXLSDaDataButton_Click(object sender, EventArgs e)

{

    //sideNavigationMenu1.SelectMenuItemByName("Data");

    ShowModule(4);

}
```

```
private void MainFormDataView_AfterClickAnalyze(DataTable data)

{

    sideNavigationMenu1.SelectMenuItemByName("Analyze");

}
```

```
private void WelcomeNextButton_Click(object sender, EventArgs e)

{

    if (HasData)

    {

        if (MessageBox.Show(this, "Você já informou dados para a análise. Deseja  
continuar usando os dados?", "Confirmação de exclusão", MessageBoxButtons.YesNo,  
MessageBoxIcon.Question) == DialogResult.No)

        {

            //limpa os dados

            this.Data = null;

        }

    }

}
```

```

    }

}

//mostra a tela para importar (se não tiver dados) ou digitar/alterar os dados

sideNavigationMenu1.SelectMenuItemByName("Data"); //importar xls

}

private void MainForm_Resize(object sender, EventArgs e)

{

    if (WindowState != LastWindowState)

    {

        LastWindowState = WindowState;

    }

    if (MainFormDataView.Visible)

    {

        //MainFormDataView.Refresh();

        //MainFormDataView.RefreshDataGrid();

    }

}

```



```
private void sideNavigationMenu1_SelectedItemChanged()
{
    switch (sideNavigationMenu1.SelectedItem.Name)
    {
        case "Home":
            ShowModule(0);
            break;

        case "Data":
            ShowModule(1);
            break;

        case "Analyze":
            ShowModule(6);
            break;

        case "Report":
            ShowModule(7);
            break;
    }
}
```

```
        case "About":  
            ShowModule(8);  
            break;  
        }  
    }  
  
private void ImportFromCSVDataButton_Click(object sender, EventArgs e)  
  
    {  
        ShowModule(3);  
    }  
  
private void sideNavigationMenu1_Load(object sender, EventArgs e)  
  
    {  
  
    }  
  
private void AnalyzeData_ReportViewerFired(string sql)  
  
    {  
        reportViewer1.ShowHTMLReport(sql);  
    }  
}
```

```
        sideNavigationMenu1.SelectMenuItemByName("Report");
    }

    private void MainFormDataView_Resize(object sender, EventArgs e)
    {
        //MainFormDataView.RefreshDataGrid();
    }

    private void MainForm_Layout(object sender, LayoutEventArgs e)
    {
        if (MainFormDataView.Visible) MainFormDataView.RefreshDataGrid();
    }

    private void VideoPictureBox_Click(object sender, EventArgs e)
    {
        //PlayYouTubeVideoFullScreen("c_jpWdoeQzw");

        System.Diagnostics.Process.Start("https://www.youtube.com/watch?v=l0MzREuu4eI");
    }
}
```

```
}
```

Oraculo\Screens\MainForm.Designer.cs

```
?namespace Oraculo
```

```
{
```

```
    partial class MainForm
```

```
    {
```

```
        /// <summary>
```

```
        /// Variável de designer necessária.
```

```
        /// </summary>
```

```
        private System.ComponentModel.IContainer components = null;
```

```
        /// <summary>
```

```
        /// Limpar os recursos que estão sendo usados.
```

```
        /// </summary>
```

```
        /// <param name="disposing">true se for necessário descartar os recursos gerenciados;  
        caso contrário, false.</param>
```

```
        protected override void Dispose(bool disposing)
```

```
        {
```

```
            if (disposing && (components != null))
```

```
{  
  
    components.Dispose();  
  
}  
  
base.Dispose(disposing);  
  
}
```

#region Código gerado pelo Windows Form Designer

/// <summary>

/// Método necessário para suporte ao Designer - não modifique

/// o conteúdo deste método com o editor de código.

/// </summary>

```
private void InitializeComponent()
```

```
{
```

```
    this.components = new System.ComponentModel.Container();
```

```
    System.ComponentModel.ComponentResourceManager resources = new  
System.ComponentModel.ComponentResourceManager(typeof(MainForm));
```

```
    this.HomePanel = new System.Windows.Forms.Panel();
```

```
    this.pictureBox1 = new System.Windows.Forms.PictureBox();
```

```
    this>WelcomeNextButton = new System.Windows.Forms.Button();
```

```
this.LabelWelcome = new System.Windows.Forms.Label();

this.LabelOraculoDescription = new System.Windows.Forms.Label();

this.ImportPanel = new System.Windows.Forms.Panel();

this.VideoPictureBox = new System.Windows.Forms.PictureBox();

this.label1 = new System.Windows.Forms.Label();

this.WebBrowserImportPanel = new System.Windows.Forms.WebBrowser();

this.ImportPanelLabel2 = new System.Windows.Forms.Label();

this.ImportFromXLSDataButton = new System.Windows.Forms.Button();

this.TypeDataButton = new System.Windows.Forms.Button();

this.ImportFromCSVDataButton = new System.Windows.Forms.Button();

this.ImportPanelLabel1 = new System.Windows.Forms.Label();

this.toolTip1 = new System.Windows.Forms.ToolTip(this.components);

this.panel2 = new System.Windows.Forms.Panel();

this.AnalyzeData = new Oraculo.Telas.Analyze();

this.reportViewer1 = new Oraculo.Screens.ReportViewer();

this.about1 = new Oraculo.Screens.About();

this.MainFormDataView = new Oraculo.Telas.DataView();

this.ExcelWizard = new Oraculo.Telas.ExcelFileImportWizardControl();

this.CSVWizard = new Oraculo.Telas.CSVFileImportWizardControl();

this.tableLayoutPanel1 = new System.Windows.Forms.TableLayoutPanel();
```

```

this.sideNavigationMenu1 = new Oraculo.Global.SideNavigationMenu();

this.HomePanel.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();

this.ImportPanel.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.VideoPictureBox)).BeginInit();

this.panel2.SuspendLayout();

this.tableLayoutPanel1.SuspendLayout();

this.SuspendLayout();

//

// HomePanel

//

this.HomePanel.BackColor = System.Drawing.Color.White;

this.HomePanel.Controls.Add(this.pictureBox1);

this.HomePanel.Controls.Add(this.WelcomeNextButton);

this.HomePanel.Controls.Add(this.LabelWelcome);

this.HomePanel.Controls.Add(this.LabelOraculoDescription);

this.HomePanel.Dock = System.Windows.Forms.DockStyle.Fill;

this.HomePanel.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.HomePanel.Location = new System.Drawing.Point(0, 0);

```

```
this.HomePanel.Margin = new System.Windows.Forms.Padding(0);

this.HomePanel.Name = "HomePanel";

this.HomePanel.Size = new System.Drawing.Size(544, 641);

this.HomePanel.TabIndex = 17;

this.HomePanel.Visible = false;

//

// pictureBox1

//

this.pictureBox1.Anchor = System.Windows.Forms.AnchorStyles.None;

this.pictureBox1.Image = global::Oraculo.Properties.Resources.logo;

this.pictureBox1.Location = new System.Drawing.Point(177, 77);

this.pictureBox1.Name = "pictureBox1";

this.pictureBox1.Size = new System.Drawing.Size(190, 201);

this.pictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;

this.pictureBox1.TabIndex = 12;

this.pictureBox1.TabStop = false;

//

// WelcomeNextButton

//

this>WelcomeNextButton.Anchor = System.Windows.Forms.AnchorStyles.None;
```



```

this>WelcomeNextButton.BackColor = System.Drawing.Color.Gainsboro;

this>WelcomeNextButton.FlatAppearance.BorderColor =
System.Drawing.Color.Gainsboro;

this>WelcomeNextButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this>WelcomeNextButton.Image = global::Oraculo.Properties.Resources.next_flat;

this>WelcomeNextButton.ImageAlign =
System.Drawing.ContentAlignment.MiddleRight;

this>WelcomeNextButton.Location = new System.Drawing.Point(239, 429);

this>WelcomeNextButton.Margin = new System.Windows.Forms.Padding(3, 4, 3, 4);

this>WelcomeNextButton.Name = "WelcomeNextButton";

this>WelcomeNextButton.Size = new System.Drawing.Size(66, 30);

this>WelcomeNextButton.TabIndex = 5;

this>WelcomeNextButton.Text = "Iniciar";

this>WelcomeNextButton.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft;

this.toolTip1.SetToolTip(this>WelcomeNextButton, "Próxima etapa");

this>WelcomeNextButton.UseVisualStyleBackColor = false;

this>WelcomeNextButton.Click += new
System.EventHandler(this>WelcomeNextButton_Click);

//

// LabelWelcome

```

```
//  
  
this.LabelWelcome.Anchor = System.Windows.Forms.AnchorStyles.None;  
  
this.LabelWelcome.AutoSize = true;  
  
this.LabelWelcome.Font = new System.Drawing.Font("Segoe UI", 14.25F,  
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));  
  
this.LabelWelcome.Location = new System.Drawing.Point(97, 308);  
  
this.LabelWelcome.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);  
  
this.LabelWelcome.Name = "LabelWelcome";  
  
this.LabelWelcome.Size = new System.Drawing.Size(333, 25);  
  
this.LabelWelcome.TabIndex = 4;  
  
this.LabelWelcome.Text = "Seja bem vindo ao Oráculo Estatístico!";  
  
//  
  
// LabelOraculoDescription  
  
//  
  
this.LabelOraculoDescription.Anchor = System.Windows.Forms.AnchorStyles.None;  
  
this.LabelOraculoDescription.Font = new System.Drawing.Font("Segoe UI", 9F,  
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));  
  
this.LabelOraculoDescription.Location = new System.Drawing.Point(62, 362);  
  
this.LabelOraculoDescription.Margin = new System.Windows.Forms.Padding(2, 0, 2,  
0);  
  
this.LabelOraculoDescription.Name = "LabelOraculoDescription";
```

```

this.LabelOraculoDescription.Size = new System.Drawing.Size(421, 42);

this.LabelOraculoDescription.TabIndex = 0;

this.LabelOraculoDescription.Text = "O Oráculo Estatístico é um software destinado a
realizar testes de Hipótese de ma" +

"neira ágil e fácil, gerando relatórios explicativos.";

//

// ImportPanel

//

this.ImportPanel.BackColor = System.Drawing.SystemColors.Control;

this.ImportPanel.Controls.Add(this.VideoPictureBox);

this.ImportPanel.Controls.Add(this.label1);

this.ImportPanel.Controls.Add(this.WebBrowserImportPanel);

this.ImportPanel.Controls.Add(this.ImportPanelLabel2);

this.ImportPanel.Controls.Add(this.ImportFromXLSDDataButton);

this.ImportPanel.Controls.Add(this.TypeDataButton);

this.ImportPanel.Controls.Add(this.ImportFromCSVDDataButton);

this.ImportPanel.Controls.Add(this.ImportPanelLabel1);

this.ImportPanel.Dock = System.Windows.Forms.DockStyle.Fill;

this.ImportPanel.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.ImportPanel.Location = new System.Drawing.Point(0, 0);

```

```

this.ImportPanel.Margin = new System.Windows.Forms.Padding(2);

this.ImportPanel.Name = "ImportPanel";

this.ImportPanel.Size = new System.Drawing.Size(544, 641);

this.ImportPanel.TabIndex = 0;

this.ImportPanel.Visible = false;

//

// VideoPictureBox

//

this.VideoPictureBox.Anchor = System.Windows.Forms.AnchorStyles.None;

this.VideoPictureBox.Image =
((System.Drawing.Image)(resources.GetObject("VideoPictureBox.Image")));

this.VideoPictureBox.Location = new System.Drawing.Point(70, 274);

this.VideoPictureBox.Name = "VideoPictureBox";

this.VideoPictureBox.Size = new System.Drawing.Size(416, 235);

this.VideoPictureBox.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.Zoom;

this.VideoPictureBox.TabIndex = 7;

this.VideoPictureBox.TabStop = false;

this.VideoPictureBox.Click += new
System.EventHandler(this.VideoPictureBox_Click);

//

```

```

// label1

//

this.label1.Anchor = System.Windows.Forms.AnchorStyles.None;

this.label1.AutoSize = true;

this.label1.Font = new System.Drawing.Font("Segoe UI", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.label1.Location = new System.Drawing.Point(108, 47);

this.label1.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(329, 25);

this.label1.TabIndex = 6;

this.label1.Text = "Não temos dados para analisar ainda.";

//

// WebBrowserImportPanel

//

this.WebBrowserImportPanel.Anchor = System.Windows.Forms.AnchorStyles.None;

this.WebBrowserImportPanel.Location = new System.Drawing.Point(70, 274);

this.WebBrowserImportPanel.MinimumSize = new System.Drawing.Size(20, 20);

this.WebBrowserImportPanel.Name = "WebBrowserImportPanel";

this.WebBrowserImportPanel.Size = new System.Drawing.Size(411, 235);

```

```

this.WebBrowserImportPanel.TabIndex = 5;

//

// ImportPanelLabel2

//

this.ImportPanelLabel2.Anchor = System.Windows.Forms.AnchorStyles.None;

this.ImportPanelLabel2.AutoSize = true;

this.ImportPanelLabel2.Location = new System.Drawing.Point(84, 241);

this.ImportPanelLabel2.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);

this.ImportPanelLabel2.Name = "ImportPanelLabel2";

this.ImportPanelLabel2.Size = new System.Drawing.Size(383, 15);

this.ImportPanelLabel2.TabIndex = 4;

this.ImportPanelLabel2.Text = "Se precisar, assista ao vídeo que demonstra o uso do
Oráculo Estatístico.";

//

// ImportFromXLSDDataButton

//

this.ImportFromXLSDDataButton.Anchor =
System.Windows.Forms.AnchorStyles.None;

this.ImportFromXLSDDataButton.BackColor = System.Drawing.Color.Gainsboro;

this.ImportFromXLSDDataButton.FlatAppearance.BorderColor =
System.Drawing.Color.Gainsboro;

```

```

this.ImportFromXLSDDataButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.ImportFromXLSDDataButton.Image =
global::Oraculo.Properties.Resources.import_excel_flat;

this.ImportFromXLSDDataButton.ImageAlign =
System.Drawing.ContentAlignment.MiddleLeft;

this.ImportFromXLSDDataButton.Location = new System.Drawing.Point(212, 157);

this.ImportFromXLSDDataButton.Margin = new System.Windows.Forms.Padding(2);

this.ImportFromXLSDDataButton.Name = "ImportFromXLSDDataButton";

this.ImportFromXLSDDataButton.Size = new System.Drawing.Size(131, 28);

this.ImportFromXLSDDataButton.TabIndex = 2;

this.ImportFromXLSDDataButton.Text = "Importar do Excel";

this.ImportFromXLSDDataButton.TextAlign =
System.Drawing.ContentAlignment.MiddleRight;

this.toolTip1.SetToolTip(this.ImportFromXLSDDataButton, "Clique para importar de
arquivo do Excel (XLS ou XLSX)");

this.ImportFromXLSDDataButton.UseVisualStyleBackColor = false;

this.ImportFromXLSDDataButton.Click += new
System.EventHandler(this.ImportFromXLSDDataButton_Click);

//

// TypeDataButton

//

this.TypeDataButton.Anchor = System.Windows.Forms.AnchorStyles.None;

```

```

this.TypeDataButton.BackColor = System.Drawing.Color.Gainsboro;

this.TypeDataButton.FlatAppearance.BorderColor =
System.Drawing.Color.Gainsboro;

this.TypeDataButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.TypeDataButton.Image = global::Oraculo.Properties.Resources.database_flat;

this.TypeDataButton.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.TypeDataButton.Location = new System.Drawing.Point(360, 157);

this.TypeDataButton.Margin = new System.Windows.Forms.Padding(2);

this.TypeDataButton.Name = "TypeDataButton";

this.TypeDataButton.Size = new System.Drawing.Size(123, 28);

this.TypeDataButton.TabIndex = 3;

this.TypeDataButton.Text = "Digitar os dados";

this.TypeDataButton.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

this.toolTip1.SetToolTip(this.TypeDataButton, "Clique para digitar os dados
diretamente");

this.TypeDataButton.UseVisualStyleBackColor = false;

this.TypeDataButton.Click += new
System.EventHandler(this.TypeDataButton_Click);

//

// ImportFromCSVDataButton

//

```



```
this.ImportFromCSVDataButton.Anchor =  
System.Windows.Forms.AnchorStyles.None;  
  
this.ImportFromCSVDataButton.BackColor = System.Drawing.Color.Gainsboro;  
  
this.ImportFromCSVDataButton.FlatAppearance.BorderColor =  
System.Drawing.Color.Gainsboro;  
  
this.ImportFromCSVDataButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat;  
  
this.ImportFromCSVDataButton.Image =  
global::Oraculo.Properties.Resources.import_csv_flat;  
  
this.ImportFromCSVDataButton.ImageAlign =  
System.Drawing.ContentAlignment.MiddleLeft;  
  
this.ImportFromCSVDataButton.Location = new System.Drawing.Point(70, 157);  
  
this.ImportFromCSVDataButton.Margin = new System.Windows.Forms.Padding(2);  
  
this.ImportFromCSVDataButton.Name = "ImportFromCSVDataButton";  
  
this.ImportFromCSVDataButton.Size = new System.Drawing.Size(124, 28);  
  
this.ImportFromCSVDataButton.TabIndex = 1;  
  
this.ImportFromCSVDataButton.Text = "Importar de CSV";  
  
this.ImportFromCSVDataButton.TextAlign =  
System.Drawing.ContentAlignment.MiddleRight;  
  
this.toolTip1.SetToolTip(this.ImportFromCSVDataButton, "Clique para importar de  
arquivo texto separado por ponto-e-vírgula (CSV)");  
  
this.ImportFromCSVDataButton.UseVisualStyleBackColor = false;  
  
this.ImportFromCSVDataButton.Click += new  
System.EventHandler(this.ImportFromCSVDataButton_Click);
```

```
//  
  
// ImportPanelLabel1  
  
//  
  
this.ImportPanelLabel1.Anchor = System.Windows.Forms.AnchorStyles.None;  
  
this.ImportPanelLabel1.AutoSize = true;  
  
this.ImportPanelLabel1.Location = new System.Drawing.Point(123, 112);  
  
this.ImportPanelLabel1.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);  
  
this.ImportPanelLabel1.Name = "ImportPanelLabel1";  
  
this.ImportPanelLabel1.Size = new System.Drawing.Size(306, 15);  
  
this.ImportPanelLabel1.TabIndex = 0;  
  
this.ImportPanelLabel1.Text = "Para começar, você precisa importar ou digitar os  
dados.";  
  
//  
  
// panel2  
  
//  
  
this.panel2.BackColor = System.Drawing.Color.WhiteSmoke;  
  
this.panel2.Controls.Add(this.ImportPanel);  
  
this.panel2.Controls.Add(this.HomePanel);  
  
this.panel2.Controls.Add(this.AnalyzeData);  
  
this.panel2.Controls.Add(this.reportViewer1);
```

```
this.panel2.Controls.Add(this.about1);

this.panel2.Controls.Add(this.MainFormDataView);

this.panel2.Controls.Add(this.ExcelWizard);

this.panel2.Controls.Add(this.CSVWizard);

this.panel2.Dock = System.Windows.Forms.DockStyle.Fill;

this.panel2.Location = new System.Drawing.Point(240, 0);

this.panel2.Margin = new System.Windows.Forms.Padding(0);

this.panel2.Name = "panel2";

this.panel2.Size = new System.Drawing.Size(544, 641);

this.panel2.TabIndex = 1;

//

// AnalyzeData

//

this.AnalyzeData.BackColor = System.Drawing.Color.White;

this.AnalyzeData.Data = null;

this.AnalyzeData.DataArrange = 0;

this.AnalyzeData.DataBackup = null;

this.AnalyzeData.Dock = System.Windows.Forms.DockStyle.Fill;

this.AnalyzeData.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));
```

```

this.AnalyzeData.Location = new System.Drawing.Point(0, 0);

this.AnalyzeData.Margin = new System.Windows.Forms.Padding(0);

this.AnalyzeData.Name = "AnalyzeData";

this.AnalyzeData.Size = new System.Drawing.Size(544, 641);

this.AnalyzeData.Step = 1;

this.AnalyzeData.TabIndex = 16;

this.AnalyzeData.Title = "Analisar";

this.AnalyzeData.TotalSteps = 5;

this.AnalyzeData.ReportViewerFired += new
Oraculo.Telas.Analyze.OnReportViewerFiredEvent(this.AnalyzeData_ReportViewerFired);

//

// reportViewer1

//

this.reportViewer1.BackColor = System.Drawing.Color.White;

this.reportViewer1.Dock = System.Windows.Forms.DockStyle.Fill;

this.reportViewer1.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.reportViewer1.Location = new System.Drawing.Point(0, 0);

this.reportViewer1.Name = "reportViewer1";

this.reportViewer1.Size = new System.Drawing.Size(544, 641);

this.reportViewer1.TabIndex = 19;

```

```
//  
  
// about1  
  
//  
  
this.about1.BackColor = System.Drawing.Color.White;  
  
this.about1.Dock = System.Windows.Forms.DockStyle.Fill;  
  
this.about1.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,  
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));  
  
this.about1.Location = new System.Drawing.Point(0, 0);  
  
this.about1.Margin = new System.Windows.Forms.Padding(0);  
  
this.about1.Name = "about1";  
  
this.about1.Size = new System.Drawing.Size(544, 641);  
  
this.about1.TabIndex = 18;  
  
//  
  
// MainFormDataView  
  
//  
  
this.MainFormDataView.AutoScroll = true;  
  
this.MainFormDataView.BackColor = System.Drawing.Color.White;  
  
this.MainFormDataView.Data = null;  
  
this.MainFormDataView.Dock = System.Windows.Forms.DockStyle.Fill;  
  
this.MainFormDataView.Font = new System.Drawing.Font("Segoe UI", 8.25F,  
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));
```

```

this.MainFormDataView.Location = new System.Drawing.Point(0, 0);

this.MainFormDataView.Margin = new System.Windows.Forms.Padding(0);

this.MainFormDataView.Name = "MainFormDataView";

this.MainFormDataView.SelectedColumn = 0;

this.MainFormDataView.Size = new System.Drawing.Size(544, 641);

this.MainFormDataView.TabIndex = 14;

this.MainFormDataView.AfterClickAnalyze += new
Oraculo.Telas.EventHandler(this.MainFormDataView_AfterClickAnalyze);

this.MainFormDataView.Resize += new
System.EventHandler(this.MainFormDataView_Resize);

//

// ExcelWizard

//

this.ExcelWizard.BackColor = System.Drawing.Color.White;

this.ExcelWizard.Data = null;

this.ExcelWizard.Dock = System.Windows.Forms.DockStyle.Fill;

this.ExcelWizard.FilePath = "";

this.ExcelWizard.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.ExcelWizard.Location = new System.Drawing.Point(0, 0);

this.ExcelWizard.Margin = new System.Windows.Forms.Padding(0);

```

```

this.ExcelWizard.Name = "ExcelWizard";

this.ExcelWizard.SelectedSheet = "";

this.ExcelWizard.ShowColumnTitles = false;

this.ExcelWizard.Size = new System.Drawing.Size(544, 641);

this.ExcelWizard.Step = 1;

this.ExcelWizard.TabIndex = 15;

this.ExcelWizard.Title = "Importar arquivos do Excel (*.xlsx)";

this.ExcelWizard.TotalSteps = 3;

this.ExcelWizard.AfterImportData += new
Oraculo.Telas.EventHandler(this.ExcelWizard_AfterImportData);

//

// CSVWizard

//

this.CSVWizard.BackColor = System.Drawing.Color.White;

this.CSVWizard.Data = null;

this.CSVWizard.Dock = System.Windows.Forms.DockStyle.Fill;

this.CSVWizard.FilePath = "";

this.CSVWizard.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.CSVWizard.Location = new System.Drawing.Point(0, 0);

this.CSVWizard.Margin = new System.Windows.Forms.Padding(0);

```

```

this.CSVWizard.Name = "CSVWizard";

this.CSVWizard.SelectedSheet = "";

this.CSVWizard.ShowColumnTitles = false;

this.CSVWizard.Size = new System.Drawing.Size(544, 641);

this.CSVWizard.Step = 1;

this.CSVWizard.TabIndex = 13;

this.CSVWizard.Title = "Importar arquivo texto (*.txt; *.csv)";

this.CSVWizard.TotalSteps = 2;

this.CSVWizard.AfterImportData += new
Oraculo.Telas.EventHandler(this.CSVWizard_AfterImportData);

//

// tableLayoutPanel1

//

this.tableLayoutPanel1.ColumnCount = 2;

this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle());

this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle());

this.tableLayoutPanel1.Controls.Add(this.sideNavigationMenu1, 0, 0);

this.tableLayoutPanel1.Controls.Add(this.panel2, 1, 0);

this.tableLayoutPanel1.Dock = System.Windows.Forms.DockStyle.Fill;

```



```
this.tableLayoutPanel1.Location = new System.Drawing.Point(0, 0);

this.tableLayoutPanel1.Margin = new System.Windows.Forms.Padding(0);

this.tableLayoutPanel1.Name = "tableLayoutPanel1";

this.tableLayoutPanel1.RowCount = 1;

this.tableLayoutPanel1.RowStyle.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 641F));

this.tableLayoutPanel1.RowStyle.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 641F));

this.tableLayoutPanel1.RowStyle.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 641F));

this.tableLayoutPanel1.Size = new System.Drawing.Size(784, 641);

this.tableLayoutPanel1.TabIndex = 2;

//

// sideNavigationMenu1

//

this.sideNavigationMenu1.Dock = System.Windows.Forms.DockStyle.Left;

this.sideNavigationMenu1.Location = new System.Drawing.Point(0, 0);

this.sideNavigationMenu1.Margin = new System.Windows.Forms.Padding(0);

this.sideNavigationMenu1.Name = "sideNavigationMenu1";

this.sideNavigationMenu1.SelectedItem = null;

this.sideNavigationMenu1.Size = new System.Drawing.Size(240, 641);
```

```

this.sideNavigationMenu1.TabIndex = 0;

this.sideNavigationMenu1.SelectedItemChanged += new
Oraculo.Global.SideNavigationMenu.SelectedItemChangedEvent(this.sideNavigationMenu1_
SelectedItemChanged);

this.sideNavigationMenu1.Load += new
System.EventHandler(this.sideNavigationMenu1_Load);

//

// MainForm

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.BackColor = System.Drawing.Color.White;

this.ClientSize = new System.Drawing.Size(784, 641);

this.Controls.Add(this.tableLayoutPanel1);

this.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));

this.Margin = new System.Windows.Forms.Padding(2);

this.MinimumSize = new System.Drawing.Size(800, 680);

this.Name = "MainForm";

this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;

```

```

this.Text = "Oráculo Estatístico";

this.WindowState = System.Windows.Forms.FormWindowState.Maximized;

this.Load += new System.EventHandler(this.TelaPrincipalForm_Load);

this.Layout += new
System.Windows.Forms.LayoutEventHandler(this.MainForm_Layout);

this.Resize += new System.EventHandler(this.MainForm_Resize);

this.HomePanel.ResumeLayout(false);

this.HomePanel.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();

this.ImportPanel.ResumeLayout(false);

this.ImportPanel.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.VideoPictureBox)).EndInit();

this.panel2.ResumeLayout(false);

this.tableLayoutPanel1.ResumeLayout(false);

this.ResumeLayout(false);

}

#endregion

private System.Windows.Forms.Panel ImportPanel;

```

```
private System.Windows.Forms.Button TypeDataButton;

private System.Windows.Forms.Button ImportFromCSVDataButton;

private System.Windows.Forms.Label ImportPanelLabel1;

private Telas.CSVFileImportWizardControl CSVWizard;

private Telas.DataView MainFormDataView;

private Telas.ExcelFileImportWizardControl ExcelWizard;

private Telas.Analyze AnalyzeData;

private System.Windows.Forms.Button ImportFromXLSDataButton;

private System.Windows.Forms.ToolTip toolTip1;

private System.Windows.Forms.Panel HomePanel;

private System.Windows.Forms.Label LabelWelcome;

private System.Windows.Forms.Label LabelOraculoDescription;

protected System.Windows.Forms.Button WelcomeNextButton;

private System.Windows.Forms.Label ImportPanelLabel2;

private System.Windows.Forms.WebBrowser WebBrowserImportPanel;

private Global.SideNavigationMenu sideNavigationMenu1;

private System.Windows.Forms.PictureBox pictureBox1;

private System.Windows.Forms.Panel panel2;

private System.Windows.Forms.TableLayoutPanelPanel tableLayoutPanel1;

private System.Windows.Forms.Label label1;
```

```
private Screens.About about1;  
  
private Screens.ReportViewer reportViewer1;  
  
private System.Windows.Forms.PictureBox VideoPictureBox;  
  
}  
  
}
```

Oraculo\Screens\RemoveColumnsForm.cs

```
?using System;  
  
using System.Collections.Generic;  
  
using System.ComponentModel;  
  
using System.Data;  
  
using System.Drawing;  
  
using System.Text;  
  
using System.Windows.Forms;  
  
namespace Oraculo.Telas  
{  
  
    public partial class RemoveColumnsForm : Oraculo.Global.EditForm
```

```
{

public RemoveColumnsForm()

{

    InitializeComponent();

}

public List<string> GetSelectedColumns()

{

    List<string> list = new List<string>();

    foreach (var item in ColumnsListBox.SelectedItems)

    {

        list.Add(item.ToString());

    }

    return list;

}

protected void Fill(DataTable data)

{

    ColumnsListBox.Items.Clear();
```

```
        foreach (DataColumn item in data.Columns)
        {
            ColumnsListBox.Items.Add(item.ColumnName);
        }

        ColumnsListBox.SelectedIndex = 0;
    }

    public void ShowInsertMode(DataTable data)
    {
        Clear();

        Fill(data);

        ShowDialog();
    }
}
}
```

Oraculo\Screens\RemoveColumnsForm.Designer.cs

```

?namespace Oraculo.Telas

{

    partial class RemoveColumnsForm

    {

        /// <summary>

        /// Required designer variable.

        /// </summary>

        private System.ComponentModel.IContainer components = null;

        /// <summary>

        /// Clean up any resources being used.

        /// </summary>

        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>

        protected override void Dispose(bool disposing)

        {

            if (disposing && (components != null))

            {

                components.Dispose();

            }

        }

    }

}

```



```

        base.Dispose(disposing);

    }

    #region Windows Form Designer generated code

    /// <summary>

    /// Required method for Designer support - do not modify

    /// the contents of this method with the code editor.

    /// </summary>

    private void InitializeComponent()

    {

        System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(RemoveColumnsForm));

        this.ColumnsLabel = new System.Windows.Forms.Label();

        this.ColumnsListBox = new System.Windows.Forms.ListBox();

        this.SuspendLayout();

        //

        // CancelButton

        //

        this.CancelButton.Location = new System.Drawing.Point(266, 409);

```

```
this.CancelButton.TabIndex = 3;

//

// OKButton

//

this.OKButton.Location = new System.Drawing.Point(165, 409);

this.OKButton.TabIndex = 2;

//

// ColumnsLabel

//

this.ColumnsLabel.AutoSize = true;

this.ColumnsLabel.Location = new System.Drawing.Point(29, 29);

this.ColumnsLabel.Margin = new System.Windows.Forms.Padding(20);

this.ColumnsLabel.Name = "ColumnsLabel";

this.ColumnsLabel.Size = new System.Drawing.Size(273, 15);

this.ColumnsLabel.TabIndex = 0;

this.ColumnsLabel.Text = "Selecione a(s) colunas(s) que você deseja remover:";

//

// ColumnsListBox

//
```

```

        this.ColumnsListBox.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)

        | System.Windows.Forms.AnchorStyles.Left)

        | System.Windows.Forms.AnchorStyles.Right))));

this.ColumnsListBox.FormattingEnabled = true;

this.ColumnsListBox.ItemHeight = 15;

this.ColumnsListBox.Location = new System.Drawing.Point(29, 57);

this.ColumnsListBox.Margin = new System.Windows.Forms.Padding(20);

this.ColumnsListBox.Name = "ColumnsListBox";

this.ColumnsListBox.ScrollAlwaysVisible = true;

this.ColumnsListBox.SelectionMode =
System.Windows.Forms.SelectionMode.MultiExtended;

this.ColumnsListBox.Size = new System.Drawing.Size(328, 334);

this.ColumnsListBox.TabIndex = 1;

//

// RemoveColumnsForm

//

this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);

this.ClientSize = new System.Drawing.Size(386, 466);

this.Controls.Add(this.ColumnsListBox);

```

```
this.Controls.Add(this.ColumnsLabel);

this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));

this.MinimumSize = new System.Drawing.Size(402, 505);

this.Name = "RemoveColumnsForm";

this.Text = "Remover coluna";

this.Controls.SetChildIndex(this.OKButton, 0);

this.Controls.SetChildIndex(this.CancelButton, 0);

this.Controls.SetChildIndex(this.ColumnsLabel, 0);

this.Controls.SetChildIndex(this.ColumnsListBox, 0);

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.Label ColumnsLabel;

private System.Windows.Forms.ListBox ColumnsListBox;

}

}
```

Oraculo\Screens\ReportViewer.cs

?using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Oraculo.Screens

{

public partial class ReportViewer : UserControl

{

public ReportViewer()

{

InitializeComponent();

```
}
```

```
public void ShowHTMLReport(string report)
```

```
{
```

```
    string html = "<html>";
```

```
    html += "<head>";
```

```
    html += "<meta content='IE=Edge' http-equiv='X-UA-Compatible'/>";
```

```
    html += "<style> body { font-family: Arial, Helvetica, sans-serif; padding: 20px; }  
table { border-collapse: collapse; } td, th { border: 1px solid #ddd; padding: 8px; } tr:nth-  
child(even) { background-color: #f2f2f2; } tr:hover { background-color: #ddd; } th {  
padding-top: 6px; padding-bottom: 6px; text-align: left; background-color: #2980b9;  
/*rgb(18, 86, 194);*/ color: white; } </style>"; //"<style> table, th, td { border: 1px solid  
black; border-collapse: collapse; }</style>";
```

```
    html += "</head>";
```

```
    html += "<body>";
```

```
    html += report;
```

```
    html += "<h3>Como citar?</h3><p>Silva, R. C. F.; Santos Junior, I. M.; Silveira, M.  
F.; Monteiro Junior, R. S. (2022). Oráculo Estatístico [Computer software].</p>";
```

```
    html += "</body>";
```

```
    html += "</html>";
```

```
        this.ResultWebBrowser.ScriptErrorsSuppressed = true;

        this.ResultWebBrowser.ScrollBarsEnabled = true;

        this.ResultWebBrowser.DocumentText = html;

        this.ResultWebBrowser.Visible = true;
    }
}
}
```

Oraculo\Screens\ReportViewer.Designer.cs

?

```
namespace Oraculo.Screens
```

```
{
```

```
    partial class ReportViewer
```

```
    {
```

```
        /// <summary>
```

```
        /// Variável de designer necessária.
```

```
        /// </summary>
```

```
        private System.ComponentModel.IContainer components = null;
```

```
/// <summary>

/// Limpar os recursos que estão sendo usados.

/// </summary>

/// <param name="disposing">true se for necessário descartar os recursos gerenciados;
caso contrário, false.</param>

protected override void Dispose(bool disposing)

{

    if (disposing && (components != null))

    {

        components.Dispose();

    }

    base.Dispose(disposing);

}

#region Código gerado pelo Designer de Componentes

/// <summary>

/// Método necessário para suporte ao Designer - não modifique

/// o conteúdo deste método com o editor de código.
```



```
/// </summary>

private void InitializeComponent()

{

    this.ResultWebBrowser = new System.Windows.Forms.WebBrowser();

    this.label1 = new System.Windows.Forms.Label();

    this.label2 = new System.Windows.Forms.Label();

    this.sectionHeader1 = new Oraculo.Global.SectionHeader();

    this.SuspendLayout();

    //

    // ResultWebBrowser

    //

    this.ResultWebBrowser.Dock = System.Windows.Forms.DockStyle.Fill;

    this.ResultWebBrowser.Location = new System.Drawing.Point(0, 119);

    this.ResultWebBrowser.Margin = new System.Windows.Forms.Padding(0);

    this.ResultWebBrowser.MinimumSize = new System.Drawing.Size(23, 23);

    this.ResultWebBrowser.Name = "ResultWebBrowser";

    this.ResultWebBrowser.Size = new System.Drawing.Size(640, 361);

    this.ResultWebBrowser.TabIndex = 7;

    this.ResultWebBrowser.Visible = false;

    //
```

```
// label1

//

this.label1.Anchor = System.Windows.Forms.AnchorStyles.None;

this.label1.AutoSize = true;

this.label1.Font = new System.Drawing.Font("Segoe UI", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.label1.Location = new System.Drawing.Point(129, 303);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(382, 25);

this.label1.TabIndex = 8;

this.label1.Text = "Não há resultados para mostrar no relatório";

//

// label2

//

this.label2.Anchor = System.Windows.Forms.AnchorStyles.None;

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(214, 338);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(216, 15);

this.label2.TabIndex = 9;
```

```
this.label2.Text = "Você precisa primeiro analisar os dados.";

//

// sectionHeader1

//

this.sectionHeader1.BreadCrumbText = "Início > Relatório";

this.sectionHeader1.Dock = System.Windows.Forms.DockStyle.Top;

this.sectionHeader1.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.sectionHeader1.Location = new System.Drawing.Point(0, 0);

this.sectionHeader1.Margin = new System.Windows.Forms.Padding(0);

this.sectionHeader1.Name = "sectionHeader1";

this.sectionHeader1.Size = new System.Drawing.Size(640, 119);

this.sectionHeader1.TabIndex = 0;

//

// ReportViewer

//

this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.BackColor = System.Drawing.Color.White;

this.Controls.Add(this.ResultWebBrowser);
```

```

        this.Controls.Add(this.label2);

        this.Controls.Add(this.label1);

        this.Controls.Add(this.sectionHeader1);

        this.Font = new System.Drawing.Font("Segoe UI", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

        this.Name = "ReportViewer";

        this.Size = new System.Drawing.Size(640, 480);

        this.ResumeLayout(false);

        this.PerformLayout();

    }

#endregion

private Global.SectionHeader sectionHeader1;

private System.Windows.Forms.WebBrowser ResultWebBrowser;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Label label2;

}

}

```

Oraculo\Screens\VideoPlayer.cs

?using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Oraculo.Screens

{

public partial class VideoPlayer : Form

{

public VideoPlayer()

{

InitializeComponent();

```
}
```

```
public void PlayYouTubeVideo(WebBrowser browser, string video)
```

```
{
```

```
    string html = "<html><head>";
```

```
    html += "<meta content='IE=Edge' http-equiv='X-UA-Compatible'/>";
```

```
    html += "<body style='background-color:#000000; margin:0px;'></html>";
```

```
    html += "<iframe width='{1}' height='{2}'  
src='https://www.youtube.com/embed/{0}?controls=0?autoplay=1&vq=hd720p'  
title='YouTube video player' frameborder='0' autoplay; allowfullscreen></iframe>";
```

```
    html += "</body></html>";
```

```
    browser.ScriptErrorsSuppressed = true;
```

```
    browser.ScrollBarsEnabled = false;
```

```
    int h = 720;
```

```
    int w = 1080;
```

```
    //w = browser.Width;
```

```
    //h = (int)(browser.Width * 0.5625f);
```

```
        browser.DocumentText = string.Format(html, video, w, h);  
    }  
}  
}
```

Oraculo\Screens\VideoPlayer.Designer.cs

?

```
namespace Oraculo.Screens
```

```
{
```

```
    partial class VideoPlayer
```

```
    {
```

```
        /// <summary>
```

```
        /// Required designer variable.
```

```
        /// </summary>
```

```
        private System.ComponentModel.IContainer components = null;
```

```
        /// <summary>
```

```
        /// Clean up any resources being used.
```

```
        /// </summary>
```

```
/// <param name="disposing">true if managed resources should be disposed; otherwise,  
false.</param>
```

```
protected override void Dispose(bool disposing)
```

```
{
```

```
    if (disposing && (components != null))
```

```
    {
```

```
        components.Dispose();
```

```
    }
```

```
    base.Dispose(disposing);
```

```
}
```

```
#region Windows Form Designer generated code
```

```
/// <summary>
```

```
/// Required method for Designer support - do not modify
```

```
/// the contents of this method with the code editor.
```

```
/// </summary>
```

```
private void InitializeComponent()
```

```
{
```

```
    this.WebBrowserVideo = new System.Windows.Forms.WebBrowser();
```



```
this.SuspendLayout();

//

// WebBrowserVideo

//

this.WebBrowserVideo.Dock = System.Windows.Forms.DockStyle.Fill;

this.WebBrowserVideo.Location = new System.Drawing.Point(0, 0);

this.WebBrowserVideo.MinimumSize = new System.Drawing.Size(20, 20);

this.WebBrowserVideo.Name = "WebBrowserVideo";

this.WebBrowserVideo.Size = new System.Drawing.Size(606, 361);

this.WebBrowserVideo.TabIndex = 6;

//

// MediaPlayer

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(606, 361);

this.Controls.Add(this.WebBrowserVideo);

this.Font = new System.Drawing.Font("Segoe UI", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));

this.MaximizeBox = false;
```

```
this.MinimizeBox = false;

this.Name = "VideoPlayer";

this.ShowIcon = false;

this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;

this.Text = "Exibindo vídeo";

this.TopMost = true;

this.WindowState = System.Windows.Forms.FormWindowState.Maximized;

this.ResumeLayout(false);

}

#endregion

public System.Windows.Forms.WebBrowser WebBrowserVideo;

}

}

Oraculo\Tests\Bartlet.cs

?using System;

using System.Collections.Generic;
```

```
using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Accord.Statistics;

using System.Data;

using Accord.Statistics.Distributions.Univariate;
```

```
namespace Accord.Statistics.Testing
```

```
{
```

```
    class Bartlet
```

```
    {
```

```
        public bool AreEqual { get; set; }
```

```
        public double M { get; private set; }
```

```
        public double C { get; private set; }
```

```
        public double MC { get; private set; }
```

```
        public Bartlet()
```

```
        {
```

```
            M = 0;
```

```
            C = 0;
```

```

MC = 0;

AreEqual = false;

}

public Bartlet(double[][] samples, double glEG, double glNG, double qmEG)

{

    Run(samples, glEG, glNG, qmEG);

}

public bool Run(double[][] samples, double glEG, double glNG, double qmEG)

{

    //calcula as variâncias e bartlet

    int k = samples.Length - 1;

    double[] bartlet = new double[k+1];

    for (int i = 0; i <= k ; i++)

    {

        double s = samples[i].Variance();

        bartlet[i] = glEG * System.Math.Log(s);

    }

    //calcula m, c e m/c

```

```

M = glNG * System.Math.Log(qmEG) - bartlet.Sum();

C = 1 + (1/9d) * (4 * ((1/glEG) - (1/glNG)));

MC = M/C;

//cria a distribuição qui-quadrado para k-1 graus e liberdade e testa a probabilidade 5%

ChiSquareDistribution chi = new ChiSquareDistribution(k);

double qui = chi.InverseDistributionFunction(0.95);

//não rejeita a hipótese se qui < mc

AreEqual = (MC < qui);

return true;

}

}

}

```

Oraculo\Tests\Bonferroni.cs

```
?using System;
```

```
using System.Collections.Generic;
```

```

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Accord.Statistics;

using System.Data;

using Accord.Statistics.Distributions.Univariate;

using Oraculo.Global;

namespace Accord.Statistics.Testing
{
    class Bonferroni
    {
        public DataTable ResultsTable { get; set; }

        public List<string> DiffList { get; set; } //lista os grupos diferentes

        public Bonferroni (double[][] samples, List<string> varnames)
        {
            ResultsTable = new DataTable();

            //inicializa a datatable dos resultados

            ResultsTable.Columns.Add("Diferenças", typeof(string));
        }
    }
}

```

```

ResultsTable.Columns.Add("p-valor", typeof(double));

ResultsTable.Columns.Add("Estatística de teste", typeof(double));

ResultsTable.Columns.Add("Valor crítico", typeof(double));

Run(samples, varnames);

}

public double getPValorFromDiferencas(string d)

{

for (int r = 0; r < this.ResultsTable.Rows.Count; r++)

{

if (this.ResultsTable.Rows[r].Field<string>(0) == d)

{

return this.ResultsTable.Rows[r].Field<double>(1);

}

}

return 0;

}

public bool Run (double[][] samples, List<string> varnames)

```

```

{

//define o número de colunas

int k = samples.Length;

//obtem o total de observações n e as médias (de cada coluna e a média total)

int n = 0;

double[] medias = new double[k];

double soma = 0;

for (int c = 0; c < k; c++)

{

    n += samples[c].Length;

    medias[c] = samples[c].Mean();

    soma += samples[c].Sum();

}

//define os graus de liberdade

double gl = (n - k);

//calcula o quadrado médio de cada coluna

```



```

double[] sq = new double[k];

for (int c = 0; c < k; c++)

{

    double nc = samples[c].Length;

    for (int l = 0; l < nc; l++)

    {

        sq[c] += System.Math.Pow(samples[c][l] - medias[c],2);

    }

}

//calcula o quadrado médio dentro dos grupos

double meansq = sq.Sum() / gl;

//calcula o erro m

double m = (k * (k - 1)) / 2;

//calcula o nível de significância

double alfa = 0.025d / m; //0,5 mas o calculo de TDistribution é unicaudal

//calcula t

```

```

TDistribution tdist = new TDistribution(gl);

double t = tdist.InverseDistributionFunction(1 - alfa);

//inicializa a lista de grupos onde há diferenças

DiffList = new List<string>();

//obtem o valor absoluto da subtração das médias, o p-valor e o valor crítico

foreach (var c in Combinatorics.Combinations(2, k))

{

    string v = string.Format("{0} - {1}", varnames[c[0] - 1], varnames[c[1] - 1]);

    double mediaabs = System.Math.Abs(medias[c[0] - 1] - medias[c[1] - 1]);

    double statTeste = medias[c[0] - 1] - medias[c[1] - 1];

    double n1 = samples[c[0] - 1].Length;

    double n2 = samples[c[1] - 1].Length;

    double SEDifference = mediaabs / System.Math.Sqrt(meansq * (1/n1 + 1/n2));

    double pvalor = (1 - tdist.DistributionFunction(SEDifference))*2*k;

    if (pvalor > 1) pvalor = 1;

    DataRow r = ResultsTable.NewRow();

    r[0] = v;

```

```

        r[1] = pvalor;

        r[2] = statTeste; //mediaabs;

        r[3] = t;

        ResultsTable.Rows.Add(r);

        //se p-valor for menor que 0.05 há diferença significativa entre os grupos

        if (pvalor < 0.05) DiffList.Add(v);
    }

    return true;
}
}
}

```

Oraculo\Tests\BonferroniTest.cs

```

?using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

```

```

using System.Threading.Tasks;

using System.Data;

using Accord.Statistics.Testing;

using System.Windows.Forms;

using Oraculo.Global;

using Oraculo.Tests.Helpers;

namespace Oraculo.Tests

{

    class BonferroniTest : Oraculo.Tests.StatisticalTest

    {

        public ResultsTable ResultsTable { get; private set; }

        public bool Run(DataTable data, List<string> varsel)

        {

            //verifica se as variáveis são numéricas

            foreach (string item in varsel)

            {

                if (!IsNumeric(data.Columns[item]))

                {

```

```
        MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe apenas variáveis numéricas e tente novamente.", item), "Erro ao realizar ANOVA com a correção do Epsilon de Greenhouse e Geisser", MessageBoxButtons.OK, MessageBoxIcon.Error);
```

```
        return false;
```

```
    }
```

```
}
```

```
//identifica primeiro todas as linhas vazias
```

```
//List<int> emptyrows = data.GetEmptyRowsList(varsel);
```

```
//converte o array do c# para um vetor double, excluindo valores nulos das amostras
```

```
//double[][] dr = data.ToBidimensionalDoubleArray(varsel, emptyrows);
```

```
double[][] dr = data.ToBidimensionalDoubleArray(varsel);
```

```
Accord.Statistics.Testing.Bonferroni b = new Accord.Statistics.Testing.Bonferroni(dr, varsel);
```

```
//analisa em que grupos há diferenças significativas
```

```
if (b.DiffList.Count > 0)
```

```
{
```

Result.Add("O Teste de Bonferroni permite identificar onde está a diferença através de comparação por pares (Pair-wise test), para isso, o p-valor tem que ser menor que 0,05.");

```
string textodiff = "Existe diferença significativa ";
```

```
for (int i = 0; i <= b.DiffList.Count - 1; i++)
```

```
{
```

```
    if (i > 0 && i < b.DiffList.Count - 1) textodiff += ", ";
```

```
    if (i > 0 && i == b.DiffList.Count - 1) textodiff += " e ";
```

```
        textodiff += "entre os grupos " + b.DiffList[i] + " com p-valor de " +  
b.getPValorFromDiferencas(b.DiffList[i]).ToString("0.000");
```

```
    }
```

```
textodiff += ".";
```

```
Result.Add(textodiff);
```

```
}
```

```
//cria a tabela html de resultado
```

```
ResultsTable = new ResultsTable("Teste de Bonferroni", b.ResultsTable);
```

```
        return true;
    }
}
}
```

Oraculo\Tests\DescriptiveTable.cs

```
?using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using Accord.Statistics.Distributions.Univariate;
using Accord.Statistics.Analysis;
using Accord.Statistics;
using System.Windows.Forms;
using Oraculo.Global;
using Oraculo.Tests.Helpers;
```

```

namespace Oraculo.Tests

{

class DescriptiveTable : StatisticalTest

{

public DescriptiveTableResultsTable HTMLTable { get; set; }

public DescriptiveTable()

{

HTMLTable = new DescriptiveTableResultsTable();

HTMLTable.Title = "Estatística descritiva das variáveis";

}

private double CalculateKurtosis(double[] data)

{

Array.Sort<double>(data);

EmpiricalDistribution dados = new EmpiricalDistribution(data);

double q1 = dados.Samples.LowerQuartile();

double q3 = dados.GetRange(0.75).Max;

double p10 = dados.Samples.Kurtosis();

```



```

double p90 = dados.GetRange(0.90).Max;

double k = (q3 - q1) / (2 * (p90 - p10));

return k;
}

public bool Run(DataTable data, List<string> varsel)
{
    //se nenhuma variável for numérica, emite mensagem de erro

    if (data.Columns.Count == 0)
    {
        MessageBox.Show("Não foi possível realizar a análise descritiva das variáveis, pois
        não há variáveis numéricas. Informe pelo menos uma variável numérica e tente novamente.",
        "Erro ao realizar a estatística descritiva", MessageBoxButtons.OK, MessageBoxIcon.Error);

        return false;
    }

    //verifica se as variáveis são numéricas e remove da lista as que não são

    for (int i = 0; i < varsel.Count; i++)
    {
        if (IsNumeric(data.Columns[varsel[i]]))

```

```

{

//converte o array do c# para vetor double

List<double> dl = new List<double>();

for (int j = 0; j < data.Rows.Count; j++)

{

if (!data.Rows[j][varsel[i]].IsNullOrEquivalent())

{

dl.Add(Convert.ToDouble(data.Rows[j][varsel[i]]));

}

}

double[] dr = dl.ToArray<double>();

//cria a distribuição (empírica)

EmpiricalDistribution d = new EmpiricalDistribution(dr);

//cria novo item na tabela

DescriptiveTableColumn rows = new DescriptiveTableColumn();

//cria a coluna no cabeçalho

rows.Name = varsel[i];

```

```
//n

int n = dr.Count();

rows.Contagem = n;

//medidas de tendência central

//média

rows.Media = dr.Mean();

//mediana

rows.Mediana = dr.Median();

//modo

rows.Moda = dr.Mode();

//medidas de dispersão

//desvio padrão (sd)

double sd = dr.StandardDeviation(dr.Mean());
```

```
rows.DesvioPadrao = sd;

//variância

rows.VarianciaAmostra = dr.Variance(dr.Mean());

//erro padrão

double ep = 0; if (n != 0) ep = sd / Math.Sqrt(n);

rows.ErroPadrao = ep;

//amplitude (range)

double min = d.Samples.Min();

double max = d.Samples.Max();

rows.Amplitude = max - min;

//mínimo

rows.Minimo = min;

//máximo

rows.Maximo = max;
```

```
//medidas de ordenamento e forma

//curtose

rows.Curtose = dr.Kurtosis();

//assimetria

rows.Assimetria = dr.Skewness();

//soma

rows.Soma = dr.Sum();

//adiciona na tabela síntese os valores

HTMLTable.AddColumn(rows);

}

}

return true;

}

}
```

Oraculo\Tests\DMS.cs

```
?namespace Accord.Statistics.Testing
```

```
{
```

```
    using System;
```

```
    using System.Collections.Generic;
```

```
    using Accord.Math;
```

```
    using Accord.Compat;
```

```
    using System.Linq;
```

```
    using System.Data;
```

```
    using Oraculo.Global;
```

```
class DMS
```

```
{
```

```
    public DataTable ResultsTable { get; set; }
```

```
    public DMS(int k, int n, List<string> varnames, double[] ranksum)
```

```
    {
```

```
        ResultsTable = new DataTable();
```

```

//inicializa a datatable dos resultados

ResultsTable.Columns.Add("Grupos comparados", typeof(string));

ResultsTable.Columns.Add("Diferença", typeof(double));

ResultsTable.Columns.Add("DMS", typeof(double));

ResultsTable.Columns.Add("São diferentes?", typeof(double));

initialize(k, n, varnames, ranksum);
}

public void initialize(int k, int n, List<string> varnames, double[] ranksum)
{
//comparar os grupos para verificar qual é o diferente (DMS)

double alfa = 0.05d / (k * (k - 1));

double z = Math.Abs(Normal.Inverse(alfa));

double dms = z * Math.Sqrt((n * k * (k + 1)) / 6d);

//preenche as linhas

foreach (int[] c in Oraculo.Global.Combinatorics.Combinations(2, varnames.Count))
{

DataRow r = ResultsTable.NewRow();

```

```

r[0] = varnames[c[0] - 1] + " - " + varnames[c[1] - 1];

double dif = Math.Abs(ranksum[c[0] - 1] - ranksum[c[1] - 1]);

r[1] = dif;

r[2] = dms.ToString("0.000");

r[3] = dif < dms ? "Não" : "Sim";

ResultsTable.Rows.Add(r);

    }

}

}

}

```

Oraculo\Tests\FDistribution.cs

```

?// Accord Statistics Library

// The Accord.NET Framework

// http://accord-framework.net

//

// Copyright © César Souza, 2009-2017

// cesarsouza at gmail.com

```



```
//  
  
// This library is free software; you can redistribute it and/or  
  
// modify it under the terms of the GNU Lesser General Public  
  
// License as published by the Free Software Foundation; either  
  
// version 2.1 of the License, or (at your option) any later version.  
  
//  
  
// This library is distributed in the hope that it will be useful,  
  
// but WITHOUT ANY WARRANTY; without even the implied warranty of  
  
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
  
// Lesser General Public License for more details.  
  
//  
  
// You should have received a copy of the GNU Lesser General Public  
  
// License along with this library; if not, write to the Free Software  
  
// Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA  
  
//
```

```
namespace Oraculo.Tests //Accord.Statistics.Distributions.Univariate
```

```
{
```

```
    using System;
```

```
    using Accord;
```

using Accord.Math;

using Accord.Statistics.Distributions.Fitting;

using Accord.Statistics.Distributions;

using Accord.Statistics.Distributions.Univariate;

using Accord.Compat;

/// <summary>

/// F (Fisher-Snedecor) distribution.

/// </summary>

///

/// <remarks>

/// <para>

/// In probability theory and statistics, the F-distribution is a continuous

/// probability distribution. It is also known as Snedecor's F distribution

/// or the Fisher-Snedecor distribution (after R.A. Fisher and George W. Snedecor).

/// The F-distribution arises frequently as the null distribution of a test statistic,

/// most notably in the analysis of variance; see <see
cref="Accord.Statistics.Testing.FTest"/>.</para>

///

/// <para>

```

/// References:

/// <list type="bullet">

/// <item><description><a href="http://en.wikipedia.org/wiki/F-distribution">

///   Wikipedia, The Free Encyclopedia. F-distribution. Available on:

///   http://en.wikipedia.org/wiki/F-distribution </a></description></item>

/// </list></para>

/// </remarks>

///

/// <example>

/// <para>

///   The following example shows how to construct a Fisher-Snedecor's F-distribution

///   with 8 and 5 degrees of freedom, respectively.</para>

///

/// <code>

/// // Create a Fisher-Snedecor's F distribution with 8 and 5 d.f.

/// FDistribution F = new FDistribution(degrees1: 8, degrees2: 5);

///

/// // Common measures

/// double mean = F.Mean; // 1.6666666666666667

/// double median = F.Median; // 1.0545096252132447

```

```

/// double var = F.Variance; // 7.6388888888888893

///

/// // Cumulative distribution functions

/// double cdf = F.DistributionFunction(x: 0.27); // 0.049463408057268315

/// double ccdf = F.ComplementaryDistributionFunction(x: 0.27); //
0.95053659194273166

/// double icdf = F.InverseDistributionFunction(p: cdf); // 0.27

///

/// // Probability density functions

/// double pdf = F.ProbabilityDensityFunction(x: 0.27); // 0.45120469723580559

/// double lpdf = F.LogProbabilityDensityFunction(x: 0.27); // -0.79583416831212883

///

/// // Hazard (failure rate) functions

/// double hf = F.HazardFunction(x: 0.27); // 0.47468419528555084

/// double chf = F.CumulativeHazardFunction(x: 0.27); // 0.050728620222091653

///

/// // String representation

/// string str = F.ToString(CultureInfo.InvariantCulture); // F(x; df1 = 8, df2 = 5)

/// </code>

/// </example>

```

```

///

[Serializable]

public class FDistribution : Accord.Statistics.Distributions.Univariate.FDistribution,
ISampleableDistribution<double> // UnivariateContinuousDistribution,
ISampleableDistribution<double>

{

    // Distribution parameters

    private double d1;

    private double d2;

    // Derived values

    private double b;

    private double? mean;

    private double? variance;

    /// <summary>

    /// Constructs a F-distribution with

    /// the given degrees of freedom.

    /// </summary>

    ///

```

```

public FDistribution()

    : this(1, 1)

{

}

/// <summary>

/// Constructs a F-distribution with

/// the given degrees of freedom.

/// </summary>

///

/// <param name="degrees1">The first degree of freedom. Default is 1.</param>

/// <param name="degrees2">The second degree of freedom. Default is 1.</param>

///

public FDistribution(double degrees1, double degrees2)

{

    if (degrees1 <= 0)

        throw new ArgumentOutOfRangeException("degrees1", "Degrees of freedom must
be positive.");

    if (degrees2 <= 0)

```

```
        throw new ArgumentOutOfRangeException("degrees1", "Degrees of freedom must  
be positive.");
```

```
        this.d1 = degrees1;
```

```
        this.d2 = degrees2;
```

```
        this.b = Beta.Function(degrees1 * 0.5, degrees2 * 0.5);
```

```
    }
```

```
    /// <summary>
```

```
    /// Gets the first degree of freedom.
```

```
    /// </summary>
```

```
    ///
```

```
    public new double DegreesOfFreedom1
```

```
    {
```

```
        get { return d1; }
```

```
    }
```

```
    /// <summary>
```

```
    /// Gets the second degree of freedom.
```

```
/// </summary>

///

public new double DegreesOfFreedom2

{

    get { return d2; }

}
```

```
/// <summary>

/// Gets the mean for this distribution.
```

```
/// </summary>

///

public override double Mean

{

    get

    {

        if (!mean.HasValue)

        {

            if (d2 <= 2)

            {

                mean = Double.NaN;
```



```

    }

    else

    {

        mean = d2 / (d2 - 2.0);

    }

}

return mean.Value;

}

}

/// <summary>

/// Gets the variance for this distribution.

/// </summary>

///

public override double Variance

{

    get

    {

        if (!variance.HasValue)

```

```

    {
        if (d2 <= 4)
        {
            variance = Double.NaN;
        }
        else
        {
            variance = (2.0 * d2 * d2 * (d1 + d2 - 2)) /
                (d1 * (d2 - 2) * (d2 - 2) * (d2 - 4));
        }
    }

    return variance.Value;
}
}

```

```

/// <summary>

```

```

/// Gets the mode for this distribution.

```

```

/// </summary>

```

```

///

```

```
/// <value>

/// The distribution's mode value.

/// </value>

///

public override double Mode

{

    get

    {

        if (d1 > 2)

        {

            double a = (d1 - 2.0) / d1;

            double b = d2 / (d2 + 2.0);

            return a * b;

        }

        return Double.NaN;

    }

}

/// <summary>
```

```

/// Gets the support interval for this distribution.

/// </summary>

///

/// <value>

/// A <see cref="DoubleRange" /> containing

/// the support interval for this distribution.

/// </value>

///

public override DoubleRange Support

{

    get { return new DoubleRange(0, Double.PositiveInfinity); }

}

/// <summary>

/// Gets the entropy for this distribution.

/// </summary>

///

public override double Entropy

{

    get { throw new NotSupportedException(); }

}

```

```
}
```

```
/// <summary>
```

```
/// Gets the cumulative distribution function (cdf) for
```

```
/// the F-distribution evaluated at point <c>x</c>.
```

```
/// </summary>
```

```
///
```

```
/// <param name="x">A single point in the distribution range.</param>
```

```
///
```

```
/// <remarks>
```

```
/// <para>
```

```
/// The Cumulative Distribution Function (CDF) describes the cumulative
```

```
/// probability that a given value or any value smaller than it will occur.</para>
```

```
/// <para>
```

```
/// The F-distribution CDF is computed in terms of the <see cref="Beta.Incomplete">
```

```
/// Incomplete Beta function  $I_x(a,b)$ </see> as  $CDF(x) = I_u(d1/2, d2/2)$  in which
```

```
///  $u$  is given as  $u = (d1 * x) / (d1 * x + d2)$ </para>.
```

```
/// </remarks>
```

```
///
```

```
protected override double InnerDistributionFunction(double x)
```

```

{

    double u = (d1 * x) / (d1 * x + d2);

    return Beta.Incomplete(d1 * 0.5, d2 * 0.5, u);

}

/// <summary>

/// Gets the complementary cumulative distribution

/// function evaluated at point <c>x</c>.

/// </summary>

///

/// <remarks>

/// <para>

/// The F-distribution complementary CDF is computed in terms of the <see
cref="Beta.Incomplete">

/// Incomplete Beta function  $I_x(a,b)$ </see> as  $CDF_c(x) = I_u(d2/2, d1/2)$  in which

/// u is given as  $u = (d2 * x) / (d2 * x + d1)$ </para>.

/// </remarks>

///

protected override double InnerComplementaryDistributionFunction(double x)

{

```

```

double u = d2 / (d2 + d1 * x);

return Beta.Incomplete(d2 * 0.5, d1 * 0.5, u);

}

/// <summary>

/// Gets the inverse of the cumulative distribution function (icdf) for

/// this distribution evaluated at probability <c>p</c>. This function

/// is also known as the Quantile function.

/// </summary>

///

/// <remarks>

/// The Inverse Cumulative Distribution Function (ICDF) specifies, for

/// a given probability, the value which the random variable will be at,

/// or below, with that probability.

/// </remarks>

///

protected override double InnerInverseDistributionFunction(double p)

{

    // Cephes Math Library Release 2.8: June, 2000

    // Copyright 1984, 1987, 1995, 2000 by Stephen L. Moshier

```

```
// Adapted under the LGPL with permission of original author.
```

```
double d1 = this.d1;
```

```
double d2 = this.d2;
```

```
double x;
```

```
double w = Beta.Incomplete(0.5 * d2, 0.5 * d1, 0.5);
```

```
if (w > p || p < 0.001)
```

```
{
```

```
    w = Beta.IncompleteInverse(0.5 * d1, 0.5 * d2, p);
```

```
    x = d2 * w / (d1 * (1.0 - w));
```

```
}
```

```
else
```

```
{
```

```
    w = Beta.IncompleteInverse(0.5 * d2, 0.5 * d1, 1.0 - p);
```

```
    x = (d2 - d2 * w) / (d1 * w);
```

```
}
```



```

    return x;

}

/// <summary>

/// Gets the probability density function (pdf) for

/// the F-distribution evaluated at point <c>x</c>.

/// </summary>

///

/// <param name="x">A single point in the distribution range.</param>

///

/// <returns>

/// The probability of <c>x</c> occurring

/// in the current distribution.

/// </returns>

///

/// <remarks>

/// The Probability Density Function (PDF) describes the

/// probability that a given value <c>x</c> will occur.

/// </remarks>

///

```

```
protected override double InnerProbabilityDensityFunction(double x)
```

```
{
```

```
    double u = Math.Pow(d1 * x, d1) * Math.Pow(d2, d2) /
```

```
        Math.Pow(d1 * x + d2, d1 + d2);
```

```
    return Math.Sqrt(u) / (x * b);
```

```
}
```

```
/// <summary>
```

```
/// Gets the log-probability density function (pdf) for
```

```
/// this distribution evaluated at point <c>x</c>.
```

```
/// </summary>
```

```
///
```

```
/// <param name="x">A single point in the distribution range.</param>
```

```
///
```

```
/// <returns>
```

```
/// The logarithm of the probability of <c>x</c>
```

```
/// occurring in the current distribution.
```

```
/// </returns>
```

```
///
```

```
/// <remarks>
```

```

/// The Probability Density Function (PDF) describes the
/// probability that a given value <c>x</c> will occur.
/// </remarks>
///
protected override double InnerLogProbabilityDensityFunction(double x)
{
    double lnu = d1 * Math.Log(d1 * x) + d2 * Math.Log(d2) -
        (d1 + d2) * Math.Log(d1 * x + d2);
    return 0.5 * lnu - Math.Log(x * b);
}

/// <summary>
/// Not available.
/// </summary>
///
public override void Fit(double[] observations, double[] weights, IFittingOptions
options)
{
    throw new NotSupportedException();
}

```

```

}

/// <summary>

/// Creates a new object that is a copy of the current instance.

/// </summary>

/// <returns>

/// A new object that is a copy of this instance.

/// </returns>

///

public override object Clone()

{

    return new FDistribution(d1, d2);

}

#region ISamplableDistribution<double> Members

/// <summary>

/// Generates a random vector of observations from the current distribution.

/// </summary>

///

```

```

/// <param name="samples">The number of samples to generate.</param>

/// <param name="result">The location where to store the samples.</param>

/// <param name="source">The random number generator to use as a source of
randomness.

/// Default is to use <see cref="Accord.Math.Random.Generator.Random"/>.</param>

///

/// <returns>A random vector of observations drawn from this distribution.</returns>

///

public override double[] Generate(int samples, double[] result, Random source)

{

    return Random(d1, d2, samples, result, source);

}

/// <summary>

/// Generates a random observation from the current distribution.

/// </summary>

///

/// <returns>A random observations drawn from this distribution.</returns>

///

public override double Generate(Random source)

```

```

    {

        return Random(d1, d2, source);

    }

    /// <summary>

    /// Generates a random vector of observations from the

    /// F-distribution with the given parameters.

    /// </summary>

    ///

    /// <param name="d1">The first degree of freedom.</param>

    /// <param name="d2">The second degree of freedom.</param>

    /// <param name="samples">The number of samples to generate.</param>

    ///

    /// <returns>An array of double values sampled from the specified F-
distribution.</returns>

    ///

    public static double[] Random(double d1, double d2, int samples)

    {

        return Random(d1, d2, samples, Accord.Math.Random.Generator.Random);

    }

```

```

/// <summary>

/// Generates a random vector of observations from the

/// F-distribution with the given parameters.

/// </summary>

///

/// <param name="d1">The first degree of freedom.</param>

/// <param name="d2">The second degree of freedom.</param>

/// <param name="samples">The number of samples to generate.</param>

/// <param name="result">The location where to store the samples.</param>

///

/// <returns>An array of double values sampled from the specified F-
distribution.</returns>

///

public static double[] Random(double d1, double d2, int samples, double[] result)

{

    return Random(d1, d2, samples, result, Accord.Math.Random.Generator.Random);

}

/// <summary>

```

```

/// Generates a random observation from the
/// F-distribution with the given parameters.
/// </summary>
///
/// <param name="d1">The first degree of freedom.</param>
/// <param name="d2">The second degree of freedom.</param>
///
/// <returns>A random double value sampled from the specified F-distribution.</returns>
///
public static double Random(double d1, double d2)
{
    return Random(d1, d2, Accord.Math.Random.Generator.Random);
}

```

```

/// <summary>
/// Generates a random vector of observations from the

```



```

/// F-distribution with the given parameters.

/// </summary>

///

/// <param name="d1">The first degree of freedom.</param>

/// <param name="d2">The second degree of freedom.</param>

/// <param name="samples">The number of samples to generate.</param>

/// <param name="source">The random number generator to use as a source of
randomness.

/// Default is to use <see cref="Accord.Math.Random.Generator.Random"/>.</param>

///

/// <returns>An array of double values sampled from the specified F-
distribution.</returns>

///

public static double[] Random(double d1, double d2, int samples, Random source)

{

    return Random(d1, d2, samples, new double[samples], source);

}

/// <summary>

/// Generates a random vector of observations from the

/// F-distribution with the given parameters.

```

```

/// </summary>

///

/// <param name="d1">The first degree of freedom.</param>

/// <param name="d2">The second degree of freedom.</param>

/// <param name="samples">The number of samples to generate.</param>

/// <param name="result">The location where to store the samples.</param>

/// <param name="source">The random number generator to use as a source of
randomness.

/// Default is to use <see cref="Accord.Math.Random.Generator.Random"/>.</param>

///

/// <returns>An array of double values sampled from the specified F-
distribution.</returns>

///

public static double[] Random(double d1, double d2, int samples, double[] result,
Random source)

{

    double[] x = GammaDistribution.Random(shape: d1 / 2.0, scale: 2, samples: samples,
result: result, source: source);

    double[] y = GammaDistribution.Random(shape: d2 / 2.0, scale: 2, samples: samples,
source: source);

    for (int i = 0; i < x.Length; i++)

```

```

        x[i] /= y[i];

    return x;

}

/// <summary>

/// Generates a random observation from the

/// F-distribution with the given parameters.

/// </summary>

///

/// <param name="d1">The first degree of freedom.</param>

/// <param name="d2">The second degree of freedom.</param>

/// <param name="source">The random number generator to use as a source of
randomness.

/// Default is to use <see cref="Accord.Math.Random.Generator.Random"/>.</param>

///

/// <returns>A random double value sampled from the specified F-distribution.</returns>

///

public static double Random(double d1, double d2, Random source)

{

    double x = GammaDistribution.Random(shape: d1 / 2.0, scale: 2, source: source);

```

```
double y = GammaDistribution.Random(shape: d2 / 2.0, scale: 2, source: source);

return x / y;

}
```

```
#endregion
```

```
/// <summary>
```

```
/// Returns a <see cref="System.String"/> that represents this instance.
```

```
/// </summary>
```

```
///
```

```
/// <returns>
```

```
/// A <see cref="System.String"/> that represents this instance.
```

```
/// </returns>
```

```
///
```

```
public new string ToString(string format, IFormatProvider formatProvider)
```

```
{
```

```
    return String.Format("F(x; df1 = {0}, df2 = {1})",
```

```
        d1.ToString(format, formatProvider),
```

```
        d2.ToString(format, formatProvider));  
    }  
}  
}
```

Oraculo\Tests\Friedman.cs

```
?namespace Accord.Statistics.Testing
```

```
{
```

```
    using System;
```

```
    using System.Collections.Generic;
```

```
    using Accord.Math;
```

```
    using Accord.Compat;
```

```
    using System.Linq;
```

```
    using System.Data;
```

```
    using Oraculo.Global;
```

```
    /// <summary>
```

```
    /// Friedman Test.
```

```
    /// </summary>
```

```
    class Friedman
```

```

{

    public double PValue { get; private set; }

    public double Stat { get; private set; }

    public double CriticalValue { get; set; }

    public Friedman(double[][] samples, List<string> varnames)
    {
        initialize(samples, varnames);
    }

    private void initialize(double[][] samples, List<string> varnames)
    {
        //passo 1. calcular n (linhas) e k (colunas)

        int k = samples.Length;

        int n = samples[0].Length; //todas as colunas tem o mesmo tamanho

        bool empates = false;

        double[][] postos = new double[n][];

        //determina o rank por linha e se há empates

```

```
for (int i = 0; i < n; i++)  
  
{  
  
    List<double> valores = new List<double>();  
  
    for (int j = 0; j < k; j++)  
  
    {  
  
        valores.Add(samples[j][i]);  
  
    }  
  
    valores.Sort();  
  
    //valores.Reverse();  
  
    double pos = 0;  
  
    double[] posicao = new double[k];  
  
    for (int j = 0; j < k; j++)  
  
    {  
  
        pos += 1;  
  
        posicao[j] = pos;
```

```

int[] replist = Enumerable.Range(0, valores.Count).Where(x => valores[x] ==
valores[j]).ToArray();

if (replist.Length > 1)

{

    empates = true; //houve empate

    double repos = mediapos(Convert.ToInt16(pos), replist.Length);

    for (int x = 0; x < replist.Length; x++)

        posicao[x + j] = repos;

    j += replist.Length - 1;

    pos += replist.Length - 1;

}

}

for (int j = 0; j < k; j++)

{

    if (postos[i] == null) postos[i] = new double[k];

    int idx = Enumerable.Range(0, valores.Count).First(x => valores[x] ==
samples[j][i]);

```



```

        postos[i][j] = posicao[idx];
    }

}

//há empates?

if (empates)
{
    //faz a soma dos postos ao quadrado

    double[] r = new double[k];

    double[] r2 = new double[k];

    double[] s = new double[k];

    for (int i = 0; i < k; i++)
    {
        for (int j = 0; j < n; j++)
        {
            r[i] += postos[j][i];

            r2[i] += Math.Pow(postos[j][i], 2);
        }
    }
}

```

```

        s[i] = Math.Pow(r[i] - n * ((k + 1) / 2), 2);
    }

    //calcula t

    double a1 = r2.Sum();

    double c1 = (k * n * Math.Pow(k + 1, 2)) / 4f;

    Stat = ((k - 1) * s.Sum()) / (a1 - c1);
}

else

{

    //passo 2. Fazer a ordenação dos valores por coluna para obter os postos

    //double[][] postos = new double[k][];

    postos = new double[k][];

    for (int i = 0; i < n; i++)

    {

        List<double> valores = new List<double>();

```

```

for (int j = 0; j < k; j++)

{

    valores.Add(samples[j][i]);

}

valores.Sort();

valores.Reverse();

double pos = 0;

double[] posicao = new double[k];

for (int j = 0; j < k; j++)

{

    pos += 1;

    posicao[j] = pos;

    int[] replot = Enumerable.Range(0, valores.Count).Where(x => valores[x] ==
valores[j]).ToArray();

    if (replot.Length > 1)

    {

        double repos = mediapos(Convert.ToInt16(pos), replot.Length);

```

```

    for (int x = 0; x < replist.Length; x++)

        posicao[x + j] = repos;

    j += replist.Length - 1;

    pos += replist.Length - 1;

}

}

for (int j = 0; j < k; j++)

{

    if (postos[j] == null) postos[j] = new double[n];

    int idx = Enumerable.Range(0, valores.Count).First(x => valores[x] ==
samples[j][i]);

    postos[j][i] = posicao[idx];

}

}

//passo 3. calcular a soma dos postos para cada grupo

double[] r = new double[k];

```

```
for (int i = 0; i < k; i++)
```

```
{
```

```
    r[i] += postos[i].Sum();
```

```
}
```

```
//passo 4. calcular o x2
```

```
double x2 = 0;
```

```
for (int i = 0; i < r.Length; i++)
```

```
{
```

```
    x2 += (r[i] * r[i]);
```

```
}
```

```
x2 = (12 / (double)(n * k * (k+1))) * x2 - (3*n*(k + 1));
```

```
Stat = x2;
```

```
}
```

```
//passo 5. verificar as hipóteses
```

```
int gl = k - 1;
```

```

var chi = new ChiSquareTest(Stat, gl);

//TODO: Faltava Friedman tratar pequenas amostras

PValue = chi.PValue;

CriticalValue = 0; //TODO: O valor crítico de Friedman não está implementado
chi.CriticalValue;

//passo 6. comparar os grupos para verificar qual é o diferente (DMS)

//obs: optamos por fazer bonferroni
}

private double mediapos(int pos, int qtde)
{
    int soma = pos;

    for (int i = 1; i < qtde; i++)
    {
        soma += pos + i;
    }
}

```

```

    double media = (double)soma / (double)qtde;

    return media;
}

private double[] obterempates(double[][] samples, double k, double b)
{
    //determina o rank por linha

    //eleva os ranks ao quadrado

    return null;
}

}

}

```

Oraculo\Tests\FriedmanTest.cs

```
?using System;
```

```
using System.Collections.Generic;
```

```

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using Accord.Statistics.Testing;

using Accord.Statistics.Distributions.Univariate;

using System.Windows.Forms;

using Oraculo.Global;

using System.Drawing.Text;

namespace Oraculo.Tests
{
    class FriedmanTest : Oraculo.Tests.StatisticalTest
    {
        public string DMSTable { get; private set; }

        public bool Run(DataTable data, List<string> varsel)
        {
            Name = "Teste de Friedman";

            //verifica se foram fornecidos pelo menos três variáveis/grupos

```



```

if (varsel.Count < 3)

{

    MessageBox.Show(String.Format("É necessário informar pelo menos três variáveis.
Selecione pelo menos três variáveis numéricas e tente novamente."), "Erro ao realizar o Teste
de Friedman", MessageBoxButtons.OK, MessageBoxIcon.Error);

    return false;

}

//valida primeiro as variáveis

foreach (string item in varsel)

{

    //verifica se as variáveis são numéricas

    if (!IsNumeric(data.Columns[item]))

    {

        MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe
apenas variáveis numéricas e tente novamente.", item), "Erro ao realizar o Teste de
Friedman", MessageBoxButtons.OK, MessageBoxIcon.Error);

        return false;

    }

}

//verifica se há pelo menos 3 amostras/linhas/sujeitos

```

```

if (data.CountNumericValuesAt(item) < 3)

{

    MessageBox.Show(String.Format(String.Format("A variável '{0}' tem menos de
três observações (linhas). Informe pelo menos três observações/linhas e tente novamente.",
item)), "Erro ao realizar o Teste de Friedman", MessageBoxButtons.OK,
MessageBoxIcon.Error);

    return false;

}

}

//identifica primeiro todas as linhas vazias

List<int> emptyrows = data.GetEmptyRowsList(varsel);

//converte o array do c# para um vetor double, excluindo valores nulos das amostras

double[][] dr = data.ToBidimensionalDoubleArray(varsel, emptyrows);

//executa o teste

Accord.Statistics.Testing.Friedman f = new
Accord.Statistics.Testing.Friedman(dr, varsel);

PValue = f.PValue;

Stat = Math.Abs(f.Stat);

```

```
CriticalValue = f.CriticalValue;
```

```
string result = string.Format("O Teste de Friedman para os '{0}' grupos retornou um p-valor de {1}. Portanto, ", dr.Length, PValue.ToString("0.000"));
```

```
//se  $p > 0.05$ , é normal
```

```
if (PValue > 0.05)
```

```
{
```

```
    result += "maior que 0,05. Sendo assim, os grupos possuem a mesma distribuição de valores.";
```

```
}
```

```
else
```

```
{
```

```
    result += "menor ou igual a 0,05. Os grupos não possuem diferença significativa.";
```

```
}
```

```
this.Result.Add(result);
```

```
//limpa da memória a instância
```

```
//r.Dispose();
```

```
        return true;

    }

}

}
```

Oraculo\Tests\FTest.cs

```
?// Accord Statistics Library
```

```
// The Accord.NET Framework
```

```
// http://accord-framework.net
```

```
//
```

```
// Copyright © César Souza, 2009-2017
```

```
// cesarsouza at gmail.com
```

```
//
```

```
// This library is free software; you can redistribute it and/or
```

```
// modify it under the terms of the GNU Lesser General Public
```

```
// License as published by the Free Software Foundation; either
```

```
// version 2.1 of the License, or (at your option) any later version.
```

```
//
```

```
// This library is distributed in the hope that it will be useful,
```

```
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
// Lesser General Public License for more details.
//
// You should have received a copy of the GNU Lesser General Public
// License along with this library; if not, write to the Free Software
// Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
//
```

```
namespace Oraculo.Tests //Accord.Statistics.Testing
```

```
{
    using System;
    using Accord.Statistics.Distributions.Univariate;
    using Accord.Statistics.Distributions;
    using Accord.Statistics.Testing;
    using System.Diagnostics;
    using Accord.Compat;
```

```
/// <summary>
```

```
/// Snedecor's F-Test.
```

```

/// </summary>

///

/// <remarks>

/// <para>

/// A F-test is any statistical test in which the test statistic has an

/// <see cref="FDistribution">F-distribution</see> under the null hypothesis.

/// It is most often used when comparing statistical models that have been fit

/// to a data set, in order to identify the model that best fits the population

/// from which the data were sampled.</para>

///

/// <para>

/// References:

/// <list type="bullet">

/// <item><description><a href="http://en.wikipedia.org/wiki/F-test">

/// Wikipedia, The Free Encyclopedia. F-Test. Available on:

/// http://en.wikipedia.org/wiki/F-test </a></description></item>

/// </list></para>

/// </remarks>

///

/// <seealso cref="OneWayAnova"/>

```

```
/// <seealso cref="TwoWayAnova"/>

///

/// <example>

/// <code>

/// // The following example has been based on the page "F-Test for Equality
/// // of Two Variances", from NIST/SEMATECH e-Handbook of Statistical Methods:
/// //
/// // http://www.itl.nist.gov/div898/handbook/eda/section3/eda359.htm
/// //
///
/// // Consider a data set containing 480 ceramic strength
/// // measurements for two batches of material. The summary
/// // statistics for each batch are shown below:
///
/// // Batch 1:
///
/// int numberOfObservations1 = 240;
///
/// double mean1 = 688.9987;
///
/// double stdDev1 = 65.54909;
///
/// double var1 = stdDev1 * stdDev1;
///
```

```

/// // Batch 2:

/// int numberOfObservations2 = 240;

/// // double mean2 = 611.1559;

/// double stdDev2 = 61.85425;

/// double var2 = stdDev2 * stdDev2;

///

/// // Here, we will be testing the null hypothesis that

/// // the variances for the two batches are equal.

///

/// int degreesOfFreedom1 = numberOfObservations1 - 1;

/// int degreesOfFreedom2 = numberOfObservations2 - 1;

///

/// // Now we can create a F-Test to test the difference between variances

/// var ftest = new FTest(var1, var2, degreesOfFreedom1, degreesOfFreedom2);

///

/// double statistic = ftest.Statistic; // 1.123037

/// double pvalue = ftest.PValue; // 0.185191

/// bool significant = ftest.Significant; // false

///

/// // The F test indicates that there is not enough evidence

```



```

/// // to reject the null hypothesis that the two batch variances
/// // are equal at the 0.05 significance level.
/// </code>
/// </example>
///
/// <seealso cref="Accord.Statistics.Distributions.Univariate.FDistribution"/>
///
[Serializable]
public class FTest : Accord.Statistics.Testing.FTest // HypothesisTest<FDistribution>
{
    /// <summary>
    /// Gets the alternative hypothesis under test. If the test is
    /// <see cref="IHypothesisTest.Significant"/>, the null hypothesis can be rejected
    /// in favor of this alternative hypothesis.
    /// </summary>
    ///
    public new TwoSampleHypothesis Hypothesis { get; private set; }

    /// <summary>

```

```

/// Gets the degrees of freedom for the

/// numerator in the test distribution.

/// </summary>

///

public new double DegreesOfFreedom1 { get { return
StatisticDistribution.DegreesOfFreedom1; } }

/// <summary>

/// Gets the degrees of freedom for the

/// denominator in the test distribution.

/// </summary>

///

public new double DegreesOfFreedom2 { get { return
StatisticDistribution.DegreesOfFreedom2; } }

/// <summary>

/// Creates a new F-Test for a given statistic with given degrees of freedom.

/// </summary>

///

/// <param name="var1">The variance of the first sample.</param>

/// <param name="var2">The variance of the second sample.</param>

```

```

/// <param name="d1">The degrees of freedom for the first sample.</param>

/// <param name="d2">The degrees of freedom for the second sample.</param>

/// <param name="alternate">The alternative hypothesis (research hypothesis) to
test.</param>

///

public FTest(double var1, double var2, double d1, double d2,

    TwoSampleHypothesis alternate =
TwoSampleHypothesis.FirstValueIsGreaterThanSecond)

{

    Compute(var1 / var2, d1, d2, alternate);

}

/// <summary>

///  Creates a new F-Test for a given statistic with given degrees of freedom.

/// </summary>

///

/// <param name="statistic">The test statistic.</param>

/// <param name="d1">The degrees of freedom for the numerator.</param>

/// <param name="d2">The degrees of freedom for the denominator.</param>

/// <param name="alternate">The alternative hypothesis (research hypothesis) to
test.</param>

```

```

///

public FTest(double statistic, double d1, double d2,

    TwoSampleHypothesis alternate =
TwoSampleHypothesis.FirstValueIsGreaterThanOrEqualToSecond)

{

    Compute(statistic, d1, d2, alternate);

}

/// <summary>

///  Computes the F-test.

/// </summary>

///

protected void Compute(double statistic, double d1, double d2, TwoSampleHypothesis
alternate)

{

    this.Statistic = statistic;

    this.StatisticDistribution = new Oraculo.Tests.FDistribution(d1, d2);

    this.Hypothesis = alternate;

    this.Tail = (DistributionTail)alternate;

    this.PValue = StatisticToPValue(Statistic);

```

```

        this.OnSizeChanged();
    }

    /// <summary>
    ///   Creates a new F-Test.
    /// </summary>
    ///
    protected FTest()
    {
    }

    /// <summary>
    ///   Converts a given test statistic to a p-value.
    /// </summary>
    ///
    /// <param name="x">The value of the test statistic.</param>
    ///
    /// <returns>The p-value for the given statistic.</returns>
    ///
    public override double StatisticToPValue(double x)

```

```

{

    if (Double.IsNaN(this.Statistic))

    {

        Trace.TraceWarning("The test statistic is NaN, probably because its standard error
is zero. This test is not applicable in this case as the samples do " +

            "not come from a normal distribution. One way to overcome this problem may be
to increase the number of samples in your experiment.");

        return Double.NaN;

    }

    double p;

    switch (Tail)

    {

        case DistributionTail.TwoTail:

            p = 2.0 * StatisticDistribution.DistributionFunction(Math.Abs(x));

            break;

        case DistributionTail.OneUpper:

            p = StatisticDistribution.ComplementaryDistributionFunction(x);

            break;

```

```

    case DistributionTail.OneLower:

        p = StatisticDistribution.DistributionFunction(x);

        break;

    default:

        throw new InvalidOperationException();

    }

    return p;

}

/// <summary>

/// Converts a given p-value to a test statistic.

/// </summary>

///

/// <param name="p">The p-value.</param>

///

/// <returns>The test statistic which would generate the given p-value.</returns>

///

public override double PValueToStatistic(double p)

{

```

```
double f;

switch (Tail)
{
    case DistributionTail.OneLower:

        f = StatisticDistribution.InverseDistributionFunction(p);

        break;

    case DistributionTail.OneUpper:

        f = StatisticDistribution.InverseDistributionFunction(1.0 - p);

        break;

    case DistributionTail.TwoTail:

        f = StatisticDistribution.InverseDistributionFunction(1.0 - p / 2.0);

        break;

    default:

        throw new InvalidOperationException();

}

return f;

}

}
```


Oraculo\Tests\GreenhouseGeisserEpsilon.cs

```
?namespace Accord.Statistics.Testing
{
    using System;

    using System.Collections.Generic;

    using Accord.Math;

    using Accord.Compat;

    using System.Linq;

    using System.Data;

    using Oraculo.Global;

    /// <summary>
    /// Greenhouse and Geisser Epsilon (EGG)
    /// </summary>
    class GreenhouseGeisserEpsilon
    {
        //GG epsilon

        public double Stat { get; private set; }
    }
}
```

```
public GreenhouseGeisserEpsilon(double[][] samples, List<string> varnames)
{
    initialize(samples, varnames);
}
```

```
private void initialize(double[][] samples, List<string> varnames)
```

```
{
    int k = samples.Length;

    int n = samples[0].Length;

    //calcula as médias e variâncias das colunas

    double[] variancias = new double[k];

    for (int i = 0; i < k; i++)
    {
        variancias[i] = samples[i].Variance();
    }
```

```
//calcula a media das variâncias
```

```
double mediavariancias = variancias.Mean();
```

```

//constroi a matriz de covariancia

double[,] matrizcovariancia = new double[k, k];

for (int c = 0; c < k; c++)

{

    for (int l = 0; l < k; l++)

    {

        if (c == l)

        {

            //usa variância da própria coluna

            matrizcovariancia[l, c] = variancias[c];

        }

        else

        {

            //usa covariancia das duas variáveis

            matrizcovariancia[l, c] = samples[l].Covariance(samples[c]);

        }

    }

}

```

```

//calcula a média da matriz de covariância por coluna (grupo)

double[] mediamatrizcovariãnciaporcoluna = matrizcovariãncia.Mean(1);

double mediamatrizcovariãncia = matrizcovariãncia.Mean();

//calcula a soma do quadrado das médias

double somaquadradomediasmatrizcovariãncia = 0;

for (int i= 0; i< mediamatrizcovariãnciaporcoluna.Length;i++)

{

    somaquadradomediasmatrizcovariãncia += mediamatrizcovariãnciaporcoluna[i] *
mediamatrizcovariãnciaporcoluna[i];

}

//calcula soma do quadrado da matriz

double somaquadradomatriz = 0;

for (int c = 0; c < k; c++)

{

    for (int l = 0; l < k; l++)

```

```

    {
        somaquadradoMatriz += matrizCovariancia[c,l] * matrizCovariancia[c,l];
    }
}

//gg numerador

double ggNumerador = Math.Pow(k * (mediavariâncias - mediaMatrizCovariancia),2);

//gg denominador

double ggDenominador = (k - 1) * (somaquadradoMatriz - 2 * k *
somaquadradoMediasMatrizCovariancia + Math.Pow(k * mediaMatrizCovariancia, 2));

//gg epsilon

Stat = ggNumerador / ggDenominador;
}
}
}

```

Oraculo\Tests\GreenhouseGeisserEpsilonTest.cs

```
?using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using Accord.Statistics.Testing;

using System.Windows.Forms;

using Oraculo.Global;

namespace Oraculo.Tests

{

    enum ANOVATests

    {

        OneWay,

        RepeatedMeasures,

        TwoWay

    }

    class GreenhouseGeisserEpsilonTest : Oraculo.Tests.StatisticalTest
```

```

{

public double GGE { get; private set; } //correção gge

public string ANOVAResultsTable { get; private set; }

public bool Run(DataTable data, List<string> varsel, ANOVATests test)
{
    //verifica se as variáveis são numéricas

    foreach (string item in varsel)
    {
        if (!IsNumeric(data.Columns[item]))
        {
            MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe apenas variáveis numéricas e tente novamente.", item), "Erro ao realizar ANOVA com a correção do Epsilon de Greenhouse e Geisser", MessageBoxButtons.OK, MessageBoxIcon.Error);

            return false;
        }
    }

    //identifica primeiro todas as linhas vazias

    List<int> emptyrows = data.GetEmptyRowsList(varsel);

```

```

//converte o array do c# para um vetor double, excluindo valores nulos das amostras

double[][] dr = data.ToBidimensionalDoubleArray(varsel, emptyrows);

//calcula egg

Accord.Statistics.Testing.GreenhouseGeisserEpsilon gge = new
Accord.Statistics.Testing.GreenhouseGeisserEpsilon(dr, varsel);

GGE = gge.Stat;

//executa o teste

if (test == ANOVATests.OneWay)

{

    Oraculo.Tests.OneWayAnovaTest anova = new
Oraculo.Tests.OneWayAnovaTest();

    if (anova.Run(data, varsel, GGE))

    {

        Result = anova.Result;

        PValue = anova.PValue;

        Stat = Math.Abs(anova.Stat);

        CriticalValue = anova.CriticalValue;

        ANOVAResultsTable = anova.ANOVAResultsTable;

```



```

    }

    else

    {

        return false;

    }

}

else if (test == ANOVATests.RepeatedMeasures)

{

    Oraculo.Tests.RepeatedMeasuresAnovaTest anova = new
Oraculo.Tests.RepeatedMeasuresAnovaTest();

    if (anova.Run(data, varsel, GGE))

    {

        Result = anova.Result;

        PValue = anova.PValue;

        Stat = anova.Stat;

        CriticalValue = anova.CriticalValue;

        ANOVAResultsTable = anova.ANOVAResultsTable;

    }

    else

    {

```

```
        return false;
    }
}

//limpa da memória a instância

//r.Dispose();

return true;
}
}
}
```

Oraculo\Tests\HuynhFeldtEpsilon.cs

```
?namespace Accord.Statistics.Testing
{
    using System;

    using System.Collections.Generic;

    using Accord.Math;

    using Accord.Compat;
```

```

using System.Linq;

using System.Data;

using Oraculo.Global;

/// <summary>

/// Huynh and Feldt Epsilon (HFE)

/// </summary>

class HuynhFeldtEpsilon

{

    //HF epsilon

    public double Stat { get; private set; }

    public HuynhFeldtEpsilon(int k, int n, double GGE)

    {

        initialize(k, n, GGE);

    }

    private void initialize(int k, int n, double GGE)

    {

        //gg numerador

```

```
double hfnumerador = n * (k-1) * GGE - 2;

//gg denominador

double hfdenominador = (k - 1) * (n - 1 - (k - 1)* GGE);

//gg epsilon

Stat = hfnumerador / hfdenominador;

}

}

}
```

Oraculo\Tests\HuynhFeldtEpsilonTest.cs

```
?using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using Accord.Statistics.Testing;
```

```

using System.Windows.Forms;

using Oraculo.Global;

namespace Oraculo.Tests
{
    class HuynhFeldtEpsilonTest : Oraculo.Tests.StatisticalTest
    {
        public string ANOVAResultsTable { get; private set; }

        public bool Run(DataTable data, List<string> varsel, ANOVATests test, double gge)
        {
            //verifica se as variáveis são numéricas

            foreach (string item in varsel)
            {
                if (!IsNumeric(data.Columns[item]))
                {
                    MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe apenas variáveis numéricas e tente novamente.", item), "Erro ao realizar ANOVA com a correção do Epsilon de Greenhouse e Geisser", MessageBoxButtons.OK, MessageBoxIcon.Error);

                    return false;
                }
            }
        }
    }
}

```

```

    }

}

//identifica primeiro todas as linhas vazias

List<int> emptyrows = data.GetEmptyRowsList(varsel);

//converte o array do c# para um vetor double, excluindo valores nulos das amostras

double[][] dr = data.ToBidimensionalDoubleArray(varsel, emptyrows);

//calcula hfe

int k = dr.Length;

int n = dr[0].Length;

//calcula hfe

Accord.Statistics.Testing.HuynhFeldtEpsilon hfe = new
Accord.Statistics.Testing.HuynhFeldtEpsilon(k, n, gge);

//executa o teste

if (test == ANOVATests.OneWay)

{

```

```

        Oraculo.Tests.OneWayAnovaTest anova = new
Oraculo.Tests.OneWayAnovaTest();

        if (anova.Run(data, varsel, hfe.Stat))

        {

            Result = anova.Result;

            PValue = anova.PValue;

            Stat = Math.Abs(anova.Stat);

            CriticalValue = anova.CriticalValue;

            ANOVAResultsTable = anova.ANOVAResultsTable;

        }

        else

        {

            return false;

        }

    }

    else if (test == ANOVATests.RepeatedMeasures)

    {

        Oraculo.Tests.RepeatedMeasuresAnovaTest anova = new
Oraculo.Tests.RepeatedMeasuresAnovaTest();

        if (anova.Run(data, varsel, hfe.Stat))

        {

```

```
Result = anova.Result;

PValue = anova.PValue;

Stat = Math.Abs(anova.Stat);

CriticalValue = anova.CriticalValue;

ANOVAResultsTable = anova.ANOVAResultsTable;

}

else

{

    return false;

}

}

//limpa da memória a instância

//r.Dispose();

return true;

}

}

}
```


Oraculo\Tests\KolmogorovSmirnovTest.cs

```
?using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
using System.Data;
```

```
using Accord.Statistics.Testing;
```

```
using Accord.Statistics.Distributions.Univariate;
```

```
using System.Windows.Forms;
```

```
using Oraculo.Global;
```

```
namespace Oraculo.Tests
```

```
{
```

```
    public class KolmogorovSmirnovTest : Oraculo.Tests.StatisticalTest
```

```
    {
```

```
        public bool IsNormal { get; set; }
```

```
        public override bool Run(DataTable data, string var)
```

```

{

    //...verifica se a variável dependente é numérica

    if (!IsNumeric(data.Columns[var]))

    {

        MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe uma
variável numérica e tente novamente.", var), "Erro ao realizar teste de normalidade",
MessageBoxButtons.OK, MessageBoxIcon.Error);

        return false;

    }

    //converte o array do c# para um vetor double

    int n = data.Rows.Count;

    List<double> dl = new List<double>();

    for (int i = 0; i < n; i++)

    {

        if (!data.Rows[i][var].IsNullOrEquivalent())

        {

            dl.Add(Convert.ToDouble(data.Rows[i][var]));

        }

    }

}

```

```

double[] dr = dl.ToArray<double>();

n = dr.Length;

//cria a distribuição para o teste

EmpiricalDistribution drd = new EmpiricalDistribution(dr.ToArray<double>());

NormalDistribution normd = new NormalDistribution(drd.Mean,
drd.StandardDeviation);

//executa o teste

 Accord.Statistics.Testing.KolmogorovSmirnovTest testResult = new
 Accord.Statistics.Testing.KolmogorovSmirnovTest(dr.ToArray<double>(),normd);

 PValue = testResult.PValue;

 this.Result.Add(string.Format("<p>O n (tamanho da amostra) da variável '{0}' é de
 {1}, sendo maior do que 50. Portanto, deve-se fazer o teste de normalidade de Kolmogorov-
 Smirnov.</p>", var, n));

 string result = string.Format("<p>O Teste Kolmogorov-Smirnov para a variável '{0}'
 retornou um p-valor de {1}. Portanto, ", var, PValue.ToString("0.000"));

 //se p > 0.05, é normal

```

```
if (PValue > 0.05)

{

    result += "maior que 0,05. Sendo assim, sua variável dependente é paramétrica
(normal).";

    this.IsNormal = true;

}

else

{

    result += "menor ou igual a 0,05. Sendo assim, sua variável dependente não é
paramétrica.";

    this.IsNormal = false;

}

this.Result.Add(result);

//limpa da memória a instância

//r.Dispose();

return true;

}

}
```

```
}
```

Oraculo\Tests\KruskalWallis.cs

```
?namespace Accord.Statistics.Testing
```

```
{
```

```
    using System;
```

```
    using System.Collections.Generic;
```

```
    using Accord.Math;
```

```
    using Accord.Compat;
```

```
    using System.Linq;
```

```
    /// <summary>
```

```
    /// KruskalWallis Test or ANOVA by ranks test.
```

```
    /// </summary>
```

```
    [Serializable]
```

```
    class KruskalWallis
```

```
    {
```

```
        public double PValue { get; private set; }
```

```
        public double Stat { get; private set; }
```

```

public double CriticalValue { get; set; }

public KruskalWallis(params double[][] samples)

{

    initialize(samples);

}

private void initialize(double[][] samples)

{

    //passo 1. calcular N e n

    int N = 0;

    int[] n = new int[samples.Length];

    for (int i = 0; i < samples.Length; i++)

    {

        N += n[i] = samples[i].Length;

    }

    //passo 2. Fazer a ordenação dos valores para obter os postos

    List<double> valor = new List<double>();

```

```

for (int i = 0; i < samples.Length; i++)

    for (int j=0;j< samples[i].Length;j++)

        valor.Add(samples[i][j]);

valor.Sort();

double pos = 0;

double[] posicao = new double[N];

for (int i = 0; i < valor.Count;i++)

{

    pos += 1;

    posicao[i] = pos;

    int[] replot = Enumerable.Range(0, valor.Count).Where(x => valor[x] ==
valor[i]).ToArray();

    if (replot.Length > 1)

    {

        double repos = mediapos(Convert.ToInt16(pos), replot.Length);

        for (int j = 0; j < replot.Length; j++)

```

```

    {
        posicao[i+j] = repos;
    }

    i += replist.Length - 1;

    pos += replist.Length - 1;
}
}

//passo 3. calcular a soma dos postos para cada grupo

double[] r = new double[samples.Length];

for (int i = 0; i < samples.Length; i++)
{
    for (int j = 0; j < samples[i].Length; j++)
    {
        r[i] += posicao[valor.IndexOf(samples[i][j])];
    }
}
}

```



```

//passo 4. calcular o h

double h = 0;

for (int i = 0; i < r.Length; i++)
{
    h += (r[i] * r[i]) / n[i];
}

h = (12 / (double)(N * (N + 1))) * h - 3 * (N + 1);

//passo 5. verificar as hipóteses

int gl = n.Length - 1;

var chi = new ChiSquareTest(h, gl);

Stat = h;

PValue = chi.PValue;

CriticalValue = 0; //TODO: Kruskal Wallis ainda não calcula o chi.CriticalValue;
}

//calcula a média das próximas posições

private double mediapos(int pos, int qtde)

```

```
{  
  
    int soma = pos;  
  
    for (int i=1; i<qtde; i++)  
  
        {  
  
            soma += pos + i;  
  
        }  
  
    double media = (double)soma / (double)qtde;  
  
    return media;  
  
}  
  
}  
  
}
```

Oraculo\Tests\KruskalWallisTest.cs

```
?using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;
```

```

using System.Data;

using Accord.Statistics.Testing;

using Accord.Statistics.Distributions.Univariate;

using System.Windows.Forms;

using Oraculo.Global;

using System.Drawing.Text;

namespace Oraculo.Tests
{
    class KruskalWallisTest : Oraculo.Tests.StatisticalTest
    {
        public bool Run(DataTable data, List<string> varsel)
        {
            //verifica se foram fornecidos pelo menos três variáveis/grupos
            if (varsel.Count < 3)
            {
                MessageBox.Show(String.Format("É necessário informar pelo menos três variáveis.
                Selecione pelo menos três variáveis numéricas e tente novamente."), "Erro ao realizar o Teste
                de Kruskal-Wallis", MessageBoxButtons.OK, MessageBoxIcon.Error);

                return false;
            }
        }
    }
}

```

```

//valida primeiro as variáveis

foreach (string item in varsel)

{

    //verifica se as variáveis são numéricas

    if (item != varsel[0])

        if (!IsNumeric(data.Columns[item]))

            {

                MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe apenas variáveis numéricas e tente novamente.", item), "Erro ao realizar o Teste de Kruskal-Wallis", MessageBoxButtons.OK, MessageBoxIcon.Error);

                return false;

            }

    //verifica se há pelo menos 5 linhas/sujeitos

    if (data.CountNumericValuesAt(item) < 5)

        {

            MessageBox.Show(String.Format(String.Format("A variável '{0}' tem menos de cinco observações (linhas). Informe pelo menos cinco observações/linhas e tente novamente.", item)), "Erro ao realizar o Teste de Kruskal-Wallis", MessageBoxButtons.OK, MessageBoxIcon.Error);

            return false;

```

```

    }
}

//converte o array do c# para um vetor double, excluindo valores nulos das amostras

double[][] dr = new double[varsel.Count][];

//percorre cada coluna selecionada do dataset

foreach (string coluna in varsel)

{

    //cria variável para armazenar a lista

    List<double> dl = new List<double>();

    //percorre cada linha do dataset

    foreach (DataRow linha in data.Rows)

    {

        //insere apenas as linhas que não tenham valores nulos

        if (!linha[coluna].IsNullOrEquivalent())

        {

            dl.Add(Convert.ToDouble(linha[coluna]));

        }

    }

}

```

```

    }

    //converte para array

    dr[varsel.IndexOf(coluna)] = dl.ToArray();

}

//executa o teste

    Accord.Statistics.Testing.KruskalWallis kw = new
Accord.Statistics.Testing.KruskalWallis(dr);

    PValue = kw.PValue;

    Stat = Math.Abs(kw.Stat);

    CriticalValue = kw.CriticalValue;

    string result = string.Format("O Teste Kruskal-Wallis para os '{0}' grupos retornou um
p-valor de {1}. Portanto,", dr.Length, PValue.ToString("0.000"));

    //se p > 0.05, é normal

    if (PValue > 0.05)

    {

        result += "maior que 0,05. Sendo assim, não existe diferença significativa entre os
grupos.";
    }

```

```
        AreEqual = true;

    }

    else

    {

        result += "menor ou igual a 0,05. Sendo assim, existe diferença significativa entre
os grupos analisados.";

        AreEqual = false;

    }

    this.Result.Add(result);

    //limpa da memória a instância

    //r.Dispose();

    return true;

}

}

}
```

Oraculo\Tests\LeveneTest.cs

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using Accord.Statistics.Testing;

using Accord.Statistics.Distributions.Univariate;

using System.Windows.Forms;

using Oraculo.Global;

using System.Drawing.Text;

namespace Oraculo.Tests

{

    class LeveneTest : Oraculo.Tests.StatisticalTest

    {

        public bool Run(DataTable data, List<string> varsel)

        {

            //verifica se foram fornecidos pelo menos três variáveis/grupos

            if (varsel.Count < 2)

```



```

    {

        MessageBox.Show(String.Format("É necessário informar pelo menos duas
variáveis. Selecione pelo menos duas variáveis numéricas e tente novamente."), "Erro ao
realizar o Teste de Levene", MessageBoxButtons.OK, MessageBoxIcon.Error);

        return false;

    }

//valida primeiro as variáveis

foreach (string item in varsel)

{

    //verifica se as variáveis são numéricas

    if (!IsNumeric(data.Columns[item]))

    {

        MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe
apenas variáveis numéricas e tente novamente.", item), "Erro ao realizar o Teste de Levene",
MessageBoxButtons.OK, MessageBoxIcon.Error);

        return false;

    }

}

//converte o array do c# para um vetor double, excluindo valores nulos das amostras

```

```

double[][] dr = new double[varsel.Count][];

//percorre cada coluna selecionada do dataset
foreach (string coluna in varsel)
{
    //cria variável para armazenar a lista
    List<double> dl = new List<double>();

    //percorre cada linha do dataset
    foreach (DataRow linha in data.Rows)
    {
        //insere apenas as linhas que não tenham valores nulos
        if (!linha[coluna].IsNullOrEquivalent())
        {
            dl.Add(Convert.ToDouble(linha[coluna]));
        }
    }

    //converte para array
    dr[varsel.IndexOf(coluna)] = dl.ToArray();
}

```

```

}

//executa o teste

    Accord.Statistics.Testing.LevneTest f = new
Accord.Statistics.Testing.LevneTest(dr);

    PValue = f.PValue;

    Stat = Math.Abs(f.Statistic);

    string result = string.Format("O Teste de Levene para os '{0}' grupos retornou um p-
valor de {1}. Portanto, ", dr.Length, PValue.ToString("0.000"));

//se p > 0.05, é normal

if (PValue > 0.05)

{

    result += "maior que 0,05. Sendo assim, os grupos possuem a mesma variância.";

    AreEqual = true;

}

else

{

    result += "menor ou igual a 0,05. Os grupos não possuem a mesma variância.";

    AreEqual = false;

```

```
    }

    this.Result.Add(result);

    //limpa da memória a instância

    //r.Dispose();

    return true;
}
}
}
```

Oraculo\Tests\MannWhitney.cs

```
?using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Collections;
```

```
using Accord.Statistics.Distributions.Univariate;
```

```
namespace Oraculo.Tests
```

```
{
```

```
    class MannWhitney
```

```
    {
```

```
        public double PValue { get; private set; }
```

```
        public double Stat { get; private set; }
```

```
        public double CriticalValue { get; set; }
```

```
        public MannWhitney(double[][] samples, List<string> varnames)
```

```
        {
```

```
            initialize(samples, varnames);
```

```
        }
```

```
        public MannWhitney(double[] sample1, double[] sample2, string var1, string var2)
```

```
        {
```

```
            double[][] samples = new double[2][];
```

```
            samples[0] = sample1;
```

```
            samples[1] = sample2;
```

```

List<string> varnames = new List<string>();

varnames.Add(var1);

varnames.Add(var2);

initialize(samples, varnames);
}

private void initialize(double[][] samples, List<string> varnames)
{

    //passo 1. calcular n (linhas) e k (colunas)

    int k = 2;

    int n1 = samples[0].Length;

    int n2 = samples[1].Length;

    int n = n1 + n2;

    //passo 2. Fazer a ordenação dos valores para obter os postos

    double[] ordenados = samples[0].ToList().Concat(samples[1].ToList()).ToArray();

    Array.Sort(ordenados);

```

```

double[] postos = new double[n];

for (int i = 0; i < n;)

{

    //verifica se os valores repetem

    int[] repetidos = Enumerable.Range(0, n).Where(x => ordenados[x] ==
ordenados[i]).ToArray();

    if (repetidos.Length > 1)

    {

        //calcula a média

        double media = mediapos(i+1, repetidos.Length);

        for (int x = 0; x < repetidos.Length; x++)

        {

            postos[i] = media;

            i += 1;

        }

    }

    else

    {

```

```

        postos[i] = i+1;

        i += 1;

    }

}

double[] r = new double[2];

for (int i = 0; i < n1; i++)

    r[0] += postos[Array.IndexOf(ordenados, samples[0][i])];

for (int i = 0; i < n2; i++)

    r[1] += postos[Array.IndexOf(ordenados, samples[1][i])];

//passo 4. calcular o u

double u1 = n1 * n2 + n1 * ((n1 + 1) / 2) - r[0];

double u2 = n1 * n2 + n2 * ((n2 + 1) / 2) - r[1];

//comparar e usar para r o menor dos 2

double u = (u1 < u2)? u1: u2;

//passo 5. calcular a média e variância

double mu = (n1 * n2) / 2f;

```



```

double varu = Math.Sqrt((n1*n2*(n1+n2+1))/12f);

//passo 6. calcular z

//double z = (u + 0.5 - mu) / varu;

//passo 7. determina o número de empates

//double[] ordenados = samples[0].ToList().Concat(samples[1].ToList()).ToArray();

//Array.Sort(ordenados);

List<int> empates = new List<int>();

for (int i = 0; i < ordenados.Length; i++)
{
    int count = 0;

    for (int j = i; j < ordenados.Length; j++)
    {
        if (ordenados[i] == ordenados[j]) count++;
    }

    if (count > 1)

```

```

    {
        empates.Add(count);

        i += count - 1;
    }
}

//passo 8. determina z com ajuste de empate e sem correção

double N = n1 + n2;

double su1 = (n1 * n2) / (N * (N - 1));

double su2 = (Math.Pow(N,3) - N) / 12f;

double su3 = 0;

foreach (int i in empates)

    su3 = su3 + (Math.Pow(i, 3) - i) / 12f;

double su = Math.Sqrt(su1 * (su2 - su3));

double z = (u - mu) / su;

//passo 9. calcula p-valor

Stat = u;

CriticalValue = z;

NormalDistribution dist = new NormalDistribution(mu,su);

```

```

//PValue = dist.DistributionFunction(u);

double a = dist.DistributionFunction(u);

double b = dist.ComplementaryDistributionFunction(u);

double c = Math.Min(a, b);

PValue = Math.Min(2 * c, 1);

}

private double mediapos(int pos, int qtde)

{

    int soma = pos;

    for (int i = 1; i < qtde; i++)

    {

        soma += pos + i;

    }

    double media = (double)soma / (double)qtde;

    return media;

}

}

```

```
}
```

```
Oraculo\Tests\ManWhitneyTest.cs
```

```
?using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
using System.Data;
```

```
using Accord.Statistics.Testing;
```

```
using System.Windows.Forms;
```

```
using Oraculo.Global;
```

```
namespace Oraculo.Tests
```

```
{
```

```
    class ManWhitneyTest : StatisticalTest
```

```
    {
```

```
        public bool Run(DataTable data, string var1, string var2)
```

```
        {
```

```

//...verifica se a variável dependente é numérica

if (!IsNumeric(data.Columns[var1]))

{

    MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe uma
variável numérica e tente novamente.", var1), "Erro ao realizar teste",
MessageBoxButtons.OK, MessageBoxIcon.Error);

    return false;

}

//verifica se a variável independente é numérica

if (!IsNumeric(data.Columns[var2]))

{

    MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe uma
variável numérica e tente novamente.", var2), "Erro ao realizar teste",
MessageBoxButtons.OK, MessageBoxIcon.Error);

    return false;

}

//converte o array do c# para um vetor double

List<double> dl1 = new List<double>();

List<double> dl2 = new List<double>();

```

```

for (int i = 0; i < data.Rows.Count; i++)

{

    if (!data.Rows[i][var1].IsNullOrEquivalent())

    {

        dl1.Add(Convert.ToDouble(data.Rows[i][var1]));

    }

    if (!data.Rows[i][var2].IsNullOrEquivalent())

    {

        dl2.Add(Convert.ToDouble(data.Rows[i][var2]));

    }

}

double[] dr1 = dl1.ToArray<double>();

double[] dr2 = dl2.ToArray<double>();

//executa o teste

//MannWhitneyWilcoxonTest testResult = new MannWhitneyWilcoxonTest(dr1, dr2,
TwoSampleHypothesis.ValuesAreDifferent, true, true);

```

```
MannWhitney testResult = new MannWhitney(dr1, dr2, var1, var2);
```

```
PValue = testResult.PValue;
```

```
Stat = Math.Abs(testResult.Stat);
```

```
CriticalValue = 0; //testResult.CriticalValue; TODO: Não foi implementado no Mann-Whitney ainda.
```

```
this.Result.Add("O teste escolhido foi o não pareado. Para dados não paramétricos (que não seguem a distribuição normal), não pareados, e com dois grupos utiliza-se o Teste de Mann-Whitney.");
```

```
string result = string.Format("<p>O resultado obtido no Teste de Mann-Whitney para as variáveis foi um p-valor de {0}. Portanto, ", PValue.ToString("0.000"));
```

```
//se  $p > 0.05$ , são iguais
```

```
if (PValue < 0.05)
```

```
{
```

```
    result += "menor que 0,05. Sendo assim, para um nível de confiança de 95%, pode-se dizer que existe diferença significativa para os dados analisados. ";
```

```
    this.AreEqual = true;
```

```
}
```

```
else
```

```
{
```

result += "maior que 0,05. Sendo assim, para um nível de confiança de 95%, pode-se dizer que não existe diferença significativa para os dados analisados. ";

```
    this.AreEqual = false;
```

```
  }
```

```
    this.Result.Add(result);
```

```
    return true;
```

```
  }
```

```
}
```

```
}
```

Oraculo\Tests\Mauchly.cs

```
?namespace Accord.Statistics.Testing
```

```
{
```

```
    using System;
```

```
    using System.Collections.Generic;
```

```
    using Accord.Math;
```

```
    using Accord.Compat;
```



```

using System.Linq;

using System.Data;

using Oraculo.Global;

/// <summary>

/// Mauchly's Test

/// </summary>

class Mauchly

{

    public double PValue { get; private set; }

    public double Stat { get; private set; }

    public Mauchly(double[][] samples, List<string> varnames)

    {

        initialize(samples, varnames);

    }

    private void initialize(double[][] samples, List<string> varnames)

    {

        int k = samples.Length;

```

```

int n = samples[0].Length;

int N = k * n;

//calcula as médias e variâncias das colunas

double[] medias = new double[k];

double[] variancias = new double[k];

for (int i = 0; i < k; i++)

{

    medias[i] = samples[i].Mean();

    variancias[i] = samples[i].Variance();

}

//calcula a diferença entre os grupos

IEnumerable<int[]> combs = Oraculo.Global.Combinatorics.Combinations(2, k);

double[][] diferencas = new double[combs.Count()][];

int idx = 0;

foreach (int[] c in combs)

{

```

```

double[] d = new double[n];

for (int i = 0; i < n; i++)

{

    d[i] = samples[c[0] - 1][i] - samples[c[1] - 1][i];

}

diferencas[idx] = d;

idx++;

}

//calcula a média e variância da diferença entre os grupos

double[] mediasdif = new double[k];

double[] varianciasdif = new double[k];

for (int i = 0; i < k; i++)

{

    mediasdif[i] = diferencas[i].Mean();

    varianciasdif[i] = diferencas[i].Variance();

}

//constroi a matriz de covariância

```

```

double[,] matrizcovariancia = new double[k,k];

for (int c = 0; c < k; c++)
{
    for (int l = 0; l < k; l++)
    {
        if (c == l)
        {
            //usa variância da própria coluna

            matrizcovariancia[l,c] = variancias[c];
        }
        else
        {
            //usa covariancia das duas variáveis

            matrizcovariancia[l,c] = samples[l].Covariance(samples[c]);
        }
    }
}

//calcula a média da matriz de covariancia

```

```

double[] mediamatrizcovariancia = new double[k];

mediamatrizcovariancia = matrizcovariancia.Mean(1);

//calcula o autovalor da matriz

 Accord.Math.Decompositions.EigenvalueDecomposition autovalor = new
 Accord.Math.Decompositions.EigenvalueDecomposition(matrizcovariancia, false, true);

//calcula o w

double wn = 1;

double wd = 0;

for (int i = 0; i < k; i++)

{

    wn *= autovalor.RealEigenvalues[i];

    wd += autovalor.RealEigenvalues[i];

}

wd = Math.Pow((wd / (k - 1)), k - 1);

double w = wn / wd;

//calcula o f

double fn = 2 * Math.Pow(k - 1, 2) + k + 1;

```

```

double fd = 6 * (k - 1) * ((k * 2) - 1);

double f = fn / fd;

//calcula o x2w

double x2w = (f - 1) * ((k * 2) - 1) * Math.Log(w);

//define os grau de liberdade

int gl = (k * (k - 1) / 2) - 1;

//verifica as hipóteses

var chi = new ChiSquareTest(x2w, gl);

//obtem a estatística do teste

Stat = x2w;

//PValue = chi.PValue;

PValue = chi.PValue;

}

}

}

```

Oraculo\Tests\MauchlyTest.cs

?using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using Accord.Statistics.Testing;

using Accord.Statistics.Distributions.Univariate;

using System.Windows.Forms;

using Oraculo.Global;

using System.Drawing.Text;

namespace Oraculo.Tests

{

class MauchlyTest : Oraculo.Tests.StatisticalTest

{

public bool AreSpherical { get; private set; }

public bool Run(DataTable data, List<string> varsel)

```

{

//verifica se foram fornecidos pelo menos três variáveis/grupos

if (varsel.Count < 3)

{

    MessageBox.Show(String.Format("É necessário informar pelo menos três variáveis.
Selecione pelo menos três variáveis numéricas e tente novamente."), "Erro ao realizar o Teste
de Mauchly", MessageBoxButtons.OK, MessageBoxIcon.Error);

    return false;

}

//valida primeiro as variáveis

foreach (string item in varsel)

{

//verifica se as variáveis são numéricas

if (!IsNumeric(data.Columns[item]))

{

    MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe
apenas variáveis numéricas e tente novamente.", item), "Erro ao realizar o Teste de
Friedman", MessageBoxButtons.OK, MessageBoxIcon.Error);

    return false;

}

```



```

//verifica se há pelo menos 3 amostras/linhas/sujeitos

if (data.CountNumericValuesAt(item) < 3)

{

    MessageBox.Show(String.Format(String.Format("A variável '{0}' tem menos de
três observações (linhas). Informe pelo menos três observações/linhas e tente novamente.",
item)), "Erro ao realizar o Teste de Friedman", MessageBoxButtons.OK,
MessageBoxIcon.Error);

    return false;

}

}

//identifica primeiro todas as linhas vazias

List<int> emptyrows = data.GetEmptyRowsList(varsel);

//converte o array do c# para um vetor double, excluindo valores nulos das amostras

double[][] dr = data.ToBidimensionalDoubleArray(varsel, emptyrows);

//executa o teste

Accord.Statistics.Testing.Mauchly m = new Accord.Statistics.Testing.Mauchly(dr,
varsel);

```

```
PValue = m.PValue;
```

```
Stat = Math.Abs(m.Stat);
```

```
AreSpherical = PValue > 0.05;
```

```
string result = string.Format("O Teste de Mauchly para os '{0}' grupos retornou um p-  
valor de {1}. Portanto, ", dr.Length, PValue.ToString("0.000"));
```

```
//se p > 0.05, é normal
```

```
if (PValue > 0.05)
```

```
{
```

```
    result += "maior que 0,05. Sendo assim, os grupos possuem simetria composta ou  
esférica, possuindo a mesma distribuição de valores.";
```

```
}
```

```
else
```

```
{
```

```
    result += "menor ou igual a 0,05. Os grupos não possuem simetria composta ou  
esférica, não apresentando a mesma distribuição de valores.";
```

```
}
```

```
this.Result.Add(result);
```

```
//limpa da memória a instância

//r.Dispose();

return true;

}

}

}
```

Oraculo\Tests\OneWayAnova.cs

```
?// Accord Statistics Library
```

```
// The Accord.NET Framework
```

```
// http://accord-framework.net
```

```
//
```

```
// Copyright © César Souza, 2009-2017
```

```
// cesarsouza at gmail.com
```

```
//
```

```
// This library is free software; you can redistribute it and/or
```

```
// modify it under the terms of the GNU Lesser General Public
```

```
// License as published by the Free Software Foundation; either
```

```
// version 2.1 of the License, or (at your option) any later version.  
  
//  
  
// This library is distributed in the hope that it will be useful,  
  
// but WITHOUT ANY WARRANTY; without even the implied warranty of  
  
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
  
// Lesser General Public License for more details.  
  
//  
  
// You should have received a copy of the GNU Lesser General Public  
  
// License along with this library; if not, write to the Free Software  
  
// Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA  
  
//
```

```
namespace Oraculo.Tests
```

```
{
```

```
    using System;
```

```
    using System.Collections.Generic;
```

```
    using Accord.Math;
```

```
    using Accord.Statistics;
```

```
    using Accord.Statistics.Testing;
```

```
/// <summary>
```

```
/// One-way Analysis of Variance (ANOVA).
```

```
/// </summary>
```

```
/// <remarks>
```

```
/// <para>
```

```
/// The one-way ANOVA is a way to test for the equality of three or more means at the same
```

```
/// time by using variances. In its simplest form ANOVA provides a statistical test of whether
```

```
/// or not the means of several groups are all equal, and therefore generalizes t-test to more
```

```
/// than two groups.</para>
```

```
///
```

```
/// <para>
```

```
/// References:
```

```
/// <list type="bullet">
```

```
/// <item><description><a href="http://en.wikipedia.org/wiki/Analysis_of_variance">
```

```
/// Wikipedia, The Free Encyclopedia. Analysis of variance. </a></description></item>
```

```
/// <item><description><a href="http://en.wikipedia.org/wiki/F_test">
```

```
/// Wikipedia, The Free Encyclopedia. F-Test. </a></description></item>
```

```
/// <item><description><a href="http://en.wikipedia.org/wiki/One-way_ANOVA">
```

```
/// Wikipedia, The Free Encyclopedia. One-way ANOVA. </a></description></item>
```

```
/// </list></para>

/// </remarks>

///

/// <example>

/// <para>

/// The following is the same example given in Wikipedia's page for the

/// F-Test [1]. Suppose one would like to test the effect of three levels

/// of a fertilizer on plant growth. </para>

///

/// <para>

/// To achieve this goal, an experimenter has divided a set of 18 plants on

/// three groups, 6 plants each. Each group has received different levels of

/// the fertilizer under question.</para>

///

/// <para>

/// After some months, the experimenter registers the growth for each plant: </para>

///

/// <code>

/// double[][] samples =

/// {
```

```

/// new double[] { 6, 8, 4, 5, 3, 4 }, // records for the first group

/// new double[] { 8, 12, 9, 11, 6, 8 }, // records for the second group

/// new double[] { 13, 9, 11, 8, 7, 12 }, // records for the third group

/// };

/// </code>

///

/// <para>

/// Now, he would like to test whether the different fertilizer levels has

/// indeed caused any effect in plant growth. In other words, he would like

/// to test if the three groups are indeed significantly different.</para>

///

/// <code>

/// // To do it, he runs an ANOVA test:

/// OneWayAnova anova = new OneWayAnova(samples);

/// </code>

///

/// <para>

/// After the Anova object has been created, one can display its findings

/// in the form of a standard ANOVA table by binding anova.Table to a

/// DataGridView or any other display object supporting data binding. To

```

```
/// illustrate, we could use Accord.NET's DataGridView to inspect the
```

```
/// table's contents.</para>
```

```
///
```

```
/// <code>
```

```
/// DataGridView.Show(anova.Table);
```

```
/// </code>
```

```
///
```

```
/// <para>Result in:</para>
```

```
///
```

```
/// 
```

```
///
```

```
/// <para>
```

```
/// The p-level for the analysis is about 0.002, meaning the test is
```

```
/// significant at the 5% significance level. The experimenter would
```

```
/// thus reject the null hypothesis, concluding there is a strong
```

```
/// evidence that the three groups are indeed different. Assuming the
```

```
/// experiment was correctly controlled, this would be an indication
```

```
/// that the fertilizer does indeed affect plant growth.</para>
```

```
///
```

```
/// <para>
```



```

/// [1] http://en.wikipedia.org/wiki/F\_test </para>

///

/// </example>

///

[Serializable]

public class OneWayAnova : IAnova
{

    private int groupCount;

    private int[] sizes;

    private int totalSize;

    private double[] means;

    private double totalMean;

    private double SSb; // Between-group sum of squares

    private double SSw; // Within-group sum of squares

    private double SST; // Total sum of squares

```

```

private double MSb; // Between-group mean square

private double MSw; // Within-group mean square

private double DFb; // Between-group degrees-of-freedom

private double DFw; // Within-group degrees-of-freedom

private double DFt; // Total degrees-of-freedom

/// <summary>

/// Gets the F-Test produced by this one-way ANOVA.

/// </summary>

///

public FTest FTest { get; private set; }

/// <summary>

/// Gets the ANOVA results in the form of a table.

/// </summary>

///

public AnovaSourceCollection Table { get; private set; }

private bool HasCorrection { get; set; } //tem ajuste EGG ou EHF?

```

```

private double Correction { get; set; } //valor de EGG ou EHF

public string ANOVAResultsTable { get; private set; }

/// <summary>

///  Creates a new one-way ANOVA test.

/// </summary>

///

/// <param name="samples">The sampled values.</param>

/// <param name="labels">The independent, nominal variables.</param>

///

public OneWayAnova(double[] samples, int[] labels)

{

    Setup(samples, labels);

}

public OneWayAnova(double[] samples, int[] labels, double correction)

{

    HasCorrection = true;

    Correction = correction;

    Setup(samples, labels);

```

```

}

/// <summary>

///  Creates a new one-way ANOVA test.

/// </summary>

///

/// <param name="samples">The grouped sampled values.</param>

///

public OneWayAnova(params double[][] samples)

{

    Setup(samples);

}

public OneWayAnova(double[][] samples, double correction)

{

    HasCorrection = true;

    Correction = correction;

    Setup(samples);

}

```

```
private void Setup(double[] samples, int[] labels)
{
    totalSize = samples.Length;

    groupCount = labels.Max();

    sizes = new int[groupCount];

    double[][] groups = new double[groupCount][];

    for (int i = 0; i < groups.Length; i++)
    {
        int[] idx = labels.Find(label => label == i);

        double[] group = samples.Get(idx);

        groups[i] = group;

        sizes[i] = group.Length;
    }

    initialize(groups);
}
```

```

private void Setup(params double[][] samples)
{
    sizes = new int[samples.Length];

    groupCount = samples.Length;

    for (int i = 0; i < samples.Length; i++)
        totalSize += sizes[i] = samples[i].Length;

    initialize(samples);
}

```

```

private void initialize(double[][] samples)
{
    DFb = groupCount - 1;

    DFw = totalSize - groupCount;

    DFt = totalSize - 1;

```

//se tem correção aplica sobre os graus de liberdade das colunas e do resíduo (EGG ou EHF)

```

if (HasCorrection)
{
    DFb = DFb * Correction;

    DFw = DFw * Correction;
}

// Step 1. Calculate the mean within each group
means = Measures.Mean(samples, 1);

// Step 2. Calculate the overall mean
totalMean = Measures.GrandMean(means, sizes);

// Step 3. Calculate the "between-group" sum of squares
for (int i = 0; i < samples.Length; i++)
{
    // between-group sum of squares

    double u = (means[i] - totalMean);

    SSb += sizes[i] * u * u;
}

```

```
// Step 4. Calculate the "within-group" sum of squares
```

```
for (int i = 0; i < samples.Length; i++)
```

```
{
```

```
    for (int j = 0; j < samples[i].Length; j++)
```

```
    {
```

```
        double u = samples[i][j] - means[i];
```

```
        SSw += u * u;
```

```
    }
```

```
}
```

```
SSt = SSb + SSw; // total sum of squares
```

```
// Step 5. Calculate the F statistic
```

```
MSb = SSb / DFb; // between-group mean square
```

```
MSw = SSw / DFw; // within-group mean square
```

```
FTest = new FTest(MSb / MSw, DFb, DFw);
```

```
// Step 6. Create the ANOVA table
```



```

List<AnovaVariationSource> table = new List<AnovaVariationSource>();

table.Add(new AnovaVariationSource(this, "Between-Groups", SSb,
Int32.Parse(DFb.ToString()), FTest));

table.Add(new AnovaVariationSource(this, "Within-Groups", SSw,
Int32.Parse(DFw.ToString()), null));

table.Add(new AnovaVariationSource(this, "Total", SSt, Int32.Parse(DFt.ToString()),
null));

this.Table = Table;

//Passo 6. Cria a tabela ANOVA

ANOVAResultsTable += "<h3>Tabela 4 - ANOVA</h3>";

ANOVAResultsTable += "<table>";

ANOVAResultsTable += "<tr><th>Fonte da variação</th><th>Graus de
liberdade</th><th>Soma dos quadrados</th><th>Quadrados médios</th><th>F
Calculado</th><th>p-valor</th><th>F-Crítico (Tabelado)</th></tr>";

ANOVAResultsTable +=
string.Format("<tr><td>{0}</td><td>{1}</td><td>{2:0.000}</td><td>{3:0.000}</td><td>{
4:0.000}</td><td>{5:0.000}</td><td>{6:0.000}</td>", "Entre grupos", DFb, SSb, MSb,
FTest.Statistic, FTest.CriticalValue, FTest.PValue);

ANOVAResultsTable +=
string.Format("<tr><td>{0}</td><td>{1}</td><td>{2:0.000}</td><td>{3:0.000}</td><td><
/td><td></td><td></td>", "Dentro dos grupos", DFw, SSw, MSw);

ANOVAResultsTable +=
string.Format("<tr><td>{0}</td><td>{1}</td><td>{2:0.000}</td><td></td><td></td><td><
/td><td></td>", "Total", DFt, SSt);

```

```
        ANOVAResultsTable += "</table>";  
  
    }  
  
}  
  
}
```

Oraculo\Tests\OneWayAnovaTest.cs

```
?using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;  
  
using System.Data;  
  
using Accord.Statistics.Testing;  
  
using System.Windows.Forms;  
  
using Oraculo.Global;  
  
using Accord;  
  
namespace Oraculo.Tests  
  
{
```

```

class OneWayAnovaTest : Oraculo.Tests.StatisticalTest

{

    public Bartlet BartletTest { get; set; }

    public Bonferroni BonferroniTest { get; set; }

    public string ANOVAResultsTable { get; private set; }

    private bool HasCorrection { get; set; } //tem ajuste EGG ou EHF?

    private double Correction { get; set; } //valor de EGG ou EHF

    public bool Run(DataTable data, List<string> varsel, double correction)

    {

        HasCorrection = true;

        Correction = correction;

        return Run(data, varsel);

    }

    public bool Run(DataTable data, List<string> varsel)

    {

        //verifica se as variáveis são numéricas

        foreach (string item in varsel)

```

```

{

    if (!IsNumeric(data.Columns[item]))

    {

        MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe apenas variáveis numéricas e tente novamente.", item), "Erro ao realizar ANOVA",
        MessageBoxButtons.OK, MessageBoxIcon.Error);

        return false;

    }

}

//converte o array do c# para um vetor double, excluindo valores nulos das amostras

double[][] dr = new double[varsel.Count][];

//percorre cada linha do dataset

foreach (string coluna in varsel)

{

    //cria variável para armazenar a lista

    List<double> dl = new List<double>();

    //percorre cada coluna seleccionada do dataset

    foreach (DataRow linha in data.Rows)

```

```

{
    //insere apenas as colunas que não tenham valores nulos

    if (!linha[coluna].IsNullOrEquivalent())
    {
        dl.Add(Convert.ToDouble(linha[coluna]));
    }
}

//converte para array

dr[varsel.IndexOf(coluna)] = dl.ToArray();
}

//executa o teste

Oraculo.Tests.OneWayAnova anova;

if (HasCorrection)

    anova = new Oraculo.Tests.OneWayAnova(dr, Correction);

else

    anova = new Oraculo.Tests.OneWayAnova(dr);

```

```
PValue = anova.FTest.PValue;
```

```
Stat = Math.Abs(anova.FTest.Statistic);
```

```
CriticalValue = anova.FTest.CriticalValue;
```

```
ANOVAResultsTable = anova.ANOVAResultsTable;
```

```
this.Result.Add("O teste escolhido foi o não pareado. Para dados paramétricos  
(distribuição normal), não pareados, e com mais de dois grupos utiliza-se o Teste ANOVA.");
```

```
string result = string.Format("O resultado obtido no Teste ANOVA foi um p-valor de  
{0}. Portanto, ",PValue.ToString("0.000"));
```

```
//se  $p > 0.05$ , são iguais
```

```
if (PValue < 0.05)
```

```
{
```

```
    result += "menor que 0,05. Sendo assim, para um nível de confiança de 95%, pode-  
se dizer que existe diferença significativa para os dados analisados. ";
```

```
    this.AreEqual = false;
```

```
}
```

```
else
```

```
{
```

result += "maior que 0,05. Sendo assim, para um nível de confiança de 95%, pode-se dizer que não existe diferença significativa para os dados analisados. ";

```
this.AreEqual = true;
```

```
}
```

```
this.Result.Add(result);
```

```
//limpa da memória a instância
```

```
//r.Dispose();
```

```
return true;
```

```
}
```

```
}
```

```
}
```

```
Oraculo\Tests\PairedTTest.cs
```

```
?using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```

using System.Text;

using System.Threading.Tasks;

using System.Data;

using Accord.Statistics.Testing;

using System.Windows.Forms;

using Oraculo.Global;

namespace Oraculo.Tests
{
    class PairedTTest : Oraculo.Tests.StatisticalTest
    {
        public bool Run(DataTable data, string var1, string var2)
        {
            //...verifica se a variável dependente é numérica

            if (!IsNumeric(data.Columns[var1]))
            {
                MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe uma
                variável numérica e tente novamente.", var1), "Erro ao realizar teste T",
                MessageBoxButtons.OK, MessageBoxIcon.Error);

                return false;
            }
        }
    }
}

```



```

//verifica se a variável independente é numérica

if (!IsNumeric(data.Columns[var2]))

{

    MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe uma
variável numérica e tente novamente.", var2), "Erro ao realizar teste T",
MessageBoxButtons.OK, MessageBoxIcon.Error);

    return false;

}

//Lista as colunas selecionadas

List<string> varsel = new List<string>(new string[] { var1, var2 });

//identifica primeiro todas as linhas vazias

List<int> emptyrows = data.GetEmptyRowsList(varsel);

if (emptyrows.Count > 0)

{

    MessageBox.Show("No Teste T pareado as duas amostras devem ter o mesmo
tamanho. Como as duas amostras não tem a mesma quantidade de elementos, serão ignorados
os valores vazios para a realização do teste de hipótese.", "Teste T Pareado",
MessageBoxButtons.OK, MessageBoxIcon.Warning);

```

```
}
```

```
//converte o array do c# para um vetor double, excluindo valores nulos das amostras
```

```
double[][] dr = data.ToBidimensionalDoubleArray(varsel, emptyrows);
```

```
//converte o array do c# para um vetor double
```

```
//List<double> dl1 = new List<double>();
```

```
//List<double> dl2 = new List<double>();
```

```
//for (int i = 0; i < data.Rows.Count; i++)
```

```
//{
```

```
// if (!data.Rows[i][var1].IsNullOrEquivalent())
```

```
// {
```

```
//     dl1.Add(Convert.ToDouble(data.Rows[i][var1]));
```

```
// }
```

```
// if (!data.Rows[i][var2].IsNullOrEquivalent())
```

```
// {
```

```
//     dl2.Add(Convert.ToDouble(data.Rows[i][var2]));
```

```
// }
```

```

//}

//double[] dr1 = dl1.ToArray<double>();

//double[] dr2 = dl2.ToArray<double>();

////verifica se as duas amostras tem o mesmo tamanho

//if (dr1.Length != dr2.Length)

//{

//    MessageBox.Show(String.Format("As duas amostras devem ter o mesmo
tamanho. Informe duas variáveis numéricas que tenham a mesma quantidade de indivíduos
(elementos) e tente novamente.", var2), "Erro ao realizar teste T Pareado",
MessageBoxButtons.OK, MessageBoxIcon.Error);

//    return false;

//}

//executa o teste

Accord.Statistics.Testing.PairedTTest testResult = new
Accord.Statistics.Testing.PairedTTest(dr[0],dr[1]);

PValue = testResult.PValue;

Stat = Math.Abs(testResult.Statistic);

CriticalValue = testResult.CriticalValue;

```

```
this.Result.Add("O teste escolhido foi o pareado. Para dados paramétricos (distribuição normal), pareados, e com dois grupos utiliza-se o Teste T pareado.");
```

```
string result = string.Format("<p>O resultado obtido no Teste T pareado foi um p-valor de {0}. Portanto, ", PValue.ToString("0.000"));
```

```
//se  $p > 0.05$ , são iguais
```

```
if (PValue < 0.05)
```

```
{
```

```
    result += "menor que 0,05. Sendo assim, para um nível de confiança de 95%, pode-se dizer que existe diferença significativa para os dados analisados. ";
```

```
    this.AreEqual = true;
```

```
}
```

```
else
```

```
{
```

```
    result += "maior que 0,05. Sendo assim, para um nível de confiança de 95%, pode-se dizer que não existe diferença significativa para os dados analisados. ";
```

```
    this.AreEqual = false;
```

```
}
```

```
        this.Result.Add(result);

        //limpa da memória a instância

        //r.Dispose();

        return true;
    }
}
}
```

Oraculo\Tests\RepeatedMeasuresAnova.cs

```
?namespace Accord.Statistics.Testing
{
    using System;

    using System.Collections.Generic;

    using Accord.Math;

    using Accord.Compat;

    using System.Linq;
```

```

/// <summary>

/// Repeated Measures Analysis of Variance (ANOVA).

/// </summary>

[Serializable]

class RepeatedMeasuresAnova : IAnova

{

    private int totalLinhas;

    private int totalGrupos;

    private int[] sizes;

    private int totalSize;

    /// <summary>

    /// Gets the F-Test produced by this Repeated Measures ANOVA.

    /// </summary>

    ///

    public FTest FTest { get; private set; }

    /// <summary>

    /// Gets the ANOVA results in the form of a table.

```

```

/// </summary>

///

public AnovaSourceCollection Table { get; private set; }

public string ANOVAResultsTable { get; private set; }

private bool HasCorrection { get; set; } //tem ajuste EGG ou EHF?

private double Correction { get; set; } //valor de EGG ou EHF

/// <summary>

///  Creates a new Repeated Measures ANOVA test.

/// </summary>

///

/// <param name="samples">The sampled values.</param>

/// <param name="labels">The independent, nominal variables.</param>

///

public RepeatedMeasuresAnova(double[] samples, int[] labels)

{

    Setup(samples, labels);

}

```

```

public RepeatedMeasuresAnova(double[] samples, int[] labels, double correction)

{

    HasCorrection = true;

    Correction = correction;

    Setup(samples, labels);

}

/// <summary>

    /// Creates a new Repeated Measures ANOVA test.

    /// </summary>

    ///

    /// <param name="samples">The grouped sampled values.</param>

    ///

public RepeatedMeasuresAnova(double[][] samples)

{

    Setup(samples);

}

```



```

public RepeatedMeasuresAnova(double[][] samples, double correction)
{
    HasCorrection = true;

    Correction = correction;

    Setup(samples);
}

private void Setup(double[] samples, int[] labels)
{
    //registra o tamanho dos grupos

    totalSize = samples.Length;

    totalGrupos = labels.Max();

    totalLinhas = samples.Length;

    sizes = new int[totalGrupos];

    //cria uma matriz com as linhas de cada grupo (samples) selecionado (labels)

    double[][] groups = new double[totalGrupos][];

    for (int i = 0; i < groups.Length; i++)

```

```

{

    int[] idx = labels.Find(label => label == i);

    double[] group = samples.Get(idx);

    groups[i] = group;

    sizes[i] = group.Length;

}

//calcula ANOVA

initialize(groups);

}

private void Setup(double[][] samples)

{

    //registra o tamanho dos grupos

    totalSize = samples.Length;

    sizes = new int[samples.Length];

    totalLinhas = samples[0].Length;

    totalGrupos = samples.Length;

```

```

for (int i = 0; i < samples.Length; i++)

    totalSize += sizes[i] = samples[i].Length;

//calcula ANOVA

initialize(samples);

}

private void initialize(double[][] samples)

{

    //Passo 1. Calcular a média em cada coluna (grupo), linha e a média geral

    double[] mediasLinhas = Measures.Mean(samples, 0);

    double[] mediasGrupos = Measures.Mean(samples, 1);

    double mediaTotal = Measures.GrandMean(mediasGrupos, sizes);

    //Passo 2. Calcular os graus de liberdade (gl)

    int glLinhas = totalLinhas - 1; //N-1

    double glGrupos = totalGrupos - 1; //n-1

    double glErro = glGrupos * glLinhas; //(n-1).(N-1)

    double glTotal = totalGrupos * totalLinhas - 1;

```

```
//se tem correção aplica sobre os graus de liberdade das colunas e do resíduo (EGG ou EHF)
```

```
if (HasCorrection)
```

```
{
```

```
    glGrupos = glGrupos * Correction;
```

```
    glErro = glErro * Correction;
```

```
}
```

```
//Passo 3. Calcular a soma dos quadrados (sq)
```

```
double sqLinhas = 0;
```

```
double sqGrupos = 0;
```

```
double sqTotal = 0;
```

```
for (int i = 0; i < totalGrupos; i++)
```

```
{
```

```
    //colunas (grupos)
```

```
    double u = (mediasGrupos[i] - mediaTotal);
```

```
    sqGrupos += u * u;
```

```
}
```

```
sqGrupos *= totalLinhas;
```

```

for (int i = 0; i < totalLinhas; i++)

{

    //linhas (indivíduos)

    double u = (mediasLinhas[i] - mediaTotal);

    sqLinhas += u * u;

}

sqLinhas *= totalGrupos;

for (int i = 0; i < totalGrupos; i++)

{

    //total (para calcular residuo)

    for (int j = 0; j < totalLinhas; j++)

    {

        double ut = (samples[i][j] - mediaTotal);

        sqTotal += ut * ut;

    }

}

double sqResiduo = sqTotal - sqGrupos - sqLinhas;

```

```
//Passo 4. Calcular a média da soma dos quadrados (qm = sq/gl)
```

```
double qmLinhas = sqLinhas / glLinhas;
```

```
double qmGrupos = sqGrupos / glGrupos;
```

```
double qmResiduo = sqResiduo / glErro;
```

```
double qmTotal = sqTotal / glTotal;
```

```
//Passo 5. Calcular estatística F
```

```
double f = qmLinhas / qmResiduo;
```

```
double fGrupos = qmGrupos / qmResiduo;
```

```
FTest = new Oraculo.Tests.FTest(fGrupos, glGrupos, glErro);
```

```
FTest FTestLinhas = new FTest(f, glLinhas, glErro.To<int>());
```

```
//Passo 6. Cria a tabela ANOVA
```

```
ANOVAResultsTable += "<h3>Tabela 4 - ANOVA (Medidas repetidas)</h3>";
```

```
ANOVAResultsTable += "<table>";
```

```
ANOVAResultsTable += "<tr><th>Fonte da variação</th><th>Graus de  
liberdade</th><th>Soma dos quadrados</th><th>Quadrados médios</th><th>F  
Calculado</th><th>p-valor</th><th>F-Crítico (Tabelado)</th></tr>";
```

```
ANOVAResultsTable +=  
string.Format("<tr><td>{0}</td><td>{1}</td><td>{2:0.000}</td><td>{3:0.000}</td><td>{  
4:0.000}</td><td>{5:0.000}</td><td>{6:0.000}</td>", "Grupos (Colunas)", glGrupos,  
sqGrupos, qmGrupos, FTest.Statistic, FTest.CriticalValue, FTest.PValue);
```

```
ANOVAResultsTable +=  
string.Format("<tr><td>{0}</td><td>{1}</td><td>{2:0.000}</td><td>{3:0.000}</td><td>{  
4:0.000}</td><td>{5:0.000}</td><td>{6:0.000}</td>", "Observações (Linhas)", glLinhas,  
sqLinhas, qmLinhas, FTestLinhas.Statistic, FTestLinhas.CriticalValue, FTestLinhas.PValue);
```

```
ANOVAResultsTable +=  
string.Format("<tr><td>{0}</td><td>{1}</td><td>{2:0.000}</td><td>{3:0.000}</td><td><  
/td><td></td><td></td>", "Resíduo", glErro, sqResiduo, qmResiduo);
```

```
ANOVAResultsTable +=  
string.Format("<tr><td>{0}</td><td>{1}</td><td>{2:0.000}</td><td></td><td></td><td><  
/td><td></td>", "Total", glTotal, sqTotal);
```

```
ANOVAResultsTable += "</table>";
```

```
}
```

```
}
```

```
}
```

Oraculo\Tests\RepeatedMeasuresAnovaTest.cs

```
?using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
using System.Data;
```

```

using Accord.Statistics.Testing;

using System.Windows.Forms;

using Oraculo.Global;

namespace Oraculo.Tests
{
    class RepeatedMeasuresAnovaTest : Oraculo.Tests.StatisticalTest
    {
        public string ANOVAResultsTable { get; private set; }

        private bool HasCorrection { get; set; } //tem ajuste EGG ou EHF?

        private double Correction { get; set; } //valor de EGG ou EHF

        public bool Run(DataTable data, List<string> varsel, double correction)
        {
            HasCorrection = true;

            Correction = correction;

            return Run(data, varsel);
        }
    }
}

```



```

public bool Run(DataTable data, List<string> varsel)

{

    //verifica se as variáveis são numéricas

    foreach (string item in varsel)

    {

        if (!IsNumeric(data.Columns[item]))

        {

            MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe apenas variáveis numéricas e tente novamente.", item), "Erro ao realizar ANOVA",
            MessageBoxButtons.OK, MessageBoxIcon.Error);

            return false;

        }

    }

    //identifica primeiro todas as linhas vazias

    List<int> emptyrows = data.GetEmptyRowsList(varsel);

    //converte o array do c# para um vetor double, excluindo valores nulos das amostras

    double[][] dr = data.ToBidimensionalDoubleArray(varsel, emptyrows);

    //executa o teste

```

```
RepeatedMeasuresAnova testResult;
```

```
if (HasCorrection)
```

```
    testResult = new RepeatedMeasuresAnova(dr, Correction);
```

```
else
```

```
    testResult = new RepeatedMeasuresAnova(dr);
```

```
PValue = testResult.FTest.PValue;
```

```
Stat = Math.Abs(testResult.FTest.Statistic);
```

```
CriticalValue = testResult.FTest.CriticalValue;
```

```
ANOVAResultsTable = testResult.ANOVAResultsTable;
```

```
    this.Result.Add("O teste escolhido foi o pareado. Para dados paramétricos  
(distribuição normal), pareados, e com mais de dois grupos utiliza-se o Teste ANOVA para  
medidas repetidas.");
```

```
    string result = string.Format("O resultado obtido no Teste ANOVA para medidas  
repetidas foi um p-valor de {0}. Portanto, ", PValue.ToString("0.000"));
```

```
//se  $p > 0.05$ , são iguais
```

```
if (PValue < 0.05)
```

```
{  
  
    result += "menor que 0,05. Sendo assim, para um nível de confiança de 95%, pode-  
se dizer que existe diferença significativa para os dados analisados. ";  
  
    this.AreEqual = true;  
  
}  
  
else  
  
{  
  
    result += "maior que 0,05. Sendo assim, para um nível de confiança de 95%, pode-  
se dizer que não existe diferença significativa para os dados analisados. ";  
  
    this.AreEqual = false;  
  
}  
  
this.Result.Add(result);  
  
//limpa da memória a instância  
  
//r.Dispose();  
  
return true;  
  
}  
  
}  
  
}
```

Oraculo\Tests\ShapiroTest.cs

```
?using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
using System.Data;
```

```
using Accord.Statistics.Testing;
```

```
using System.Windows.Forms;
```

```
using Oraculo.Global;
```

```
namespace Oraculo.Tests
```

```
{
```

```
    public class ShapiroTest : Oraculo.Tests.StatisticalTest
```

```
    {
```

```
        public bool IsNormal { get; set; }
```

```
        public override bool Run(DataTable data, string var)
```

```

{

    //...verifica se a variável dependente é numérica

    if (!IsNumeric(data.Columns[var]))

    {

        MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe uma
variável numérica e tente novamente.", var), "Erro ao realizar teste de normalidade",
MessageBoxButtons.OK, MessageBoxIcon.Error);

        return false;

    }

    //converte o array do c# para um vetor double

    int n = data.Rows.Count;

    List<double> dl = new List<double>();

    for (int i = 0; i < n; i++)

    {

        if (!data.Rows[i][var].IsNullOrEquivalent())

        {

            dl.Add(Convert.ToDouble(data.Rows[i][var]));

        }

    }

}

```

```

double[] dr = dl.ToArray<double>();

n = dr.Length;

//verifica se a amostra tem pelo menos 3 observações

if (n < 4)
{
    MessageBox.Show(String.Format("A amostra deve conter pelo menos quatro
indivíduos (elementos). Informe uma amostra com tamanho adequado e tente novamente.",
var), "Erro ao realizar teste de Shapiro-Wilk", MessageBoxButtons.OK,
MessageBoxIcon.Error);

    return false;
}

//executa o teste

ShapiroWilkTest testResult = new ShapiroWilkTest(dr);

PValue = testResult.PValue;

this.Result.Add(string.Format("O n (tamanho da amostra) da variável '{0}' é de {1},
sendo menor ou igual a 50. Portanto, deve-se fazer o teste de normalidade de Shapiro-Wilk.",
var, n));

```

```
string result = string.Format("O Teste Shapiro-Wilk para a variável '{0}' retornou um p-valor de {1}. Portanto ", var, PValue.ToString("0.000"));
```

```
//se p > 0.05, é normal
```

```
if (PValue > 0.05)
```

```
{
```

```
    result += "maior que 0,05, sendo assim as variáveis são paramétricas (distribuição normal).";
```

```
    this.IsNormal = true;
```

```
}
```

```
else
```

```
{
```

```
    result += "menor ou igual a 0,05, sendo assim as variáveis não são paramétricas (não segue a distribuição normal).";
```

```
    this.IsNormal = false;
```

```
}
```

```
this.Result.Add(result);
```

```
//limpa da memória a instância
```

```
//r.Dispose();
```

```
        return true;
    }
}
}
```

Oraculo\Tests\StatisticalTest.cs

```
?using System;

using System.Linq;

using System.Data;

using System.Collections.Generic;

namespace Oraculo.Tests
{
    public class StatisticalTest
    {
        public string Name { get; set; }

        public double PValue { get; set; }

        public double Stat { get; set; }
    }
}
```



```
public double CriticalValue { get; set; }
```

```
public bool AreEqual { get; set; }
```

```
public List<string> Result { get; set; }
```

```
public StatisticalTest()
```

```
{
```

```
    Name = "";
```

```
    PValue = 0;
```

```
    Stat = 0;
```

```
    CriticalValue = 0;
```

```
    AreEqual = false;
```

```
    Result = new List<string>();
```

```
}
```

```
public StatisticalTest(string name)
```

```
{
```

```
    Name = name;
```

```
    PValue = 0;
```

```
    Stat = 0;
```

```
    CriticalValue = 0;
```

```

AreEqual = false;

Result = new List<string>();

}

//testa se a coluna da tabela é um valor numérico

public static bool IsNumeric(DataColumn col)

{

    if (col == null) return false;

    //Make this const

    var numericTypes = new[] { typeof(Byte), typeof(Decimal), typeof(Double),

    typeof(Int16), typeof(Int32), typeof(Int64), typeof(SByte),

    typeof(Single), typeof(UInt16), typeof(UInt32), typeof(UInt64)};

    return numericTypes.Contains(col.DataType);

}

//executa o teste (recebe a tabela e o nome da variável que será testada

public virtual bool Run(DataTable data, string var) { return true; }

```

```

public double[,] ConvertDataTableToDouble(DataTable data)
{
    int rowCount = data.Rows.Count;

    int columnCount = data.Columns.Count;

    double[,] arr = new double[rowCount, columnCount];

    for (int i = 0; i < rowCount; i++)
    {
        for (int j = 0; j < columnCount; j++)
        {
            arr[i, j] = Convert.ToDouble(data.Rows[i][j]);
        }
    }

    return arr;
}
}
}

```

```
?using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using Accord.Statistics.Testing;

using System.Windows.Forms;

using Oraculo.Global;

namespace Oraculo.Tests
{
    public class TTest : StatisticalTest
    {
        public TTest()
        {
            Name = "Teste T Simples";
        }
    }
}
```

```

//executa o Teste T

public bool Run(DataTable data, string varsel, double mean)

{

    //...verifica se a variável selecionada é numérica

    if (!IsNumeric(data.Columns[varsel]))

    {

        MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe uma
variável numérica e tente novamente.", varsel), "Erro ao realizar teste T",
MessageBoxButtons.OK, MessageBoxIcon.Error);

        return false;

    }

    //converte o array do c# para um vetor double

    List<double> dl = new List<double>();

    for (int i = 0; i < data.Rows.Count; i++)

    {

        if (!data.Rows[i][varsel].IsNullOrEquivalent())

        {

            dl.Add(Convert.ToDouble(data.Rows[i][varsel]));

        }

    }

```

```

}

double[] dr = dl.ToArray<double>();

//executa o teste

Accord.Statistics.Testing.TTest test = new Accord.Statistics.Testing.TTest(dr, mean);

PValue = test.PValue;

Stat = Math.Abs(test.Statistic);

CriticalValue = test.CriticalValue;

this.Result.Add("O teste escolhido foi o Teste T Simples. Este teste permite testar se a
média de dois grupos é significamente diferente");

string result = string.Format("O resultado obtido no Teste T foi um p-valor de {0}.
Portanto, ", PValue.ToString("0.000"));

//se p > 0.05, são iguais

if (PValue < 0.05)

{

    result += "menor que 0,05. Sendo assim, para um nível de confiança de 95%, pode-
se dizer que existe diferença significativa para os dados analisados. ";
}

```

```

        this.AreEqual = true;

    }

    else

    {

        result += "maior que 0,05. Sendo assim, para um nível de confiança de 95%, pode-
se dizer que não existe diferença significativa para os dados analisados. ";

        this.AreEqual = false;

    }

    this.Result.Add(result);

    //limpa da memória a instância

    //r.Dispose();

    return true;

}

//executa o Teste T independente

public bool Run(DataTable data, string var1, string var2)

{

```

```

this.Name = "Teste T independente";

//...verifica se a variável dependente é numérica

if (!IsNumeric(data.Columns[var1]))

{

    MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe uma
variável numérica e tente novamente.", var1), "Erro ao realizar teste T independente",
MessageBoxButtons.OK, MessageBoxIcon.Error);

    return false;

}

//verifica se a variável independente é numérica

if (!IsNumeric(data.Columns[var2]))

{

    MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe uma
variável numérica e tente novamente.", var2), "Erro ao realizar teste T independente",
MessageBoxButtons.OK, MessageBoxIcon.Error);

    return false;

}

//converte o array do c# para um vetor double

```



```
List<double> dl1 = new List<double>();

List<double> dl2 = new List<double>();

for (int i = 0; i < data.Rows.Count; i++)

{

    if (!data.Rows[i][var1].IsNullOrEquivalent())

    {

        dl1.Add(Convert.ToDouble(data.Rows[i][var1]));

    }

    if (!data.Rows[i][var2].IsNullOrEquivalent())

    {

        dl2.Add(Convert.ToDouble(data.Rows[i][var2]));

    }

}

double[] dr1 = dl1.ToArray<double>();

double[] dr2 = dl2.ToArray<double>();

//executa o teste
```

```
TwoSampleTTest test = new TwoSampleTTest(dr1, dr2);
```

```
PValue = test.PValue;
```

```
Stat = Math.Abs(test.Statistic);
```

```
CriticalValue = test.CriticalValue;
```

```
this.Result.Add("Para dois grupos, não pareados, ambos com a distribuição normal e  
com variâncias iguais utiliza-se o Teste T independente supondo variâncias iguais.");
```

```
string result = string.Format("<p>O resultado obtido no Teste T independente foi um  
p-valor de {0}. Portanto, ", PValue.ToString("0.000"));
```

```
//se  $p > 0.05$ , são iguais
```

```
if (PValue < 0.05)
```

```
{
```

```
    result += "menor que 0,05. Sendo assim, para um nível de confiança de 95%, pode-  
se dizer que existe diferença significativa para os dados analisados. ";
```

```
    this.AreEqual = true;
```

```
}
```

```
else
```

```
{
```

result += "maior que 0,05. Sendo assim, para um nível de confiança de 95%, pode-se dizer que não existe diferença significativa para os dados analisados. ";

```
this.AreEqual = false;
```

```
}
```

```
this.Result.Add(result);
```

```
//limpa da memória a instância
```

```
//r.Dispose();
```

```
return true;
```

```
}
```

```
}
```

```
}
```

Oraculo\Tests\TwoSampleTTestTest.cs

```
?using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```

using System.Text;

using System.Threading.Tasks;

using System.Data;

using Accord.Statistics.Testing;

using System.Windows.Forms;

using Oraculo.Global;

namespace Oraculo.Tests
{
    class TwoSampleTTestTest : StatisticalTest
    {
        public bool Run(DataTable data, string var1, string var2)
        {
            //verifica se a variável dependente é numérica

            if (!IsNumeric(data.Columns[var1]))
            {
                MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe uma
                variável numérica e tente novamente.", var1), "Erro ao realizar teste",
                MessageBoxButtons.OK, MessageBoxIcon.Error);

                return false;
            }
        }
    }
}

```

```

//verifica se a variável independente é numérica

if (!IsNumeric(data.Columns[var2]))

{

    MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe uma
variável numérica e tente novamente.", var2), "Erro ao realizar teste",
MessageBoxButtons.OK, MessageBoxIcon.Error);

    return false;

}

//converte o array do c# para um vetor double

List<double> dl1 = new List<double>();

List<double> dl2 = new List<double>();

for (int i = 0; i < data.Rows.Count; i++)

{

    if (!data.Rows[i][var1].IsNullOrEquivalent())

    {

        dl1.Add(Convert.ToDouble(data.Rows[i][var1]));

    }

}

```

```

if (!data.Rows[i][var2].IsNullOrEquivalent())
{
    dl2.Add(Convert.ToDouble(data.Rows[i][var2]));
}
}

```

```
double[] dr1 = dl1.ToArray<double>();
```

```
double[] dr2 = dl2.ToArray<double>();
```

```
//executa o teste
```

```
TwoSampleTTest testResult = new TwoSampleTTest(dr1, dr2, false);
```

```
PValue = testResult.PValue;
```

```
Stat = Math.Abs(testResult.Statistic);
```

```
CriticalValue = testResult.CriticalValue;
```

this.Result.Add("Para dois grupos, não pareados, ambos com a distribuição normal e com variâncias diferentes utiliza-se o Teste T independente supondo variâncias diferentes.");

```
string result = string.Format(@"<p>O resultado obtido no Teste T para variâncias diferentes foi um p-valor de {0}. Portanto, ", PValue.ToString("0.000"));
```

```
//se p > 0.05, são iguais

if (PValue < 0.05)

{

    result += "menor que 0,05. Sendo assim, para um nível de confiança de 95%, pode-
se dizer que existe diferença significativa entre os grupos analisados. ";

    this.AreEqual = true;

}

else

{

    result += "maior que 0,05. Sendo assim, para um nível de confiança de 95%, pode-
se dizer que não existe diferença significativa entre os grupos analisados. ";

    this.AreEqual = false;

}

this.Result.Add(result);

return true;

}

}

}
```

Oraculo\Tests\Welch.cs

```
?namespace Accord.Statistics.Testing
{
    using System;

    using System.Collections.Generic;

    using Accord.Math;

    using Accord.Compat;

    using System.Linq;

    using System.Data;

    using Oraculo.Global;

    using Accord.Statistics.Distributions.Univariate;

    /// <summary>
    /// Welch T-Test.
    /// </summary>

    class Welch

    {

        public double PValue { get; private set; }

        public double Stat { get; private set; }
```



```

public double CriticalValue { get; set; }

public Welch(double[][] samples, List<string> varnames)
{
    initialize(samples, varnames);
}

private void initialize(double[][] samples, List<string> varnames)
{
    //determina o número de observações nos grupos

    double n1 = samples[0].Length;

    double n2 = samples[1].Length;

    //calcula a média estimada dos grupos

    double m = samples[0].Mean() - samples[1].Mean();

    //calcula a variância de cada grupo

    double var1 = samples[0].Variance();

    double var2 = samples[1].Variance();
}

```

```

//determina o erro padrão empírico

double erro = Math.Sqrt(var1 / n1 + var2 / n2);

//calcula a estatística t

double t = (m - 0) / erro;

Stat = t;

//determina os graus de liberdade

double gl = Math.Pow(var1/n1 + var2/n2, 2) / (Math.Pow(var1/n1, 2)/(n1 - 1) +
Math.Pow(var2/n2, 2)/(n2 - 1));

//determina o p-valor

TDistribution dist = new TDistribution(gl);

PValue = 2.0 * dist.ComplementaryDistributionFunction(Math.Abs(t));

}

}

}

```

Oraculo\Tests\WelchTest.cs

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using Accord.Statistics.Testing;

using System.Windows.Forms;

using Oraculo.Global;

namespace Oraculo.Tests
{
    class WelchTest : StatisticalTest
    {
        public bool Run(DataTable data, List<string> varsel)
        {
            //verifica se foram fornecidos pelo menos duas variáveis/grupos

            if (varsel.Count < 2)
            {

```

```
        MessageBox.Show(String.Format("É necessário informar pelo menos duas  
variáveis. Selecione pelo menos duas variáveis numéricas e tente novamente."), "Erro ao  
realizar o Teste T de Welch", MessageBoxButtons.OK, MessageBoxIcon.Error);
```

```
        return false;
```

```
    }
```

```
//valida primeiro as variáveis
```

```
foreach (string item in varsel)
```

```
{
```

```
    //verifica se as variáveis são numéricas
```

```
    if (!IsNumeric(data.Columns[item]))
```

```
    {
```

```
        MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe  
apenas variáveis numéricas e tente novamente.", item), "Erro ao realizar o Teste T de Welch",  
        MessageBoxButtons.OK, MessageBoxIcon.Error);
```

```
        return false;
```

```
    }
```

```
}
```

```
//converte o array do c# para um vetor double, excluindo valores nulos das amostras
```

```
double[][] dr = new double[varsel.Count][];
```

```

//percorre cada coluna selecionada do dataset

foreach (string coluna in varsel)

{

    //cria variável para armazenar a lista

    List<double> dl = new List<double>();

    //percorre cada linha do dataset

    foreach (DataRow linha in data.Rows)

    {

        //insere apenas as linhas que não tenham valores nulos

        if (!linha[coluna].IsNullOrEquivalent())

        {

            dl.Add(Convert.ToDouble(linha[coluna]));

        }

    }

    //converte para array

    dr[varsel.IndexOf(coluna)] = dl.ToArray();

}

```

```

//executa o teste

Accord.Statistics.Testing.Welch w = new Accord.Statistics.Testing.Welch(dr, varsel);

PValue = w.PValue;

Stat = Math.Abs(w.Stat);

string result = string.Format("O Teste T de Welch para os '{0}' grupos retornou um p-
valor de {1}. Portanto, ", dr.Length, PValue.ToString("0.000"));

//se p > 0.05, é normal

if (PValue > 0.05)

{

    result += "maior que 0,05. Sendo assim, os grupos possuem diferença
significativa.";

    AreEqual = true;

}

else

{

    result += "menor ou igual a 0,05. Os grupos não possuem diferença significativa.";

    AreEqual = false;

}

```

```
        this.Result.Add(result);

        //limpa da memória a instância

        //r.Dispose();

        return true;
    }
}
}
```

Oraculo\Tests\WilcoxonTest.cs

```
?using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using Accord.Statistics.Testing;
```

```

using System.Windows.Forms;

using Oraculo.Global;

namespace Oraculo.Tests
{
    class WilcoxonTest : StatisticalTest
    {
        public bool Run(DataTable data, string var1, string var2)
        {
            //...verifica se a variável dependente é numérica

            if (!IsNumeric(data.Columns[var1]))
            {
                MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe uma
                variável numérica e tente novamente.", var1), "Erro ao realizar teste T",
                MessageBoxButtons.OK, MessageBoxIcon.Error);

                return false;
            }

            //verifica se a variável independente é numérica

            if (!IsNumeric(data.Columns[var2]))
            {

```



```
        MessageBox.Show(String.Format("A variável '{0}' não é numérica. Informe uma  
variável numérica e tente novamente.", var2), "Erro ao realizar teste T",  
        MessageBoxButtons.OK, MessageBoxIcon.Error);
```

```
        return false;
```

```
    }
```

```
    //converte o array do c# para um vetor double
```

```
    List<double> dl1 = new List<double>();
```

```
    List<double> dl2 = new List<double>();
```

```
    for (int i = 0; i < data.Rows.Count; i++)
```

```
    {
```

```
        if (!data.Rows[i][var1].IsNullOrEquivalent())
```

```
        {
```

```
            dl1.Add(Convert.ToDouble(data.Rows[i][var1]));
```

```
        }
```

```
        if (!data.Rows[i][var2].IsNullOrEquivalent())
```

```
        {
```

```
            dl2.Add(Convert.ToDouble(data.Rows[i][var2]));
```

```
        }
```

```

    }

    double[] dr1 = dl1.ToArray<double>();

    double[] dr2 = dl2.ToArray<double>();

    //verifica se as duas amostras tem o mesmo tamanho

    if (dr1.Length != dr2.Length)

    {

        MessageBox.Show(String.Format("As duas amostras devem ter o mesmo tamanho. Informe duas variáveis numéricas que tenham a mesma quantidade de indivíduos (elementos) e tente novamente.", var2), "Erro ao realizar teste Wilcoxon", MessageBoxButtons.OK, MessageBoxIcon.Error);

        return false;

    }

    //executa o teste

    TwoSampleWilcoxonSignedRankTest testResult = new
TwoSampleWilcoxonSignedRankTest(dr1,
dr2,TwoSampleHypothesis.ValuesAreDifferent,true,true);

    PValue = testResult.PValue;

    Stat = Math.Abs(testResult.Statistic);

```

```
//Accord.Statistics.Distributions.Univariate.NormalDistribution dist = new  
Accord.Statistics.Distributions.Univariate.NormalDistribution();
```

```
//Stat = dist.InverseDistributionFunction(PValue / 2.0); //estatística de teste  
padronizado do spss
```

```
Stat = testResult.StatisticDistribution.InverseDistributionFunction(PValue / 2.0);  
//estatística de teste do spss
```

```
CriticalValue = testResult.CriticalValue;
```

```
this.Result.Add("O teste escolhido foi o pareado. Para dados não paramétricos (que  
não seguem a distribuição normal), pareados, e com dois grupos utiliza-se o Teste de  
Wilcoxon.");
```

```
string result = string.Format("O resultado obtido no Teste de Wilcoxon foi um p-valor  
de {0}. Portanto, ", PValue.ToString("0.000"));
```

```
//se  $p > 0.05$ , são iguais
```

```
if (PValue < 0.05)
```

```
{
```

```
    result += "menor que 0,05. Sendo assim, para um nível de confiança de 95%, pode-  
se dizer que existe diferença significativa para os dados analisados. ";
```

```
    this.AreEqual = true;
```

```
}
```

```
else

{

    result += "maior que 0,05. Sendo assim, para um nível de confiança de 95%, pode-
se dizer que não existe diferença significativa para os dados analisados. ";

    this.AreEqual = false;

}

this.Result.Add(result);

//limpa da memória a instância

//r.Dispose();

return true;

}

}

}
```

Oraculo\Tests\Helpers\DescriptiveTableResultsTable.cs

```
?using System;
```

```

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Oraculo.Tests.Helpers
{
    public class DescriptiveTableResultsTable : ResultsTable
    {
        public DescriptiveTableResultsTable()
        {
            //inicializa a tabela criando a primeira coluna

            AddColumn("Variável", typeof(string));

            Data.Rows.Add("Contagem");

            Data.Rows.Add("Média");

            Data.Rows.Add("Mediana");

            Data.Rows.Add("Moda");

            Data.Rows.Add("Desvio padrão");

            Data.Rows.Add("Variância da amostra");

            Data.Rows.Add("Erro padrão");
        }
    }
}

```

```

Data.Rows.Add("Amplitude");

Data.Rows.Add("Mínimo");

Data.Rows.Add("Máximo");

Data.Rows.Add("Curtose");

Data.Rows.Add("Assimetria");

Data.Rows.Add("Soma dos valores");

}

public void AddColumn(DescriptiveTableColumn var)

{

    //inclui a coluna

    AddColumn(var.Name, typeof(double));

    int c = Data.Columns.Count - 1;

    Data.Rows[0][c] = var.Contagem;

    Data.Rows[1][c] = var.Media;

    Data.Rows[2][c] = var.Mediana;

    Data.Rows[3][c] = var.Moda;

    Data.Rows[4][c] = var.DesvioPadrao;

    Data.Rows[5][c] = var.VarianciaAmostra;

```

```
Data.Rows[6][c] = var.ErroPadrao;

Data.Rows[7][c] = var.Amplitude;

Data.Rows[8][c] = var.Minimo;

Data.Rows[9][c] = var.Maximo;

Data.Rows[10][c] = var.Curtose;

Data.Rows[11][c] = var.Assimetria;

Data.Rows[12][c] = var.Soma;

}

}
```

```
public class DescriptiveTableColumn

{

    public string Name { get; set; }

    public int Contagem { get; set; }

    public double Media { get; set; }

    public double Mediana { get; set; }

    public double Moda { get; set; }

    public double DesvioPadrao { get; set; }

    public double VarianciaAmostra { get; set; }

    public double ErroPadrao { get; set; }
```

```
public double Amplitude { get; set; }

public double Minimo { get; set; }

public double Maximo { get; set; }

public double Curtose { get; set; }

public double Assimetria { get; set; }

public double Soma { get; set; }

}

}
```

Oraculo\Tests\Helpers\EqualityOfVariancesTestsResultsTable.cs

```
?using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

namespace Oraculo.Tests.Helpers

{
```



```

public class EqualityOfVariancesTestsResultsTable : ResultsTable
{
    public EqualityOfVariancesTestsResultsTable()
    {
        AddColumn("Teste", typeof(string));

        AddColumn("p-valor", typeof(double));

        Title = "Testes de igualdade das variâncias";
    }

    public void AddRow(string teste, double pvalor)
    {
        DataRow dr = Data.NewRow();

        dr[0] = teste;

        dr[1] = pvalor;

        Data.Rows.Add(dr);
    }
}

```

```
}
```

```
Oraculo\Tests\Helpers\HypothesisTestsResultsTable.cs
```

```
?using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
using System.Data;
```

```
namespace Oraculo.Tests.Helpers
```

```
{
```

```
    public class HypothesisTestsResultsTable : ResultsTable
```

```
    {
```

```
        public bool ShowCriticalValue { get; set; }
```

```
        public HypothesisTestsResultsTable(bool showcv = true)
```

```
        {
```

```
            ShowCriticalValue = showcv;
```

```

AddColumn("Teste", typeof(string));

AddColumn("p-valor", typeof(double));

AddColumn("Estatística de teste", typeof(double));

if (ShowCriticalValue) AddColumn("Valor crítico", typeof(double));

Title = "Testes de hipótese";

}

public void AddRow(string teste, double pvalor, double stat, double criticalvalue)

{

DataRow dr = Data.NewRow();

dr[0] = teste;

dr[1] = pvalor;

dr[2] = stat;

if (ShowCriticalValue) dr[3] = criticalvalue;

Data.Rows.Add(dr);

}

```

```
public override string ToHTML()
{
    if (!ShowCriticalValue)
        if (Data.Columns.Count == 4)
            Data.Columns.RemoveAt(3);

    return base.ToHTML();
}
}
```

Oraculo\Tests\Helpers\NormalityTestsResultsTable.cs

```
?using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
```

```

namespace Oraculo.Tests.Helpers

{

    public class NormalityTestsResultsTable : ResultsTable

    {

        public NormalityTestsResultsTable()

        {

            AddColumn("Grupo", typeof(string));

            AddColumn("Tamanho", typeof(int));

            AddColumn("Teste", typeof(string));

            AddColumn("p-valor", typeof(double));

            AddColumn("Distribuição", typeof(string));

            Title = "Testes de normalidade";

        }

        public void AddRow(string grupo, int tamanho, string teste, double pvalor, bool
isnormal)

        {

            DataRow dr = Data.NewRow();

```

```
dr[0] = grupo;

dr[1] = tamanho;

dr[2] = teste;

dr[3] = pvalor;

dr[4] = isnormal ? "Normal" : "Outra";

Data.Rows.Add(dr);

}

}

}
```

Oraculo\Tests\Helpers\OneGroupTestResultTable.cs

```
?using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;
```

```

namespace Oraculo.Tests.Helpers

{

class OneGroupTestResultTable : ResultsTable

{

public OneGroupTestResultTable()

{

AddColumn("Grupo", typeof(string));

AddColumn("p-valor", typeof(double));

AddColumn("Valor calculado", typeof(double));

AddColumn("Valor crítico", typeof(double));

Title = "Teste de hipótese";

}

public void AddRow(string grupo, double pvalor, double stat, double criticalvalue)

{

DataRow dr = Data.NewRow();

dr[0] = grupo;

dr[1] = pvalor;

```

```
        dr[2] = stat;

        dr[3] = criticalvalue;

        Data.Rows.Add(dr);
    }
}
}
```

Oraculo\Tests\Helpers\ResultsTable.cs

```
?using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

namespace Oraculo.Tests

{

    public class ResultsTable
```



```
{  
  
    public string Title { get; set; }  
  
    public DataTable Data { get; set; }  
  
    public ResultsTable ()  
    {  
        Title = "";  
  
        Data = new DataTable();  
    }  
  
    public ResultsTable(string title)  
    {  
        Title = title;  
  
        Data = new DataTable();  
    }  
  
    public ResultsTable(string title, DataTable data)  
    {  
        Title = title;  
  
        Data = data;  
    }  
}
```

```
}
```

```
public void AddColumn(string name, Type type)
```

```
{
```

```
    Data.Columns.Add(name, type);
```

```
}
```

```
public virtual string ToHTML ()
```

```
{
```

```
    //inicia a tabela
```

```
    string html = "<table>";
```

```
    //monta o cabeçalho
```

```
    html += "<tr>";
```

```
    for (int c = 0; c < Data.Columns.Count; c++)
```

```
    {
```

```
        Type tipo = Data.Columns[c].DataType;
```

```
        if (tipo == typeof(int) || tipo == typeof(double))
```

```
        {
```

```

        html += string.Format("<th style='font-weight:bold; text-align:right;'>{0}</th>",
Data.Columns[c].ColumnName);

    }

    else

    {

        html += string.Format("<th style='font-weight:bold;'>{0}</th>",
Data.Columns[c].ColumnName);

    }

}

html += "</tr>";

//monta as linhas

for (int l = 0; l < Data.Rows.Count; l++)

{

    html += "<tr>";

    for (int c = 0; c < Data.Columns.Count; c++)

    {

        Tipo tipo = Data.Columns[c].DataType;

        if (tipo == typeof(int))

```

```

        {
            html += string.Format("<td style='text-align:right;'>{0}</td>",
Data.Rows[l][c]);
        }
        else if (tipo == typeof(double))
        {
            html += string.Format("<td style='text-align:right;'>{0:0.000}</td>",
Data.Rows[l][c]);
        }
        else
        {
            html += string.Format("<td>{0}</td>", Data.Rows[l][c]);
        }
    }

    html += "</tr>";
}

//fecha a tabela

html += "</table>";

```

```
        return html;
    }

}

}
```

Oraculo\Tests\Helpers\ResultsTables.cs

```
?using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Oraculo.Tests.Helpers
{
    class ResultsTables
    {
        public List<object> Items { get; set; }
    }
}
```

```
public ResultsTables()

{

    Items = new List<object>();

}

public string ToHTML()

{

    string html = "";

    html += "<h2>Tabelas</h2>";

    int i = 1;

    foreach (ResultsTable item in Items)

    {

        if (item.Data.Rows.Count > 0)

        {

            html += string.Format("<h3>Tabela {0} - {1}</h3>", i, item.Title);

            html += item.ToHTML();

            i++;

        }

    }

}
```

```
    }  
  
    return html;  
  
    }  
  
    }  
  
}
```

Oraculo\Tests\Helpers\ResultsText.cs

```
?using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;  
  
namespace Oraculo.Tests.Helpers  
{  
  
    public class ResultsText  
  
    {
```

```
public string Title { get; set; }

public List<ResultsTextSection> Items { get; set; }

public ResultsText()

{

    Title = "";

    Items = new List<ResultsTextSection>();

}

public ResultsText(string title)

{

    Title = title;

    Items = new List<ResultsTextSection>();

}

public string ToHTML()

{

    string html = string.Format("<h2>{0}</h2>", Title);

    foreach (ResultsTextSection i in Items)
```



```
        {  
            if (i.Text.Count > 0) html += i.ToHTML();  
        }  
  
        return html;  
    }  
}  
}
```

Oraculo\Tests\Helpers\ResultsTextSection.cs

```
?using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;  
  
namespace Oraculo.Tests.Helpers  
{  
  
    public class ResultsTextSection
```

```
{  
  
    public string Title { get; set; }  
  
    public List<string> Text { get; set; }  
  
    public ResultsTextSection()  
  
    {  
  
        Title = "";  
  
        Text = new List<string>();  
  
    }  
  
    public ResultsTextSection(string title)  
  
    {  
  
        Title = title;  
  
        Text = new List<string>();  
  
    }  
  
    public string ToHTML()  
  
    {  
  
        string html = string.Format("<h3>{0}</h3>", Title);
```

```
        foreach (string p in Text)
        {
            html += string.Format("<p>{0}</p>", p);
        }

        return html;
    }
}
}
```

Oraculo\Tests\Helpers\SphericityTestsResultsTable.cs

```
?using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

namespace Oraculo.Tests.Helpers
```

```
{  
  
public class SphericityTestsResultsTable : ResultsTable  
  
{  
  
public SphericityTestsResultsTable()  
  
{  
  
AddColumn("Teste", typeof(string));  
  
AddColumn("p-valor", typeof(double));  
  
Title = "Testes de esfericidade";  
  
}  
  
public void AddRow(string teste, double pvalor)  
  
{  
  
DataRow dr = Data.NewRow();  
  
dr[0] = teste;  
  
dr[1] = pvalor;  
  
Data.Rows.Add(dr);  
  
}  
  
}
```

```
}  
  
}
```

Oraculo\Unit Tests\StatisticsTestsTest.cs

```
?using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;  
  
using NUnit.Framework;  
  
using System.Data;  
  
using Accord;  
  
using Accord.Statistics;  
  
using Oraculo.Global;  
  
namespace Oraculo.Tests  
  
{  
  
    [TestFixture]  
  
    class StatisticsTestsTest
```

```
{  
  
    [Test]  
  
    public void OneWayAnovaTest()  
  
    {  
  
        DataTable dt = new DataTable();  
  
        dt.Clear();  
  
        dt.Columns.Add("t1", typeof(double));  
  
        dt.Columns.Add("t2", typeof(double));  
  
        dt.Columns.Add("t3", typeof(double));  
  
        dt.Columns.Add("t4", typeof(double));  
  
  
        dt.Rows.Add(3129, 3200, 2800, 2600);  
  
        dt.Rows.Add(3000, 3300, 2900, 2700);  
  
        dt.Rows.Add(2865, 2975, 2985, 2600);  
  
        dt.Rows.Add(2890, 3150, 3050, 2765);  
  
  
        List<string> var = new List<string>();  
  
        var.Add("t1");  
  
        var.Add("t2");  
  
    }  
}
```

```

var.Add("t3");

var.Add("t4");

OneWayAnovaTest anova = new Tests.OneWayAnovaTest();

Assert.IsTrue(anova.Run(dt, var)); //testa se o teste está sendo realizado

Assert.AreEqual(Math.Round(12.72810749d, 5), Math.Round(anova.Stat, 5));
//verifica se a estatística F está correta

Assert.AreEqual(Math.Round(3.490294819, 5), Math.Round(anova.CriticalValue, 5));
//verifica se o F crítico está correto

Assert.AreEqual(Math.Round(0.000488715d, 5), Math.Round(anova.PValue, 5));
//verifica se o pvalor está correto

}

[Test]

public void BartletTest()

{

double[][] samples =

{

new double[] { 47, 48, 48, 49, 49, 49, 50, 50, 51 },

```

```

        new double[] { 48, 49, 49, 50, 50, 50, 51, 51, 52 },

        new double[] { 46, 47, 47, 48, 48, 48, 49, 49, 50 },

};

        Accord.Statistics.Testing.Bartlet b = new Accord.Statistics.Testing.Bartlet(samples,
2d, 24d, 1.5d);

        Assert.AreEqual(Math.Round(7.298371946, 5), Math.Round(b.M, 5));//c

        Assert.AreEqual(Math.Round(1.203703704, 5), Math.Round(b.C, 5)); //m

        Assert.AreEqual(Math.Round(6.063262847, 5), Math.Round(b.MC, 5)); //m/c

        Assert.IsFalse(b.AreEqual); //rejeita h0 porque B > QUI(K-1) -> as variâncias não são
iguais

    }

[Test]

public void OneWayAnovaBartletTest()

{

    DataTable dt = new DataTable();

    dt.Clear();

    dt.Columns.Add("g1", typeof(double));

```



```
dt.Columns.Add("g2", typeof(double));
```

```
dt.Columns.Add("g3", typeof(double));
```

```
dt.Rows.Add(48, 50, 51);
```

```
dt.Rows.Add(50, 49, 50);
```

```
dt.Rows.Add(49, 49, 52);
```

```
dt.Rows.Add(48, 50, 50);
```

```
dt.Rows.Add(47, 46, 53);
```

```
List<string> var = new List<string>();
```

```
var.Add("g1");
```

```
var.Add("g2");
```

```
var.Add("g3");
```

```
OneWayAnovaTest anova = new Tests.OneWayAnovaTest();
```

```
Assert.IsTrue(anova.Run(dt, var)); //testa se o teste está sendo realizado
```

```
Assert.AreEqual(Math.Round(6.03508772, 5), Math.Round(anova.Stat, 5)); //verifica  
se a estatística F está correta
```

```
Assert.AreEqual(Math.Round(0.01535366, 5), Math.Round(anova.PValue, 5));  
//verifica se o pvalor está correto
```

```
Assert.AreEqual(Math.Round(3.88529383, 5), Math.Round(anova.CriticalValue, 5));  
//verifica se a estatística F-Crítico está correta
```

```
Assert.AreEqual(Math.Round(4.129758057, 5), Math.Round(anova.BartletTest.M,  
5));//c
```

```
Assert.AreEqual(Math.Round(1.185185185, 5), Math.Round(anova.BartletTest.C, 5));  
//m
```

```
Assert.AreEqual(Math.Round(3.484483361, 5), Math.Round(anova.BartletTest.MC,  
5)); //m/c
```

```
Assert.IsTrue(anova.BartletTest.AreEqual); //não rejeita h0  
}
```

```
[Test]
```

```
public void BonferroniTest()
```

```
{
```

```
double[][] samples =
```

```
{
```

```
new double[] { 48, 49, 49, 50, 50, 50, 51, 51, 52 },
```

```
new double[] { 45, 45, 46, 47, 47, 47, 48, 48, 49 },
```

```
new double[] { 49, 46, 49, 50, 52, 51 },
```

```
};
```

```

List<string> var = new List<string>();

var.Add("g1");

var.Add("g2");

var.Add("g3");

    Accord.Statistics.Testing.Bonferroni b = new
Accord.Statistics.Testing.Bonferroni(samples, var);

    Assert.IsTrue(b.Run(samples, var)); //testa se o teste está sendo realizado

    //Assert.IsTrue(0.001, Math.Round(b.ResultsTable.Rows[0][1], 3)); //verifica se o
pvalor está correto

}

[Test]

public void RepeatedMeasuesAnovaTest()

{

    DataTable dt = new DataTable();

    dt.Clear();

    dt.Columns.Add("Grupo 1", typeof(double));

    dt.Columns.Add("Grupo 2", typeof(double));

```

```
dt.Columns.Add("Grupo 3", typeof(double));
```

```
dt.Rows.Add(47, 48, 46);
```

```
dt.Rows.Add(48, 49, 47);
```

```
dt.Rows.Add(48, 49, 47);
```

```
dt.Rows.Add(49, 50, 48);
```

```
dt.Rows.Add(49, 50, 48);
```

```
dt.Rows.Add(49, 50, 48);
```

```
dt.Rows.Add(50, 51, 49);
```

```
dt.Rows.Add(50, 51, 49);
```

```
dt.Rows.Add(51, 52, 51);
```

```
List<string> var = new List<string>();
```

```
var.Add("Grupo 1");
```

```
var.Add("Grupo 2");
```

```
var.Add("Grupo 3");
```

```
RepeatedMeasuresAnovaTest anova = new Tests.RepeatedMeasuresAnovaTest();
```

```
Assert.IsTrue(anova.Run(dt, var)); //testa se o teste está sendo realizado
```

```
Assert.AreEqual(0.0000, Math.Round(anova.PValue, 4)); //verifica se o pvalor está  
correto
```

```
Assert.AreEqual(217d, Math.Round(anova.Stat, 4)); //verifica se a estatística F está  
correta
```

```
}
```

```
[Test]
```

```
public void KruskalWallisTest()
```

```
{
```

```
    DataTable dt = new DataTable();
```

```
    dt.Clear();
```

```
    dt.Columns.Add("Grupo A", typeof(double));
```

```
    dt.Columns.Add("Grupo B", typeof(double));
```

```
    dt.Columns.Add("Grupo C", typeof(double));
```

```
    dt.Columns.Add("Grupo D", typeof(double));
```

```
    dt.Rows.Add(161, 128, 121, 109);
```

```
    dt.Rows.Add(160, 131, 98, 65);
```

```
    dt.Rows.Add(128, 125, 81, 97);
```

```
    dt.Rows.Add(168, 141, 128, 96);
```

```
dt.Rows.Add(131, 157, 91, 110);  
  
dt.Rows.Add(107, 132, 117, 67);  
  
dt.Rows.Add(219, 143, 136, 106);  
  
dt.Rows.Add(147, 112, 95, 102);  
  
dt.Rows.Add(175, 131, 105, 80);  
  
dt.Rows.Add(119, 128, 128, 96);  
  
dt.Rows.Add(190, 139, 90, null);  
  
dt.Rows.Add(203, 135, 151, null);  
  
dt.Rows.Add(107, 127, 129, null);  
  
dt.Rows.Add(153, 121, 95, null);  
  
dt.Rows.Add(128, 168, 112, null);  
  
dt.Rows.Add(209, null, null, null);  
  
dt.Rows.Add(142, null, null, null);  
  
dt.Rows.Add(124, null, null, null);
```

```
List<string> var = new List<string>();
```

```
var.Add("Grupo A");
```

```
var.Add("Grupo B");
```

```
var.Add("Grupo C");
```

```
var.Add("Grupo D");
```

```

KruskalWallisTest kwt = new KruskalWallisTest();

Assert.IsTrue(kwt.Run(dt, var)); //testa se o teste está sendo realizado

Assert.AreEqual(29.5865d, Math.Round(kwt.Stat, 4)); //verifica se a estatística de
teste está correta

Assert.AreEqual(1.68596E-06d, Math.Round(kwt.PValue, 11)); //verifica se o pvalor
está correto

}

[Test]

public void FriedmanTest()

{

    DataTable dt = new DataTable();

    dt.Clear();

    dt.Columns.Add("IR", typeof(double));

    dt.Columns.Add("Pré", typeof(double));

    dt.Columns.Add("Pós", typeof(double));

    dt.Columns.Add("3-M", typeof(double));

    dt.Rows.Add(21, 21, 16, 19);

```

```
dt.Rows.Add(30, 38, 10, 21);
```

```
dt.Rows.Add(38, 19, 15, 6);
```

```
dt.Rows.Add(43, 33, 30, 24);
```

```
dt.Rows.Add(35, 34, 25, 10);
```

```
dt.Rows.Add(40, 40, 31, 30);
```

```
dt.Rows.Add(27, 15, 11, 6);
```

```
dt.Rows.Add(18, 11, 4, 7);
```

```
dt.Rows.Add(31, 42, 23, 27);
```

```
dt.Rows.Add(21, 23, 21, 17);
```

```
dt.Rows.Add(18, 24, 16, 13);
```

```
dt.Rows.Add(28, 8, 5, 2);
```

```
dt.Rows.Add(40, 37, 31, 19);
```

```
dt.Rows.Add(35, 32, 12, 21);
```

```
List<string> var = new List<string>();
```

```
var.Add("IR");
```

```
var.Add("Pré");
```

```
var.Add("Pós");
```

```
var.Add("3-M");
```



```
FriedmanTest f = new FriedmanTest();

Assert.IsTrue(f.Run(dt, var)); //testa se o teste está sendo realizado

Assert.AreEqual(3.05735E-07d, Math.Round(f.PValue, 12)); //verifica se o pvalor está
correto

Assert.AreEqual(33.10714286d, Math.Round(f.Stat, 8)); //verifica se a estatística de
teste está correta

}
```

```
[Test]
```

```
public void LeveneTest()
```

```
{
```

```
    DataTable dt = new DataTable();
```

```
    dt.Clear();
```

```
    dt.Columns.Add("Grupo 1", typeof(double));
```

```
    dt.Columns.Add("Grupo 2", typeof(double));
```

```
    dt.Columns.Add("Grupo 3", typeof(double));
```

```
    dt.Rows.Add(47, 48, 46);
```

```
    dt.Rows.Add(48, 49, 47);
```

```
    dt.Rows.Add(48, 49, 47);
```

```
dt.Rows.Add(49, 50, 48);
```

```
dt.Rows.Add(49, 50, 48);
```

```
dt.Rows.Add(49, 50, 48);
```

```
dt.Rows.Add(50, 51, 49);
```

```
dt.Rows.Add(50, 51, 49);
```

```
dt.Rows.Add(51, 52, 51);
```

```
List<string> var = new List<string>();
```

```
var.Add("Grupo 1");
```

```
var.Add("Grupo 2");
```

```
var.Add("Grupo 3");
```

```
Oraculo.Tests.LevneTest b = new Oraculo.Tests.LevneTest();
```

```
Assert.IsTrue(b.Run(dt, var)); //testa se o teste está sendo realizado
```

```
Assert.IsTrue(b.AreEqual); //verifica se o pvalor está correto
```

```
}
```

```
[Test]
```

```
public void MauchlyTest()
```

```
{
```

```

/*

double[,] matriz = new double[,]

{

    { 0.346278274, 0.319151167, 0.299565069 },

    { 0.319151167, 0.331020615, 0.318104234 },

    { 0.299565069, 0.318104234, 0.323789964 }

};

    Accord.Math.Decompositions.EigenvalueDecomposition solver = new
Accord.Math.Decompositions.EigenvalueDecomposition(matriz, false, true);

    Assert.AreEqual(solver.RealEigenvalues[0], 0.958395);

    Assert.AreEqual(solver.RealEigenvalues[0], 0.0365834);

    Assert.AreEqual(solver.RealEigenvalues[0], 0.00611019);

*/

    DataTable dt = new DataTable();

    dt.Clear();

    dt.Columns.Add("Grupo 1", typeof(double));

    dt.Columns.Add("Grupo 2", typeof(double));

```

```
dt.Columns.Add("Grupo 3", typeof(double));
```

```
dt.Rows.Add(4.515, 4.982, 5.408);
```

```
dt.Rows.Add(4.235, 4.199, 4.683);
```

```
dt.Rows.Add(3.98, 4.364, 4.792);
```

```
dt.Rows.Add(4.357, 4.772, 5.098);
```

```
dt.Rows.Add(4.34, 4.953, 5.423);
```

```
dt.Rows.Add(4.586, 5.081, 5.362);
```

```
dt.Rows.Add(4.406, 4.56, 4.952);
```

```
dt.Rows.Add(4.243, 4.638, 4.952);
```

```
dt.Rows.Add(4.815, 5.174, 5.759);
```

```
dt.Rows.Add(3.84, 4.167, 4.669);
```

```
dt.Rows.Add(4.069, 4.31, 4.9);
```

```
dt.Rows.Add(4.283, 4.795, 5.27);
```

```
dt.Rows.Add(4.474, 4.887, 5.233);
```

```
dt.Rows.Add(4.455, 4.837, 5.111);
```

```
dt.Rows.Add(4.595, 4.079, 4.171);
```

```
dt.Rows.Add(3.731, 4.146, 4.613);
```

```
dt.Rows.Add(4.672, 4.884, 5.178);
```

```
dt.Rows.Add(2.962, 3.465, 3.756);
```

dt.Rows.Add(3.242, 3.931, 4.762);
dt.Rows.Add(4.359, 4.77, 5.019);
dt.Rows.Add(4.035, 4.638, 4.86);
dt.Rows.Add(3.534, 4.252, 4.675);
dt.Rows.Add(4.215, 4.689, 5.074);
dt.Rows.Add(2.789, 3.096, 3.304);
dt.Rows.Add(3.302, 3.895, 4.341);
dt.Rows.Add(3.339, 3.805, 4.206);
dt.Rows.Add(3.761, 4.355, 4.699);
dt.Rows.Add(4.489, 4.758, 5.154);
dt.Rows.Add(4.878, 5.141, 5.515);
dt.Rows.Add(4.88, 5.315, 5.626);
dt.Rows.Add(3.802, 4.16, 4.451);
dt.Rows.Add(4.459, 4.621, 5);
dt.Rows.Add(4.286, 4.819, 5.321);
dt.Rows.Add(4.058, 4.577, 4.807);
dt.Rows.Add(5.161, 5.429, 5.708);
dt.Rows.Add(3.81, 4.115, 4.419);
dt.Rows.Add(5.091, 5.618, 5.903);
dt.Rows.Add(4.134, 4.706, 5.274);

```
dt.Rows.Add(4.846, 5.356, 5.516);
```

```
dt.Rows.Add(4.11, 4.621, 4.95);
```

```
dt.Rows.Add(4.948, 5.394, 5.821);
```

```
dt.Rows.Add(4.363, 4.652, 5.035);
```

```
dt.Rows.Add(4.046, 4.651, 5.089);
```

```
dt.Rows.Add(3.756, 4.272, 4.588);
```

```
dt.Rows.Add(2.842, 3.254, 3.694);
```

```
dt.Rows.Add(4.326, 4.796, 5.091);
```

```
dt.Rows.Add(3.987, 4.553, 4.907);
```

```
dt.Rows.Add(3.495, 3.754, 3.972);
```

```
dt.Rows.Add(3.312, 3.447, 4.164);
```

```
dt.Rows.Add(3.028, 3.548, 3.969);
```

```
dt.Rows.Add(3.268, 3.829, 4.441);
```

```
dt.Rows.Add(3.559, 4.179, 4.698);
```

```
dt.Rows.Add(3.39, 3.734, 3.922);
```

```
dt.Rows.Add(3.723, 4.161, 4.548);
```

```
List<string> var = new List<string>();
```

```
var.Add("Grupo 1");
```

```
var.Add("Grupo 2");
```

```
var.Add("Grupo 3");

Oraculo.Tests.MauchlyTest m = new Oraculo.Tests.MauchlyTest();

Assert.IsTrue(m.Run(dt, var)); //testa se o teste está sendo realizado

Assert.AreEqual(0.0000, Math.Round(m.PValue, 4)); //verifica se o pvalor está
correto

}

[Test]

public void GGETest()

{

    DataTable dt = new DataTable();

    dt.Clear();

    dt.Columns.Add("Grupo 1", typeof(double));

    dt.Columns.Add("Grupo 2", typeof(double));

    dt.Columns.Add("Grupo 3", typeof(double));

    dt.Rows.Add(4.515, 4.982, 5.408);

    dt.Rows.Add(4.235, 4.199, 4.683);
```

dt.Rows.Add(3.98, 4.364, 4.792);

dt.Rows.Add(4.357, 4.772, 5.098);

dt.Rows.Add(4.34, 4.953, 5.423);

dt.Rows.Add(4.586, 5.081, 5.362);

dt.Rows.Add(4.406, 4.56, 4.952);

dt.Rows.Add(4.243, 4.638, 4.952);

dt.Rows.Add(4.815, 5.174, 5.759);

dt.Rows.Add(3.84, 4.167, 4.669);

dt.Rows.Add(4.069, 4.31, 4.9);

dt.Rows.Add(4.283, 4.795, 5.27);

dt.Rows.Add(4.474, 4.887, 5.233);

dt.Rows.Add(4.455, 4.837, 5.111);

dt.Rows.Add(4.595, 4.079, 4.171);

dt.Rows.Add(3.731, 4.146, 4.613);

dt.Rows.Add(4.672, 4.884, 5.178);

dt.Rows.Add(2.962, 3.465, 3.756);

dt.Rows.Add(3.242, 3.931, 4.762);

dt.Rows.Add(4.359, 4.77, 5.019);

dt.Rows.Add(4.035, 4.638, 4.86);

dt.Rows.Add(3.534, 4.252, 4.675);

dt.Rows.Add(4.215, 4.689, 5.074);
dt.Rows.Add(2.789, 3.096, 3.304);
dt.Rows.Add(3.302, 3.895, 4.341);
dt.Rows.Add(3.339, 3.805, 4.206);
dt.Rows.Add(3.761, 4.355, 4.699);
dt.Rows.Add(4.489, 4.758, 5.154);
dt.Rows.Add(4.878, 5.141, 5.515);
dt.Rows.Add(4.88, 5.315, 5.626);
dt.Rows.Add(3.802, 4.16, 4.451);
dt.Rows.Add(4.459, 4.621, 5);
dt.Rows.Add(4.286, 4.819, 5.321);
dt.Rows.Add(4.058, 4.577, 4.807);
dt.Rows.Add(5.161, 5.429, 5.708);
dt.Rows.Add(3.81, 4.115, 4.419);
dt.Rows.Add(5.091, 5.618, 5.903);
dt.Rows.Add(4.134, 4.706, 5.274);
dt.Rows.Add(4.846, 5.356, 5.516);
dt.Rows.Add(4.11, 4.621, 4.95);
dt.Rows.Add(4.948, 5.394, 5.821);
dt.Rows.Add(4.363, 4.652, 5.035);

```
dt.Rows.Add(4.046, 4.651, 5.089);  
  
dt.Rows.Add(3.756, 4.272, 4.588);  
  
dt.Rows.Add(2.842, 3.254, 3.694);  
  
dt.Rows.Add(4.326, 4.796, 5.091);  
  
dt.Rows.Add(3.987, 4.553, 4.907);  
  
dt.Rows.Add(3.495, 3.754, 3.972);  
  
dt.Rows.Add(3.312, 3.447, 4.164);  
  
dt.Rows.Add(3.028, 3.548, 3.969);  
  
dt.Rows.Add(3.268, 3.829, 4.441);  
  
dt.Rows.Add(3.559, 4.179, 4.698);  
  
dt.Rows.Add(3.39, 3.734, 3.922);  
  
dt.Rows.Add(3.723, 4.161, 4.548);
```

```
List<string> var = new List<string>();
```

```
var.Add("Grupo 1");
```

```
var.Add("Grupo 2");
```

```
var.Add("Grupo 3");
```

```
//converte para array double
```

```
double[][] dr = new double[3][];
```

```

//percorre cada coluna selecionada do dataset

foreach (string coluna in var)

{

    //cria variável para armazenar a lista

    List<double> dl = new List<double>();

    //percorre cada linha do dataset

    foreach (DataRow linha in dt.Rows)

    {

        //insere apenas as linhas que não tenham valores nulos

        if (!linha[coluna].IsNullOrEquivalent())

        {

            dl.Add(Convert.ToDouble(linha[coluna]));

        }

    }

    //converte para array

    dr[var.IndexOf(coluna)] = dl.ToArray();

}

```

```
    Accord.Statistics.Testing.GreenhouseGeisserEpsilon gge = new
Accord.Statistics.Testing.GreenhouseGeisserEpsilon(dr, var);

    Assert.AreEqual(0.66426364, Math.Round(gge.Stat, 8)); //verifica se o gge está
correto

}
```

[Test]

```
public void HFETest()

{

    DataTable dt = new DataTable();

    dt.Clear();

    dt.Columns.Add("Grupo 1", typeof(double));

    dt.Columns.Add("Grupo 2", typeof(double));

    dt.Columns.Add("Grupo 3", typeof(double));

    dt.Rows.Add(4.515, 4.982, 5.408);

    dt.Rows.Add(4.235, 4.199, 4.683);

    dt.Rows.Add(3.98, 4.364, 4.792);

    dt.Rows.Add(4.357, 4.772, 5.098);
```

dt.Rows.Add(4.34, 4.953, 5.423);

dt.Rows.Add(4.586, 5.081, 5.362);

dt.Rows.Add(4.406, 4.56, 4.952);

dt.Rows.Add(4.243, 4.638, 4.952);

dt.Rows.Add(4.815, 5.174, 5.759);

dt.Rows.Add(3.84, 4.167, 4.669);

dt.Rows.Add(4.069, 4.31, 4.9);

dt.Rows.Add(4.283, 4.795, 5.27);

dt.Rows.Add(4.474, 4.887, 5.233);

dt.Rows.Add(4.455, 4.837, 5.111);

dt.Rows.Add(4.595, 4.079, 4.171);

dt.Rows.Add(3.731, 4.146, 4.613);

dt.Rows.Add(4.672, 4.884, 5.178);

dt.Rows.Add(2.962, 3.465, 3.756);

dt.Rows.Add(3.242, 3.931, 4.762);

dt.Rows.Add(4.359, 4.77, 5.019);

dt.Rows.Add(4.035, 4.638, 4.86);

dt.Rows.Add(3.534, 4.252, 4.675);

dt.Rows.Add(4.215, 4.689, 5.074);

dt.Rows.Add(2.789, 3.096, 3.304);

dt.Rows.Add(3.302, 3.895, 4.341);
dt.Rows.Add(3.339, 3.805, 4.206);
dt.Rows.Add(3.761, 4.355, 4.699);
dt.Rows.Add(4.489, 4.758, 5.154);
dt.Rows.Add(4.878, 5.141, 5.515);
dt.Rows.Add(4.88, 5.315, 5.626);
dt.Rows.Add(3.802, 4.16, 4.451);
dt.Rows.Add(4.459, 4.621, 5);
dt.Rows.Add(4.286, 4.819, 5.321);
dt.Rows.Add(4.058, 4.577, 4.807);
dt.Rows.Add(5.161, 5.429, 5.708);
dt.Rows.Add(3.81, 4.115, 4.419);
dt.Rows.Add(5.091, 5.618, 5.903);
dt.Rows.Add(4.134, 4.706, 5.274);
dt.Rows.Add(4.846, 5.356, 5.516);
dt.Rows.Add(4.11, 4.621, 4.95);
dt.Rows.Add(4.948, 5.394, 5.821);
dt.Rows.Add(4.363, 4.652, 5.035);
dt.Rows.Add(4.046, 4.651, 5.089);
dt.Rows.Add(3.756, 4.272, 4.588);

```
dt.Rows.Add(2.842, 3.254, 3.694);  
  
dt.Rows.Add(4.326, 4.796, 5.091);  
  
dt.Rows.Add(3.987, 4.553, 4.907);  
  
dt.Rows.Add(3.495, 3.754, 3.972);  
  
dt.Rows.Add(3.312, 3.447, 4.164);  
  
dt.Rows.Add(3.028, 3.548, 3.969);  
  
dt.Rows.Add(3.268, 3.829, 4.441);  
  
dt.Rows.Add(3.559, 4.179, 4.698);  
  
dt.Rows.Add(3.39, 3.734, 3.922);  
  
dt.Rows.Add(3.723, 4.161, 4.548);
```

```
List<string> var = new List<string>();
```

```
var.Add("Grupo 1");
```

```
var.Add("Grupo 2");
```

```
var.Add("Grupo 3");
```

```
//converte para array double
```

```
double[][] dr = new double[3][];
```

```
//percorre cada coluna selecionada do dataset
```

```
foreach (string coluna in var)
{
    //cria variável para armazenar a lista

    List<double> dl = new List<double>();

    //percorre cada linha do dataset

    foreach (DataRow linha in dt.Rows)
    {
        //insere apenas as linhas que não tenham valores nulos

        if (!linha[coluna].IsNullOrEquivalent())
        {
            dl.Add(Convert.ToDouble(linha[coluna]));
        }
    }

    //converte para array

    dr[var.IndexOf(coluna)] = dl.ToArray();
}
```



```
    Accord.Statistics.Testing.HuynhFeldtEpsilon hfe = new  
    Accord.Statistics.Testing.HuynhFeldtEpsilon(3, 54, 0.66426364d);
```

```
    Assert.AreEqual(0.67484503, Math.Round(hfe.Stat, 8)); //verifica se o hfe está correto  
}
```

```
[Test]
```

```
public void TwoSampleTTestTest()
```

```
{
```

```
    DataTable dt = new DataTable();
```

```
    dt.Clear();
```

```
    dt.Columns.Add("Grupo 1", typeof(double));
```

```
    dt.Columns.Add("Grupo 2", typeof(double));
```

```
    dt.Columns.Add("Grupo 3", typeof(double));
```

```
    dt.Rows.Add(47, 48, 46);
```

```
    dt.Rows.Add(48, 49, 47);
```

```
    dt.Rows.Add(48, 49, 47);
```

```
    dt.Rows.Add(49, 50, 48);
```

```
    dt.Rows.Add(49, 50, 48);
```

```
dt.Rows.Add(49, 50, 48);
```

```
dt.Rows.Add(50, 51, 49);
```

```
dt.Rows.Add(50, 51, 49);
```

```
dt.Rows.Add(51, 52, 51);
```

```
List<string> var = new List<string>();
```

```
var.Add("Grupo 1");
```

```
var.Add("Grupo 2");
```

```
var.Add("Grupo 3");
```

```
Oraculo.Tests.TwoSampleTTestTest b = new Oraculo.Tests.TwoSampleTTestTest();
```

```
Assert.IsTrue(b.Run(dt, var[0], var[1])); //testa se o teste está sendo realizado
```

```
Assert.IsTrue(b.AreEqual); //verifica se o pvalor está correto
```

```
}
```

```
[Test]
```

```
public void MannWhitneyTest()
```

```
{
```

```
    DataTable dt = new DataTable();
```

```
    dt.Clear();
```

```
dt.Columns.Add("Grupo A", typeof(double));
```

```
dt.Columns.Add("Grupo D", typeof(double));
```

```
dt.Rows.Add(47,48);
```

```
dt.Rows.Add(48, 48);
```

```
dt.Rows.Add(48, 48);
```

```
dt.Rows.Add(49, 48);
```

```
dt.Rows.Add(49, 48);
```

```
dt.Rows.Add(49, 48);
```

```
dt.Rows.Add(50, 48);
```

```
dt.Rows.Add(50, 51);
```

```
dt.Rows.Add(51, 51);
```

```
List<string> var = new List<string>();
```

```
var.Add("Grupo A");
```

```
var.Add("Grupo D");
```

```
double[][] dr = dt.ToBidimensionalDoubleArray(var);
```

```

MannWhitney m = new MannWhitney(dr,var);

Assert.AreEqual(0.368, Math.Round(m.PValue, 3));

Assert.AreEqual(31, Math.Round(m.Stat, 0));

//Assert.AreEqual(0, Math.Round(m.CriticalValue, 0));

}

```

[Test]

```

public void WelchTest()

{

//double[][] samples =

//{

// new double[] { 47, 48, 48, 49, 49, 49, 50, 50, 51 },

// new double[] { 46, 44, 37, 68, 62, 41, 55, 49, 48 },

// new double[] { 49, 46, 49, 50, 52, 51 },

//};

//List<string> var = new List<string>();

//var.Add("A");

//var.Add("G");

```

```

//var.Add("E");

//BASEADO NO EXEMPLO EM https://en.wikipedia.org/wiki/Welch%27s\_t-test

double[][] samples =

{

    new double[] { 17.2, 20.9, 22.6, 18.1, 21.7, 21.4, 23.5, 24.2, 14.7, 21.8 },

    new double[] { 21.5, 22.8, 21, 23, 21.6, 23.6, 22.5, 20.7, 23.4, 21.8, 20.7, 21.7, 21.5,
22.5, 23.6, 21.5, 22.5, 23.5, 21.5, 21.8 },

};

List<string> var = new List<string>();

var.Add("A1");

var.Add("A2");

 Accord.Statistics.Testing.Welch w = new Accord.Statistics.Testing.Welch(samples,
var);

Assert.AreEqual(-1.57d, Math.Round(w.Stat,2)); //t

Assert.AreEqual(0.149d, Math.Round(w.PValue,3)); //p-valor

//Assert.IsTrue(w.Run(samples, var)); //testa se o teste está sendo realizado

```

```
//Assert.IsTrue(0.001, Math.Round(b.ResultsTable.Rows[0][1], 3)); //verifica se o  
pvalor está correto
```

```
}
```

```
[Test]
```

```
public void FriedmanComEmpatesTest()
```

```
{
```

```
    DataTable dt = new DataTable();
```

```
    dt.Clear();
```

```
    dt.Columns.Add("Grupo A", typeof(double));
```

```
    dt.Columns.Add("Grupo B", typeof(double));
```

```
    dt.Columns.Add("Grupo D", typeof(double));
```

```
    dt.Rows.Add(47, 48, 48);
```

```
    dt.Rows.Add(48, 49, 48);
```

```
    dt.Rows.Add(48, 49, 48);
```

```
    dt.Rows.Add(49, 50, 48);
```

```
    dt.Rows.Add(49, 50, 48);
```

```
    dt.Rows.Add(49, 50, 48);
```

```
dt.Rows.Add(50, 51, 48);
```

```
dt.Rows.Add(50, 51, 51);
```

```
dt.Rows.Add(51, 52, 51);
```

```
List<string> var = new List<string>();
```

```
var.Add("Grupo A");
```

```
var.Add("Grupo B");
```

```
var.Add("Grupo D");
```

```
FriedmanTest f = new FriedmanTest();
```

```
Assert.IsTrue(f.Run(dt, var)); //testa se o teste está sendo realizado
```

```
Assert.AreEqual(0.001918d, Math.Round(f.PValue, 6)); //verifica se o pvalor está  
correto
```

```
Assert.AreEqual(12.452d, Math.Round(f.Stat, 3)); //verifica se a estatística de teste  
está correta
```

```
}
```

```
}
```

```
}
```

```
Oraculo\Global>EditForm.resx
```

```
?<?xml version="1.0" encoding="utf-8"?>
```

```
<root>
```

```
<!--
```

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```



```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"  
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET  
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
```

```
<xsd:element name="root" msdata:IsDataSet="true">
```

```
<xsd:complexType>
```

```
<xsd:choice maxOccurs="unbounded">
```

```
<xsd:element name="metadata">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element name="value" type="xsd:string" minOccurs="0" />
```

```
</xsd:sequence>
```

```
<xsd:attribute name="name" use="required" type="xsd:string" />
```

```
<xsd:attribute name="type" type="xsd:string" />
```

```
<xsd:attribute name="mimetype" type="xsd:string" />
```

```
<xsd:attribute ref="xml:space" />
```

```
</xsd:complexType>
```

```

</xsd:element>

<xsd:element name="assembly">

  <xsd:complexType>

    <xsd:attribute name="alias" type="xsd:string" />

    <xsd:attribute name="name" type="xsd:string" />

  </xsd:complexType>

</xsd:element>

<xsd:element name="data">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

      <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

  </xsd:complexType>

```

```
</xsd:element>

<xsd:element name="resheader">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" />

  </xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

  <value>text/microsoft-resx</value>

</resheader>

<resheader name="version">

  <value>2.0</value>

</resheader>
```

```
<resheader name="reader">
```

```
  <value>System.Resources.ResXResourceReader, System.Windows.Forms,  
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
```

```
</resheader>
```

```
<resheader name="writer">
```

```
  <value>System.Resources.ResXResourceWriter, System.Windows.Forms,  
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
```

```
</resheader>
```

```
</root>
```

Oraculo\Global\Info.resx

```
?<?xml version="1.0" encoding="utf-8"?>
```

```
<root>
```

```
<!--
```

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes

associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"  
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET  
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

</data>

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: `System.Runtime.Serialization.Formatters.Binary.BinaryFormatter`

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: `System.Runtime.Serialization.Formatters.Soap.SoapFormatter`

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a `System.ComponentModel.TypeConverter`

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
```

```
<xsd:element name="root" msdata:IsDataSet="true">
```

```
<xsd:complexType>
```

```
<xsd:choice maxOccurs="unbounded">
```

```
<xsd:element name="metadata">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" />

    </xsd:sequence>

    <xsd:attribute name="name" use="required" type="xsd:string" />

    <xsd:attribute name="type" type="xsd:string" />

    <xsd:attribute name="mimetype" type="xsd:string" />

    <xsd:attribute ref="xml:space" />

  </xsd:complexType>

</xsd:element>

<xsd:element name="assembly">

  <xsd:complexType>

    <xsd:attribute name="alias" type="xsd:string" />

    <xsd:attribute name="name" type="xsd:string" />

  </xsd:complexType>

</xsd:element>

<xsd:element name="data">

  <xsd:complexType>

    <xsd:sequence>
```

```

        <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

        <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

        </xsd:sequence>

        <xsd:attribute name="name" type="xsd:string" use="required" />

    </xsd:complexType>

</xsd:element>

```

```
</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

  <value>text/microsoft-resx</value>

</resheader>

<resheader name="version">

  <value>2.0</value>

</resheader>

<resheader name="reader">

  <value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

  <value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<metadata name="toolTip1.TrayLocation" type="System.Drawing.Point, System.Drawing,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">

  <value>17, 17</value>
```

</metadata>

</root>

Oraculo\Global\MoveBetwenListboxesControl.resx

?<?xml version="1.0" encoding="utf-8"?>

<root>

<!--

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

<resheader name="resmimetype">text/microsoft-resx</resheader>

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"  
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET  
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a

type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
```

```
<xsd:element name="root" msdata:IsDataSet="true">
```

```
<xsd:complexType>
```

```
<xsd:choice maxOccurs="unbounded">
```

```
<xsd:element name="metadata">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element name="value" type="xsd:string" minOccurs="0" />
```

```
</xsd:sequence>
```

```
<xsd:attribute name="name" use="required" type="xsd:string" />
```



```

    <xsd:attribute name="type" type="xsd:string" />

    <xsd:attribute name="mimetype" type="xsd:string" />

    <xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="assembly">

    <xsd:complexType>

        <xsd:attribute name="alias" type="xsd:string" />

        <xsd:attribute name="name" type="xsd:string" />

    </xsd:complexType>

</xsd:element>

<xsd:element name="data">

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

            <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

        </xsd:sequence>

        <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

```

```

    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

  </xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" />

  </xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

  <value>text/microsoft-resx</value>

```

</resheader>

<resheader name="version">

<value>2.0</value>

</resheader>

<resheader name="reader">

<value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

<value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

</root>

Oraculo\Global\SectionHeader.resx

?<?xml version="1.0" encoding="utf-8"?>

<root>

<!--

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format

that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />

<xsd:element name="root" msdata:IsDataSet="true">

  <xsd:complexType>

    <xsd:choice maxOccurs="unbounded">

      <xsd:element name="metadata">

        <xsd:complexType>

          <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" />

          </xsd:sequence>

          <xsd:attribute name="name" use="required" type="xsd:string" />

          <xsd:attribute name="type" type="xsd:string" />

          <xsd:attribute name="mimetype" type="xsd:string" />

          <xsd:attribute ref="xml:space" />

        </xsd:complexType>

      </xsd:element>

    <xsd:element name="assembly">

      <xsd:complexType>

        <xsd:attribute name="alias" type="xsd:string" />

        <xsd:attribute name="name" type="xsd:string" />

      </xsd:complexType>

    </xsd:element>

  </xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:element>
```

```

</xsd:element>

<xsd:element name="data">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

      <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

  </xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

```



```
</xsd:sequence>

<xsd:attribute name="name" type="xsd:string" use="required" />

</xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

  <value>text/microsoft-resx</value>

</resheader>

<resheader name="version">

  <value>2.0</value>

</resheader>

<resheader name="reader">

  <value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

  <value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
```

```
</resheader>
```

```
</root>
```

Oraculo\Global\SideNavigationMenu.resx

```
?<?xml version="1.0" encoding="utf-8"?>
```

```
<root>
```

```
<!--
```

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"  
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET  
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a

type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
```

```
<xsd:element name="root" msdata:IsDataSet="true">
```

```
<xsd:complexType>
```

```
<xsd:choice maxOccurs="unbounded">
```

```
<xsd:element name="metadata">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element name="value" type="xsd:string" minOccurs="0" />
```

```
</xsd:sequence>
```

```
<xsd:attribute name="name" use="required" type="xsd:string" />
```

```

<xsd:attribute name="type" type="xsd:string" />

<xsd:attribute name="mimetype" type="xsd:string" />

<xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="assembly">

  <xsd:complexType>

    <xsd:attribute name="alias" type="xsd:string" />

    <xsd:attribute name="name" type="xsd:string" />

  </xsd:complexType>

</xsd:element>

<xsd:element name="data">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

      <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

```

```

    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

        </xsd:sequence>

        <xsd:attribute name="name" type="xsd:string" use="required" />

    </xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

    <value>text/microsoft-resx</value>

```

</resheader>

<resheader name="version">

<value>2.0</value>

</resheader>

<resheader name="reader">

<value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

<value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<metadata name="toolTip1.TrayLocation" type="System.Drawing.Point, System.Drawing,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">

<value>17, 17</value>

</metadata>

</root>

Oraculo\Global\WizardControl.resx

?<?xml version="1.0" encoding="utf-8"?>

<root>

<!--

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
  <value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"  
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
  <value>[base64 mime encoded string representing a byte array form of the .NET  
Framework object]</value>
```

```
  <comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the

ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
```

```
<xsd:element name="root" msdata:IsDataSet="true">
```

```
<xsd:complexType>
```

```
<xsd:choice maxOccurs="unbounded">
```

```
<xsd:element name="metadata">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element name="value" type="xsd:string" minOccurs="0" />
```

```
</xsd:sequence>
```

```
<xsd:attribute name="name" use="required" type="xsd:string" />
```

```
<xsd:attribute name="type" type="xsd:string" />
```

```
<xsd:attribute name="mimetype" type="xsd:string" />
```

```
<xsd:attribute ref="xml:space" />
```

```
</xsd:complexType>
```

```
</xsd:element>
```

```
<xsd:element name="assembly">
```

```

<xsd:complexType>

  <xsd:attribute name="alias" type="xsd:string" />

  <xsd:attribute name="name" type="xsd:string" />

</xsd:complexType>

</xsd:element>

<xsd:element name="data">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

      <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

  </xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

```

```

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

        </xsd:sequence>

        <xsd:attribute name="name" type="xsd:string" use="required" />

    </xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

    <value>text/microsoft-resx</value>

</resheader>

<resheader name="version">

    <value>2.0</value>

</resheader>

<resheader name="reader">

    <value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

```

```
</resheader>
```

```
<resheader name="writer">
```

```
  <value>System.Resources.ResXResourceWriter, System.Windows.Forms,  
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
```

```
</resheader>
```

```
<metadata name="toolTip1.TrayLocation" type="System.Drawing.Point, System.Drawing,  
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">
```

```
  <value>17, 17</value>
```

```
</metadata>
```

```
</root>
```

Oraculo\Properties\Resources.resx

```
?<?xml version="1.0" encoding="utf-8"?>
```

```
<root>
```

```
<!--
```

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format

that is mostly human readable. The generation and parsing of the

various data types are done through the TypeConverter classes

associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"  
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET  
Framework object]</value>
```


<comment>This is a comment</comment>

</data>

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
```

```
<xsd:element name="root" msdata:IsDataSet="true">
```

```
<xsd:complexType>
```

```
<xsd:choice maxOccurs="unbounded">

  <xsd:element name="metadata">

    <xsd:complexType>

      <xsd:sequence>

        <xsd:element name="value" type="xsd:string" minOccurs="0" />

      </xsd:sequence>

      <xsd:attribute name="name" use="required" type="xsd:string" />

      <xsd:attribute name="type" type="xsd:string" />

      <xsd:attribute name="mimetype" type="xsd:string" />

      <xsd:attribute ref="xml:space" />

    </xsd:complexType>

  </xsd:element>

  <xsd:element name="assembly">

    <xsd:complexType>

      <xsd:attribute name="alias" type="xsd:string" />

      <xsd:attribute name="name" type="xsd:string" />

    </xsd:complexType>

  </xsd:element>

  <xsd:element name="data">

    <xsd:complexType>
```

```

    <xsd:sequence>

        <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

        <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

        </xsd:sequence>

        <xsd:attribute name="name" type="xsd:string" use="required" />

    </xsd:complexType>

```

```
</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

  <value>text/microsoft-resx</value>

</resheader>

<resheader name="version">

  <value>2.0</value>

</resheader>

<resheader name="reader">

  <value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

  <value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<assembly alias="System.Windows.Forms" name="System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
```

<data name="next_flat" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\006-next.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="information" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\information.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="home" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\home.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="import_excel" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\import_excel.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="delete_column_flat" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\delete-column.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="home_flat" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\002-home.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="clipboard" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\clipboard.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="find" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\find.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="report_flat" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\file.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="DataByLines" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\DataByLines.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="DataByColumns" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\DataByColumns.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="cancel" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\cancel.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="logo" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\logo aprovada.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="database_flat" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\001-edit-table.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="ok" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\ok.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="analysis" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\analysis.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="previous" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\previous.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="import_excel_flat" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\005-xls.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="previous_flat" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\007-previous.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="add" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\add.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="import" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\import.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="delete" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\delete.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="next" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\next.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="import_csv" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\import_csv.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="add_column_flat" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\add-column.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="oraculo" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\oraculo.ico;System.Drawing.Icon, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="database" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\database.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="analysis_flat" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\003-magic-wand.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="type_data" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\type_data.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="menu_flat" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\009-menu.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="delete_row_flat" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\008-delete-row.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="about_flat" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\about.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="import_csv_flat" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\004-csv.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

<data name="video_print" type="System.Resources.ResXFileRef, System.Windows.Forms">

<value>..\Resources\video_print.png;System.Drawing.Bitmap, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>

</data>

</root>

Oraculo\Screens\About.resx

?<?xml version="1.0" encoding="utf-8"?>

<root>

<!--

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format

that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />

<xsd:element name="root" msdata:IsDataSet="true">

  <xsd:complexType>

    <xsd:choice maxOccurs="unbounded">

      <xsd:element name="metadata">

        <xsd:complexType>

          <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" />

          </xsd:sequence>

          <xsd:attribute name="name" use="required" type="xsd:string" />

          <xsd:attribute name="type" type="xsd:string" />

          <xsd:attribute name="mimetype" type="xsd:string" />

          <xsd:attribute ref="xml:space" />

        </xsd:complexType>

      </xsd:element>

    <xsd:element name="assembly">

      <xsd:complexType>

        <xsd:attribute name="alias" type="xsd:string" />

        <xsd:attribute name="name" type="xsd:string" />

      </xsd:complexType>

    </xsd:element>

  </xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:element>
```



```

</xsd:element>

<xsd:element name="data">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

      <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

  </xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

```

```
</xsd:sequence>

<xsd:attribute name="name" type="xsd:string" use="required" />

</xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

  <value>text/microsoft-resx</value>

</resheader>

<resheader name="version">

  <value>2.0</value>

</resheader>

<resheader name="reader">

  <value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

  <value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
```

```
</resheader>
```

```
</root>
```

Oraculo\Screens\AddColumnForm.resx

```
?<?xml version="1.0" encoding="utf-8"?>
```

```
<root>
```

```
<!--
```

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"  
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET  
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a

type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatter.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
```

```
<xsd:element name="root" msdata:IsDataSet="true">
```

```
<xsd:complexType>
```

```
<xsd:choice maxOccurs="unbounded">
```

```
<xsd:element name="metadata">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element name="value" type="xsd:string" minOccurs="0" />
```

```
</xsd:sequence>
```

```
<xsd:attribute name="name" use="required" type="xsd:string" />
```

```

    <xsd:attribute name="type" type="xsd:string" />

    <xsd:attribute name="mimetype" type="xsd:string" />

    <xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="assembly">

    <xsd:complexType>

        <xsd:attribute name="alias" type="xsd:string" />

        <xsd:attribute name="name" type="xsd:string" />

    </xsd:complexType>

</xsd:element>

<xsd:element name="data">

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

            <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

        </xsd:sequence>

        <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

```

```

    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

  </xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" />

  </xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

  <value>text/microsoft-resx</value>

```


</resheader>

<resheader name="version">

<value>2.0</value>

</resheader>

<resheader name="reader">

<value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

<value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<metadata name="toolTip1.TrayLocation" type="System.Drawing.Point, System.Drawing,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">

<value>17, 17</value>

</metadata>

<assembly alias="System.Drawing" name="System.Drawing, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />

<data name="\$this.Icon" type="System.Drawing.Icon, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">

<value>

AAABAAEAEBaQaAAAAACoAQAAFGAAACgAAAAQAAAAIAAAAAEABAAAAA
AwAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAACAAACAAAAAgIAAgAAAAIAAgACAgAAAgICAAMDAwAAAAP8A
AP8AAAD//wD/AAAA/wD/AP//

AAD///8AAA
AAAAAAAAAAAAAAAAAAAAAAAAA

AAA
AAAAAAAAAAAAAAAAAAAAAAAAA

AAA
AAAAAAAAAAAAAAAAAAAAAAAAA

AAA
AAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMP/AADf/wAA3/8AAN//AADf/wAA3/
8AAN//AADf/wAA3/8AAN//

AADf/wAA3/8AAN//AADf/wAA3/8AAMB/AAA=

</value>

</data>

</root>

Oraculo\Screens\Analyze.resx

```
?<?xml version="1.0" encoding="utf-8"?>
```

```
<root>
```

```
<!--
```

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"  
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET  
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support

text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
```

```
<xsd:element name="root" msdata:IsDataSet="true">
```

```
<xsd:complexType>
```

```
<xsd:choice maxOccurs="unbounded">
```

```
<xsd:element name="metadata">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element name="value" type="xsd:string" minOccurs="0" />
```

```
</xsd:sequence>
```

```
<xsd:attribute name="name" use="required" type="xsd:string" />
```

```
<xsd:attribute name="type" type="xsd:string" />
```

```

    <xsd:attribute name="mimetype" type="xsd:string" />

    <xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="assembly">

    <xsd:complexType>

        <xsd:attribute name="alias" type="xsd:string" />

        <xsd:attribute name="name" type="xsd:string" />

    </xsd:complexType>

</xsd:element>

<xsd:element name="data">

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

            <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

        </xsd:sequence>

        <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

        <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

```

```

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

        </xsd:sequence>

        <xsd:attribute name="name" type="xsd:string" use="required" />

    </xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

    <value>text/microsoft-resx</value>

</resheader>

```


<resheader name="version">

<value>2.0</value>

</resheader>

<resheader name="reader">

<value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

<value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<metadata name="toolTip1.TrayLocation" type="System.Drawing.Point, System.Drawing,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">

<value>17, 17</value>

</metadata>

<assembly alias="System.Drawing" name="System.Drawing, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />

<data name="infoPareado.BackgroundImage" type="System.Drawing.Bitmap,
System.Drawing" mimetype="application/x-microsoft.net.object.bytearray.base64">

<value>

iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAIAAAAF8/9hAAAABGdBTUEAAK
/INwWK6QAAABI0RVh0U29m

dHdhcmUAQWRvYmUgSW1hZ2VSZWVkeXhJZTwAAAKoSURBVDhPpZPtT1JRHMf9
W/RP6N7etDZ90VybDZK0

F5k1TBO1la7CbLBKMRGfAAVsCIaaD5RPUJliJHAI+QhNPXmwAza2uoNZC3LbwdaqN
XaWi8+d/ece77f

3/ne3zkZAP6L3ybyNStZx1QhUZ4iyB5p8cZzZJ54tmyBzZa5RYdvM1m/rt834KiXeUflgVhF
rx+99ghG

mc0UemsYfPVT0KLZGC228vZq0i+kKi+31ZdomVyFxrtdz3QEEuNL3Bpehty0CqNjA42jI
dCvEwn64kTa

JPXIUwYzc9sWo0mx6VkmliK6NhDAkOs1hpwbuGrwQ9gfgGF2HZL7ftBlxihdOpyZNsiV
L4rLtV6YPW9R

NxCCkES4rPPH4+ftFNU6D6p7PBDe9WLgSRinmh+DKuoXpw1y2jys3hqB0sTiSq8PNTovq
rUe9NnXYSBU

djM/0LhIrADU0yugC3Vs2uCQhNkaX3iDuj4/qfQcVXfcZDFDPgE7BIHKifLOOYIDFzRO
GJkwKF7XVIKb

MqDF9q0x92Yqa7JShdpFRC6EIh+Q+LSNsg47StttOEcQKGwYmScGXMVeAyurtbKQPlh
ClcYNQZcT55Vz

eBF+Twy+oEQ2gxKSm99sQW0PA6UICPp4x24EutYiLpbbMeSioErFQCB3wOLZwNdvO6
kI/IfvUCwxEOzo

fhhC4fVRYtC++xPpS2OZtMAYvTnig8G6hjK5DfyWaZyRTqG4wYTT9RMoqh9H64iXtHe
ebF8apbnS3TYm

ofiDvAMF+oRo0AP9zAqEWifZ8iOcbTSjRmWD2hSAsI8BzWIIEPH+g5Q2KdTzqILu2Mk
bk1BOLeGekyWs

ocMcRH6dERSnMUZzm/58IH9Cn1BnUbxOEZWvYCluc5ziNMVJRZYgOsh+Ptl+neQ8R0
4hsvsalKMmgAA

AABJRU5ErkJggg==

</value>

</data>

<metadata name="\$this.TrayHeight" type="System.Int32, mscorlib, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089">

<value>60</value>

</metadata>

</root>

Oraculo\Screens\CSVFileImportWizardControl.resx

?<?xml version="1.0" encoding="utf-8"?>

<root>

<!--

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>

</data>

<data name="Icon1" type="System.Drawing.Icon, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">

    <value>[base64 mime encoded string representing a byte array form of the .NET
Framework object]</value>

    <comment>This is a comment</comment>

</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
```

```
<xsd:element name="root" msdata:IsDataSet="true">
```

```
<xsd:complexType>
```

```
<xsd:choice maxOccurs="unbounded">
```

```
<xsd:element name="metadata">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element name="value" type="xsd:string" minOccurs="0" />
```

```
</xsd:sequence>
```

```
<xsd:attribute name="name" use="required" type="xsd:string" />
```

```
<xsd:attribute name="type" type="xsd:string" />
```

```
<xsd:attribute name="mimetype" type="xsd:string" />
```

```
<xsd:attribute ref="xml:space" />
```

```
</xsd:complexType>
```

```
</xsd:element>
```

```
<xsd:element name="assembly">
```

```
<xsd:complexType>
```

```
<xsd:attribute name="alias" type="xsd:string" />
```

```

    <xsd:attribute name="name" type="xsd:string" />

</xsd:complexType>

</xsd:element>

<xsd:element name="data">

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

            <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

        </xsd:sequence>

        <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

        <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

        <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

        <xsd:attribute ref="xml:space" />

    </xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

    <xsd:complexType>

        <xsd:sequence>

```



```

        <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" />

</xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

    <value>text/microsoft-resx</value>

</resheader>

<resheader name="version">

    <value>2.0</value>

</resheader>

<resheader name="reader">

    <value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

```

```
<value>System.Resources.ResXResourceWriter, System.Windows.Forms,  
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
```

```
</resheader>
```

```
</root>
```

Oraculo\Screens\DataView.resx

```
?<?xml version="1.0" encoding="utf-8"?>
```

```
<root>
```

```
<!--
```

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"  
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET  
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple

name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture. Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
```

```
<xsd:element name="root" msdata:IsDataSet="true">
```

```
<xsd:complexType>
```

```
<xsd:choice maxOccurs="unbounded">
```

```
<xsd:element name="metadata">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element name="value" type="xsd:string" minOccurs="0" />
```

```

</xsd:sequence>

<xsd:attribute name="name" use="required" type="xsd:string" />

<xsd:attribute name="type" type="xsd:string" />

<xsd:attribute name="mimetype" type="xsd:string" />

<xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="assembly">

  <xsd:complexType>

    <xsd:attribute name="alias" type="xsd:string" />

    <xsd:attribute name="name" type="xsd:string" />

  </xsd:complexType>

</xsd:element>

<xsd:element name="data">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"

/>

      <xsd:element name="comment" type="xsd:string" minOccurs="0"

msdata:Ordinal="2" />

    </xsd:sequence>

```

```

    <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

        </xsd:sequence>

        <xsd:attribute name="name" type="xsd:string" use="required" />

    </xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

```

```
<value>text/microsoft-resx</value>

</resheader>

<resheader name="version">

  <value>2.0</value>

</resheader>

<resheader name="reader">

  <value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

  <value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<metadata name="contextMenuColuna.TrayLocation" type="System.Drawing.Point,
System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">

  <value>17, 17</value>

</metadata>

<metadata name="contextMenuLinha.TrayLocation" type="System.Drawing.Point,
System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">

  <value>172, 17</value>

</metadata>
```


</root>

Oraculo\Screens\ExcelFileImportWizardControl.resx

?<?xml version="1.0" encoding="utf-8"?>

<root>

<!--

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

<resheader name="resmimetype">text/microsoft-resx</resheader>

<resheader name="version">2.0</resheader>

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"  
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET  
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support

text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
```

```
<xsd:element name="root" msdata:IsDataSet="true">
```

```
<xsd:complexType>
```

```
<xsd:choice maxOccurs="unbounded">
```

```
<xsd:element name="metadata">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element name="value" type="xsd:string" minOccurs="0" />
```

```
</xsd:sequence>
```

```
<xsd:attribute name="name" use="required" type="xsd:string" />
```

```
<xsd:attribute name="type" type="xsd:string" />
```

```

    <xsd:attribute name="mimetype" type="xsd:string" />

    <xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="assembly">

    <xsd:complexType>

        <xsd:attribute name="alias" type="xsd:string" />

        <xsd:attribute name="name" type="xsd:string" />

    </xsd:complexType>

</xsd:element>

<xsd:element name="data">

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

            <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

        </xsd:sequence>

        <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

        <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

```

```

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

        </xsd:sequence>

        <xsd:attribute name="name" type="xsd:string" use="required" />

    </xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

    <value>text/microsoft-resx</value>

</resheader>

```

<resheader name="version">

<value>2.0</value>

</resheader>

<resheader name="reader">

<value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

<value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

</root>

Oraculo\Screens\LoginForm.resx

?<?xml version="1.0" encoding="utf-8"?>

<root>

<!--

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```



```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format

that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />

<xsd:element name="root" msdata:IsDataSet="true">

  <xsd:complexType>

    <xsd:choice maxOccurs="unbounded">

      <xsd:element name="metadata">

        <xsd:complexType>

          <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" />

          </xsd:sequence>

          <xsd:attribute name="name" use="required" type="xsd:string" />

          <xsd:attribute name="type" type="xsd:string" />

          <xsd:attribute name="mimetype" type="xsd:string" />

          <xsd:attribute ref="xml:space" />

        </xsd:complexType>

      </xsd:element>

      <xsd:element name="assembly">

        <xsd:complexType>

          <xsd:attribute name="alias" type="xsd:string" />

          <xsd:attribute name="name" type="xsd:string" />

        </xsd:complexType>

      </xsd:element>

    </xsd:choice>

  </xsd:complexType>

</xsd:element>
```

```
</xsd:element>

<xsd:element name="data">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

      <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

  </xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>
```

```
</xsd:sequence>

<xsd:attribute name="name" type="xsd:string" use="required" />

</xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

  <value>text/microsoft-resx</value>

</resheader>

<resheader name="version">

  <value>2.0</value>

</resheader>

<resheader name="reader">

  <value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

  <value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
```

```
</resheader>
```

```
</root>
```

Oraculo\Screens\MainForm.resx

```
?<?xml version="1.0" encoding="utf-8"?>
```

```
<root>
```

```
<!--
```

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"  
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET  
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a

type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
```

```
<xsd:element name="root" msdata:IsDataSet="true">
```

```
<xsd:complexType>
```

```
<xsd:choice maxOccurs="unbounded">
```

```
<xsd:element name="metadata">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element name="value" type="xsd:string" minOccurs="0" />
```

```
</xsd:sequence>
```

```
<xsd:attribute name="name" use="required" type="xsd:string" />
```

```

    <xsd:attribute name="type" type="xsd:string" />

    <xsd:attribute name="mimetype" type="xsd:string" />

    <xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="assembly">

    <xsd:complexType>

        <xsd:attribute name="alias" type="xsd:string" />

        <xsd:attribute name="name" type="xsd:string" />

    </xsd:complexType>

</xsd:element>

<xsd:element name="data">

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

            <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

        </xsd:sequence>

        <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

```

```

<xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

<xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

<xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" />

  </xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

  <value>text/microsoft-resx</value>

```

</resheader>

<resheader name="version">

<value>2.0</value>

</resheader>

<resheader name="reader">

<value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

<value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<metadata name="toolTip1.TrayLocation" type="System.Drawing.Point, System.Drawing,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">

<value>17, 17</value>

</metadata>

<assembly alias="System.Drawing" name="System.Drawing, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />

<data name="VideoPictureBox.Image" type="System.Drawing.Bitmap, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">

<value>

iVBORw0KGgoAAAANSUhEUgAAAn8AAAFnCAIAAABYQbjHAAAABGdBTUEAAL
GPC/xhBQAAAAAlwSFlzAAAO

wwAADsMBx2+oZAAA/7JJREFUeF7snQdAG1e+9ckmb/v7tmY3u9lUx4m7494r7h0X3Dsu
YNN777333nsH

0REgBBJCoEpvEiA6dpwep+c7MyMwxrgkm33rOHNyMr5z55bRHc39zX80QionJnX8uPqx
Y8eOHEFKmXO/

1NXVsTyOYqQmMyFkUquUkIOWkMASorZiSYksQqxSCUpTN03VZPuUILm0aNGiRYv
WT1kqJ0+cOHny5Inj

6kePHD529Ah85LAaVpFFbCIIocCDIrfMvAkCKZUpsuTUVQqiypUpQj5QDdZSq2TzD22
fFi1atGjR+ulK

5dSpk6dOnjiidnDn9m27dmzftXP7zh3b1Q4dBP6ITfj/JLFcsWLF2rVrqTpU5tGjR/fv3480tUo
lKKmq

qmI5NXPPnj3Lly9fv3490mAqlTINyD948CDaBIaxSoW8VFMP07R+adGiRYsWrZ+EVM6
cOXXq+LED+/bs

UN26XXULvHf3zn1795w+dfLMmdPw6VOnXnvtTNnzoCLADASIHbv3n2alHJ9Qmj0nXf
eQT7VAZW5ZMkS

KvylVrH17bffptKALlZRBel9+/ZRMag4Z84cqvhJRh7Ug73TokWLFi1a/8cicUfoyamkcu7s6
dMn1Pfs

2n7y+NHdO7fv37v71InjB/fvO3vm9LlzZ2EwbtWqVedILVy4EACeN2/ekSNH1q1bt3LISiT
A0a1bt6IM

omHkgKaLFy/eu3cvCr/11ltnz6KRc6i1dOnSXbt2LVu2DBHwjh07XnzxxcOHD2/cuPHNN9
8EdP/6178e

OnRo7ty5aBZN7dy5829/+5uamhoaR7OzZs3atm0bAmgUmD17Nvab2h+qcVq0aNGiReu/Im
AIONdgAMi1

fPny1atXA3aITpWbHy6VCxfOnTt94uC+PefPnDpz6sSpE+qnT504euTwyRMnLlw4D6Px+f
PnXbhwar2g

dXAUeN+++fTsSWAVosQkCd8F8kBVoxLRoEYJdZKIYCHr+/HnsEGpRu4g0SIIVV6xYQ
QXBqIJVLFEmSiM

NCqi2GQX8+fPxxKkP3jwIjPCGgUgciMtWrRo0aL1X9A0DIGsytQjpXL54vkL586cPXXi8
KEDh9UOHj2i

dvDAPkS858+fu3Tp4uXLly5eJIC3Zs0axJ3AJNgONAKQyAHhEekiMEUsCyKCjohosR9IJJ
BdsGABomT0

cfnyZaow4leEuXPmzEELL730EsogOH755Zexr1g9duwYuti/fz8qosA//vEPRL1gM7pAMYA
csTLiaWxF

g5cuXcKSFil1atGjReqp08SLQ+XipXLp44eKF81giyj1//uzJkyfAcSCXxNsIDQ2Kc98Ddf9RL
mrQokWL

Fi1aP32pXLt2HfLVq1fgK1gS/2L1adQ1WrRo0aJF66cvleua15UmdW3CT780adGiRYsWrZ+mVCZxO2lK

2EYtIa0HROU/WITJydYeqx9WHpqaqfxu/Iel7OYBKtTtokWLFi1aN26oAFyTpqRMKwt8P6ldNV110fHN

s95/OhX0yxPhzx2PgpHAKjKxCQWURf9Lo17jf07KbqboJiltbW0dWrRo0aJFi5QK4ED539FZTcOIF11+

fzJU5XjUY41iKIwqysrPusBdSJeUHi1atGjNJMwPmpqaV69evUw+NEs9N4tVZNJTx2OFOVaZ+ulIRUnh

HyqNmWYLL7pP4+sTGhVRXdnQMMy2cPPqkDGjRokXrfmFm0NLS0tDQyM/Pb21tHR4e/pAUElhFJjahAD2B

PEK4TEGoo1z5iUhF+e8P0o7rDr86GT6Nqd/LqI5GIM39IJFQe6qFnTQ0NDQmZUKLfi1aU4RpAdFtWlra

rVu3KOg+KGxCARSj55CHCVcnADACSuX6T0EqRj9U7172nIbSH2w0pWz0+wtge8qFncQ5Y0rKjBYtWrQm

hDkBTBWLxUrMPIIohsL0NDKjEPhifDA4iHaUWU+9VJQU/p6afdFvGkH/TaNBZdPfU0RE+XQLO4n3hDkp

C1q0aNEihQIBW1tbIpEo6foEQmFUoWeSB6Wjo6NQKEZGRkA1BDzK3KdbKhSEv5fevug/jZ0/itGssoPv

I4DtKRe1n5aWIIY/jtDOj9UULVq0/msCJDlzMz96QNR3JZQrDwhVUFHQBK0J6erqDgw
MYHxGR0cBNsQ8

yg1PsVRAhe+lZVd9pIHZRzQaV3bzxKIuIp5mUftpY2Nj+yMIjVB+Qn3f8rRo0fq/ECYEAO
P27dsUU6eK

eIRtufKAUAUVf6T55FkQMAbW6unpUfSFAGBkmpubK0s8rVIhGTyDyF8Pmq61Z0yn8fJ
H9z4dN+UePFuy

I2X/w4W6sC3le6l7toOJDqiCxALFbLAkt5NZ/xFR/T25v6+mVX+sadH6CQhsKCKpoWgxT
eS3UfSUKzMp

JycHF/TKhn6uwnRqZmZmYGAQGhrK5/PlcvkHH3ygHCASwLhAecpHSYWiwoOifgJpqo
6fv/rLE2HTYDnN

z6lHPH8yWuVY+HMnon5xIlrlZCwSsMqxsOePh79wMkpFPeq5kzEoMK3ipH99MsLIylG5
E8+QnEg5/3Ch

LmmiEfJ/Z2fHe3ZydHbEEpmTBcktDuRSuf4f0GRn38tPrmkVn9C0aD3tAn07Ozs/nknU05r
KlZmEiqiu

bOhnKQSOiHfLysrGxsaUg/KAsAmEA4OVdZ4+qTg8IEdHR3DChZTrFK3SCpxGygf93Ino
X5yMBWJVjkeD

sr9QD3sOq8fB4yiQmEDviWiyTMy0ilO9UtNfuSv/YeHYKNIIBo3K3B8k4kqGbI1KKHOnS
DmI/4ZwMCjf

SxEHiPrHmTR5wJR5E1soP1w4ytRbYdrhfrQmauF94jzZ2RP7yTWt4hOaFq2nXaampuPj40
pQ3C/qaU3l

ykwCVxD2KRv6+cnKyioiIuIR3J0UymDuxbSsrPmUSYWcQ+8TQV0XlxX3a9HabdMYOa
OfOxYK4v7P8fDn

jgQ9fyxERT32uVNxvzga/MuTkSonYlROgMSIgyOmVpnRBrZuyr15AuFaAVcMFOQg6up
Bue3hQhkUtra2

trCwwPURaik3/CChOhp5RFMeD5e7uzt1MKiRp4RVNzc3bFIWQrEJk/8r024eHq6Euu658
d62qX6I0D72

jfoQAomp3T1C1A5j9PC2xkC6u2Mvpng2WD+5plV8QtOi9bQL+FQi4gERT2k+fCsIFFA29
DMTJlgmk/nh

hx8qB+JxGhoawlhhLIXWf5qkgt2aUZhhJ+diT0/PzbpB0wA5oy8Es7Ri+Trx/BtRHO2Yuhsx
defCeW9q

pz9/PPwXx0J/fS5BRR0Yjn7ugYrTvEk7gOTRYwRKAaJggKWIJfmONTM3NwdQAQYKY
I8QKuIaKo6nuHPn

zrA4y9fe/rFVHiZURLxrVN2FprqqjZB+sCnvwhwhji8KAH3Yb+4MIJbzJ8CpwCLy8vJRFp8jJ
1R0tYpsT

7OlP7+Xl5unl4e7p6eHh5enp5eXphYa9vWBUnqyPptAdRLWJQ60lVWucJAYAYnDG91RJ
R/U1LqQi0tg

UdPI+HBbbYwtgEd28uR+ck2r+ISmRetpF67UIXx4QMSzmhYWgYGBIIcy6wGhgLKhn5M
wtRYUFCiH4Ake

SMfExGCCVdZ/yqRCTakPSPmU8bsBpH5/MmQaIGf0yO0PvvrUU2/gb7757ttvviX15bffNQ
9/phVZ88IR

fyICPh7znPpjw8/nI2gwP8IARUYV1AqJLeuY2AcACE0PtTJL0zwMAdcJ68eHhQ2oYBpln
gYNboqoq2s

0JRy2/cXWgMp9apI+lbpUdRUbpuQ30zy9fXFUFvnSrEbM2hYXOTi4uPjoyN4n6+Hp5eW
Lf389Nz91PT

M9+tZXBI10jX3d3Ox8fZxw+bfH190Kqvn6e3n7eXHyoTtQj7+oBMeJmIcdEpse7jA4hSY9
BdHY5NU/q6

TyhMEtcZBoV9fLzd3ILKOsbv3JbXxwL9nijffwwTSt2z9j/aTmPNC1aT7twwX3r1q1PZIJ/f
39wcDB1

FV5YWIiZQLlhQqiIkEPZ0M9GmFQxLACqchQeJ5SMi4tDLWX9p08qSgo/ICV1SaldMZI
Gx4d59IPPgN6a

juHY2r746o7EquYcXrdo+Msvvv3u2+++s0xgI/Z97kiQyrFHfe5L+YatHxVmPUxAGi6F4mv
kBHjH+zsF

tdU1go5eEsMKQa6nJSJOIAFpSPwRjKRWqRxHx8jCpoHxYWmRux11Z1jZLhknUeUnq0x
mTq5OzYECHR31

lfTVRxqbkK1UIxDG2t/fP+B+IQfFLLIIQ3gFnXU11TW1NbWcGg7MreXy2IxxT0+UgVAd
CPQmuOpr6eu/

9oLuH7ae+N9dF36b3YZOGzJ3rr6kaeTn50m26e/n5evj5uXr7unr4evn7efv6+8P1no5Ozuxuol
95NgD

mV4EfWMqmfGB8RZAU5Ok+hFR5SQJlrz98f7wdGxQARMD4sKHB3w0FyxPxokEiv2
S1VFewnjDRykE8Z

6clNKIR15RDck7Lkg/af8LT8h5sWraddmHN6enqUoJhJIpElkwOiCwhpZS4pVER1ZUM/D
2HGwPXX8PCw

cgjuFy5HsOmjjz5Srn/yCdLx8fEYQGx9p1Iq5Hw7g/BqKSG9x+iJA194/L33v/rmW82oWpX
TicSTVsfj

f6Ue/KvLWRnslq+/+67vw2//cD72f04Tn/4+9ubzbqNgEgEzC6gA0sxTBP2gyUBjtpspLgYR
dFpZeUZV

d4HAw6J0RLcYfTAY+eA0hASFZCxxLM3MzExMTHCbiVXQiGoWCTc3N5REg1Sb2Ip2
qKaokBo0okriHKBy

kDBkkfRIGVI3O1AemegUHaEdlEH7gYGBuHybVFBQENqxyiXoOyxKQ1/YNzRLiSoPIU
F1Ye/kZODiMmeP

+l+2n/7tHo1f7b2qktmKTvlZB3+/7cji09ctAwOd/fy9PD1cXewdnG3tne3c3J0RuPr5+Xh6ultb
W5V3

onhHNfIBCDWG2DdcLiCNjnCssQPYf7wodIc0usZ7wN3dzcIip5Ggb2Oupbm7uytePfnhrwv
GLDAwAC8E

uEV5JycHJyfiKWtceCAnKCgQxrKDSFO0htE1xgdvPtRSDkQwEpNGlQAsSRM5QcFBgeR
ySnpDgz0pxJT

2qRF6+kVTnAWi6VvxUMehKAM5pDs7GxIFqmSkhKcsMqGfh7CdDp1uDAYuARBTh4
OMYHkxUmz0k2Y2ti

YuLTP0Qq1Pz+oCg4U1qk+aQ/pTDy3kdfffdjaga4vPdY2Eqx2N/cTxCRT16sWE66Hv3k4/e
uJGiciL2

hROPf/AKnSq7n0ngBOb+HCk4O96cb4I0xhqowEGysUmvV9y5c7u1wtYW+Va1PQBOJ6e
klfh0805PrZWD

g19CuaBzUHmzenywk5tBBKloEy2ADX5Z3Klb+QVEX1COZAT8acq3AzxQGCWVoaSD
Aw6/STWx0l1tgnyc

Wg4O8SWNXcNUM7fHhzs5eRFOIFloaGjYhJBGjm0eced5WJyFncchQeakQkJC0JFfWqW
4d+Lm9GBfbXbA

wl3nfx/A71NmUWoL3XFus7aZvkd0GqdDMVF8uFdUmWgH9BY0Y+enqKcWO+IU0IyCI
9JcvHwMoFNgepmk

d/g2WeD28EBrbV6AA/aqVkbmTGikKR/XIWxiXLvr3FyoN4tz0P11myrSgxyJCDjAz93DL
aZc2HVvSLt4

OUIHoHaRUDgQxEiEwEmFheNFBiPPJNPJB1MCg0KDAkMDQkCBTYyN7O1vkhxCfiYq
kwwD4a9evki0QdTBm

5NYw611OpSFsmkxQolZp0fo/Ft57mDTef/99ChiPEAK7qcWQRsWf21t3cqxAVolEghMc8x
KmnclxwKw7

MjJCFUhOTsa8SuU/zVLBVDejqGmL0ksXI6eh8WEeuP3xd19/fj2yRuVkynNHQ144GfU/x
8N+fTz4/110

+fqbb7/8/LPXb6T8gvhC8OPpi06V3c8kDK6zcz7xuNCwgGFvj4NBBT24bsBhyCBvkkryiE9
zTdkEFAkN

yro6uxqKbCLZ3QS05U117NoaTpOCWOnkxDoAvUCCVz75KexwZwO7tpbd0EmuSPO9E
A5mEh+RDkvzbAAq

HHh0VEmEu50ce3uw2YzsqJttBpK5uMRXdRKwGe7g17JruSI50cygqDjEDWMbERERSQo
J7LZ9fhPRtjgL

neMIEEzAWyo0LDwsHIXdQqrbiD1UNNfVsmrrK7vGiC6ZIX8663LCMfBSMdGppMj7jFv
AxmNGf959I66J

oOxwZ2NtLacG+Cf7LXJ1CUvKSEIN4RDFe+qTErLTtojFoDgyi6xFJzqpVq9TUriSJ0fioQsz
h1HLqpMTA

3BlozHN3j0rNMjYNFxmRUqQsKT4jMQg4J688OrnOjiCfp2dqNTmo/aKarIwsnpT4HH68h5
3qQUA9qqaH

HluuxpraHAarjXjQbaA2+BwOIi5dw8JDMRIYjYOH9ju7OCIdGhZMDg7yCR04uN+RyI/09
/PdrqpKFQ4J

DY6IjAgPD507dw7erQiXDQz0liLCqa0QxpAaaiSoHCqNNz0Sk6I20aL1fy+8/8vLyz/9nkIV
VFQ28fMQ

Ttu8vDy89tu3b2Pixct/8MwFjEHfjz/+ODU1FRRQ5j7deih9p+o3p5+UvsPv3/3uu290Y2p+cS
ruN+oB

KkdCiO/+Hou2SG34+tvvxj/+8h9XE587GqxyIn5axQeNTpXdzyQcAEdHDnEbVc6Nd0H4F
Ujlg15gGEXc

bjbx7BWVHu9kxlbhA8aurrn8vvHx/sZ8GxtsJe4tlxEfnfbyQsk70gk1COlud7IjiI+NIRsfVhviu
R62

n4NDBkXfXGtwmsL8JH2RNp+gL3qxKmwFb4ZFGaiFTegoqoag04AwjXrrRJHCACAOUw
gcFqXbox1XNw93

D08PT18f35DgELDZkUMEnvIa4pbuFQvbP5xMrOXpvgbTHbr/3HDp7bwObG3MO/37/df/d
sL4/+1hNI7d

Gu9g+zk5u7I7ODg6JjSCdeNtZcQFCi4LIDvs6Ij9h6iuB6o9NDU13dzqiI3dlav+9S+UvHjRM
K+J2OWW

IvfTp0/t2RNYR17rFBFfhXI+cHBfVjPRVOFJdW9vT7viVnB7WJhiaWmxecsmV1fnNOEAs
M8J2nfzZkbz

KPHyCvyJ57N27dqlF1JJXJkMNOZ7emKvTpw8rnHlsour0/z5c/bs3engYOvj63n69IkTx49F
RoQ5OznM

mzdn196dtg62mtevhYeGYNx0dXXUj6vbOdidPHXy9ddfO3P2NK67rKwssQljq6Ojo66ubm
1tjeHFMT1x

4sS5c+fi8Y4CepGJBPhEJ0XI06L1fy+8/cCMjo4OCqtPIhRGIZ/b+xazBF64XC7HXIpTWJ17
vzDtDwwM

AL2YM5VZ/y3h4DzZ8VFB8PFYEX+s6gE6zujx9+58/c134eUth31Z+7xYh73KL4VxfPKF7
39y9/PP714J

qiC4qx5O+IG604xOld3PJAyxszOXoG9XJWZYcJfKx1US0GhJIPROd40F8GIRQ0HRFicHt
YAcvH2dJwTg

GaaLENSNSLKBHifYRoJ17cU4zCgMNOLA51Y2NHCKEtcyMeDEfva4voLPaKdKjL+A
8yQnuwUzRa2AC8K

YaoNdgYlyUC5sgkQAnJ8fTHusaRiYmKww44kAh9QT4OfPwDsUtZOgLy1OC3A8aKx1V
93XfjLfr1/qd5c

uMvotdXnljGIMRAWnP/zoSt/O673v/sunHX3tXJ3d3X3cCfpS30a3cO1xwvHzlC3ynnu7ngT
o2vqznNf

kQHJ5nIsk21eF5fgcmHz5s07duzw8fH29fV2cLBzcy8qB8eH6ovcnF966UUvb48KouGuqFf
+CRiXEBcb

srpQDTc3123bVBHQLr3iwqypZeV7LvCtAZgHGhP9/HwRf6qpHbKysioix0eS5fLrX/8qJibax9c7Ojry

4MF9uP6JjY22srbA4Hh5uC9dsigmhoiJ3b3ckbNI80bQV+PypZvaN7Hq5OyEjpYtW4KB9PHxUlXdgsyD

Bw/o6elhYKkPrd955x1kIv6eNWsWEtS0ha2TQiYtWv8t4R2IqaOzs1NJ10cKxVD4qX/TYvc
m/cM0tQXC

3t7eeO1Y4vxVFnlAONkxXWNOU67/tzR1xx8nFZLBJ9H3oO8Hn3zxHaLfr777+pPvvvny
++++xKrX3/6

ydffOWfyVE6n/upYMIFe4ntH0+tOMzp9hDDKylhNXpfs6QnuUvkIfYA66rPenlobTMHWy
rQ18lGMpGZM

FrdjQPmRrFKgLwpP3InNxrGkbn5gib5wXDGDkz/3ZThgFU0hh7r7Wkd+V8dG2ZEV6Dt
xV9YZ3aERFEbI

S3wufLu91tcX508Cqfj4eLTvUkggcLyLz+Vw63n1ML+e39jAKg8Lj46KCvTPZjYjjiQ01Nd
dWZykpaX1

2uary/fordxz4+0sInAXFl5589SNfx298o/9Z29G5ZWIE0bue3Ggrx32GSNAfljbVe/lRY2Vax
HRdX8p

8eVCvMykqvYB6oPbYYWkpqAo2dfb2wu7HxYWoq4eTtE36oS6voGuvz/12rtiZ73h4+NJN
ttUcVML5ffu

3RMfH/fb3/0GgxcY6K9bRVwfYAdCQoLj4uIQ0iL83Z1LBM5dLOfZs986f+EcNRpHjx5xd
LKLi4vGwOjq

al+6eP6tN99ISIg7evSwk6tTRHTErp07QoMCbayt5s+f5+XtiSoREWFLI76LBFb37duLxN///
neMKjrC

IB8/ftzc3Jwa7Xnz5iET73hsxSotWk+J8IbEtW9lZeUHH3zw2UOETSiAYk/9uxe7N83fV9Oq
EwZ3ES1M

TpszirqqFz5cTV1Xx6r71NYhYLwo/Wb09PR+DCPfeP807d98H1mxwfs5v5aSTevrf+9T77
86utv7nz1

3QHPCpXjMSqnEv7nWOBjn3lGp8ruZxKQ5uNTTH4SKSrx8MC4U5eEgAoOA4XJ5iLiWz
S2HBKKHAI/2BoY

GFvagg3jA81cZn56ekZ6fAn55JEkB8Gx8ikkSQ5AFR0djQaxpBiMnFwp2WyBE0CFTJwM
FIHq3NzQqR3V

Ua0N4mbyASwiH9zFjqEw+uUirL7dzvH3BwaSSSUIJaF99+IWqlNfPz+UR2GifKsRxxAm
OhoBe2w6ky1u

6+4nPtO9c2uwOt5xiZruu0f1/kXSV1SgufCc4YLj1zdcDKiRI2OkR1xdnpOdmZ2ZwOvFuqz
OKSw0LDgo

mHpUik9eAUx23Zt/A1cJyMhr8knIMHRPb+0ZUNxCyTsjbZWemtc2b97o7V3KGQR9+RH
HDuvqafsH+FIg

B339/al0M/OGJgLcPXt3JybGv/rqKxi5kJAg6mvQPVxbjCGwqK5+DIG1H494YqyL7YZ9M
DU1UVHB+zD6

2LEjX14eyckJ//znS4ihMTyv/OtljNLhw2quiInjYjdtWB8ZFpqclBgTG71h04ZLGpdwhbly5Q
qUAdp3

7dqZIJT429/+lhpevLmPHj3q6OhJjTPoi1FFAQu0aD1VwtsSZwfmk6qqqr6+vvfee4+CLhJYR
SY2ocBT

/+7F7s3oJ9e0ikpTM6GyyEOE6xJl6vGa3r4ye0YR25OmWJn9UE02SfmRUsFOP1b/vBw9D
Y0P89D7d7/9

+kvdqOrnTkT/z9HAX5xOUFGP+cfVRHbHre++/hKB1a/Ox71wNOxJgul/Xo7CNPowYdZG
uEbe8BxvK7an

7vUjE5dIHh75jYgVb7ezyYfi7LkkFLn2VBjq7V1PkLG7KsjVFVvxtrYhHzkekeYC1e6hDY
DXeEsRY18U

phpMymEw8jOi/f3zSPo2MRzRFE6GidiXuJF7ryOOraenJ7ljCnGmE0pSj6VrcQd56FRSEhi
YmJqaRS

UIJwyeBRQtK3iREaFoZ3Gc4xQuShTkhMyCgoKWRkRTk6XTl//uLF6xo+rBZwsZ9nft5w1
hWrv1Cf+2ad

f33H1dmbTx6JJh5EGxKkevr6AuS+gf7UxYesziUyIjI0JJS6Ow764sICuPUkux4WpgBOUdm
FhUWFWqeW

6+vr4+WsOGhCvrphTsA+PT2dwMDyfHB8qL7Q2f7lf/0jINCPvPPcnbvt4mJEflAtZjjo+np6
bFt21aQ

b+2Ra/EpyXlpwfN9ajAWg8KUsLBQ5B88eMDKyooen4GmPG8XF5fExIS5c+fExsVcuHD
O0ckew/LCC7/A

0szU5O9/exGJM2dOW9tZJ6Ukbtq4Piw42N/PFyPn5uG2ecumxKSE1157FauBgQG7d+9CY
uOmjTY2NhhY

asJ68803kcbbY+HChUgQg06L1lMpvD9xVmKuQJyAq3kICawi8yfy1sVOzugn17SK0/yjaFq
bk36YUnBg

7vn+kpOV721QrIDlqayHskVJ4UdquS7595mfwEO3Pvj2+9uRHNUtsapHIt8Tp38XaPj0Y
cDON98+90n

dz/7u1baL45FqJx8/FNXy3RjlByaSQAYcOuRjyXuyQ6Ji0PcAGO8WX18IILqiCdsATOQD8
GcUx3xSS6C

P8zF4I2fH59giqw22M0NjPH29o7iENHhiDQPc3RAAIPfT94fDnGIPvf1DK8iHijqa0j39w8g
I7aeWuU3

cV1dcxuIx3eJG7koqeyI64hm3crbiX2Q5gHR2AQSJNbJCAhJ83AupaenZ5JKz8hISEz0KWsl
6ZuPkIEx

0bFxcYTj4+ISE6JiY8j72YNNeV54LcCnu0cyEzmKatODp36vfu1PDIK+gswz72y9YOWS7V
/eRl1JePj5

+QQFe/IFkfH6HTnPLToqGgAmH+HqXg5jKHARg66xGWOFEP1QmgTbBgTJ2H8M3fXr1
00JuHmkjel/7tzZ

Y8fCye/7Csp8PP38vNWP80lvmnc3RDop65+5ECyGK900Zjg7u5qZGSISDRViHEcEyYc
d3QsacEAjkjK

onxwCPT09Y44MJqR08sqDQk5e/bspUsXwc60NMwwyefOnfHx9QoKDkSmj7cXAJyGd3F
K0vIL59093a2t

LBOwz95ep0+fuqZ5LSUtBWPo7OyI3cN1pq2tTWZmRnp6mra29vnz593d3THOeC1IlyeD1
OTI06JF68dW

xkP8JJpWZUY/TNOKPdqYAWb0tGKTnlYMRqZSU8tIKBdwemZGGplQFnuYVJQUfqTU
7RKmofFhHrr94dff

facVxQFxn1cPf/54hMqJmOeOR/36ROjXX3316Zff/OtawvPHw1WOPP6pq2N28Zh3HyEA
GOFdDr8P8/6d

8YEeKb+OL+3uJ9buDEiZMf7gHOJXVx4JRZ4raI3YKzQ0h0fcm70z0i1u4Apbesl7uQSB8q
n7yWHkjWls

FtXxeHWibnKltSwCF6E+aQIC1OMDrfV1PPSlrNqFUBJ1Jzsir16zaruJW7cjXUIeUZbsZriJl
RSG8czM

ysrKzoaRSEpO9iOROd4t4PP4DQ2NjQ2NxLJRIBTVlkSG+zHbiLrjvW184vNgAdnreGvRP
m3dfxw8+adw

AdZHOxpq6+tYjKjQMDZxrYBdbBfyGpp6Jv4Ap4znDtbGxsSWtWGvxvuk9SJ+RUFMDG
hNVG8pxguPiKgm

brtja0sj0ZW4Z+Q2ANpWEx6WmBgfEZlbT7z44a7GejGnOCwshAR5jyA0GNdCUVGFXD
nR8mBLYz2vvrGV

OCi35HWFUZGIsVPqyb9HNtljqefzG5vIsQDUo3AE8RbMJkSNxwzG2z0rG8NFGIusjIz7C
0zWpRJEETI9

g7BpojtatGg9kXDWUCcOsSSSxKn4oJCdgZMTzMIKJ52RhemNKHuvNFJEA1NFZj6BiFJ
Uz9M8dY3cwQed

AWdmpGJ9KwsJNKysjGrpGdnYzKZakwO0zytwKTJnu8TtQNpWRmp5FoqkSBf/b39I8tN
lQoVEzxa9iGp

09D4ME+j73PHQsnfNYp84UQk8j/86rs3b6aoqEf+6vTjb2XbB6cgUnmE0tLSEJcDG2kVop4
hJWbujl/I

JKzChOD4eALeWPo3EtDobfRHGuQDgIPiy+snyDk+1MPLJT8a7WtliyGi7ejo6LgSQc8Q+
bEnEDLUIygh

bnRjE+LFNHZrL4E3xMfjI911LCKylAlCQ7FpsiNwDr2EhORVS2XKp59uj4/KGiozwrHPO
Tk5uX15IHNY

cxH+hrE6lHszXb1CXEEkJMSViToVE0VujciaqhkmQmZmX6ZmTZxJeVdo+SGodZqYif9
8nnSfuXOj/Q3

VbCIm8Kj7RWxySlpqWnhjAbyK8Ao3lQdGxvO6kDl0XYmKmK4wpMrGzoHCehCt2+Nyi
W8/JDU1BS8c9LS

UuMqW/qo1y7nYdhJGMtFUeHJyYk4HJGppezW/om6l/2t7LK0SNTC9WBCQnxWbbNseG
Jlh2Wi8mT0iPdf

nlKTQzKDcydMLX6wclGdFi1a30c4a+4TeU4S+cT0lZedm5uTR3ji5Jxm/K+sQhVD+ameqS
4llicnLyc7

N5tyDrq6t3WqJivCOQ+a2Me8rJycDDgbxM0BelOxhHNy03JzM3LzKGeiGOW8XNJ5mJ2
wzCSNxKSRDxON

5+RmZeekZ2anZmVjkkzOy03Ny8sgl2l5uTCxy1P3b5pUKAg/Vn+58PinlOEZYt9jUcQTzkc
jPv3y24++

+u6tG4kqx5NeUA+dVnGa0R15afQY4YoM8z51FxpMILOiIgwA8gwUUDCIBbgUYskU
YO8rEVOSgPgSJI

ozpqAY3YimIAA9UghARWkUltQjuognx0hE0QqqNBcH1qR9gx5KDNycK4UEDv+fn5Ba
QYE87Lz8/MyEyI

S4iJp6DCo+ODoqJ8Y6JMXJ3swoljM3Kck3PcgmNtPYmHtYKiI3liY+/YmUeEBudxGBE
MRhGQTEG3sGe

8fG+iYnRWVmBqRlwUGKiX1xcSFxcaEKcfUiIjpNjCK5UiLMgD9eC2EPq0oQaDewYXjh
2GGcWcpKtk0B8

7AoWCGrBTrxtGAzseB4Kk3VjUQb5KSmoCoLi/ZfFYDDwJiHrRkdG4RBgTJKxCRVhnG
W4WMIVCdqE0Qia

ws6gFjUaU8ZjBjMmTKxi8UM1pTtatGj9mMKZmV+A2W268xgwNt2z8qwm/YQiG8d8kfu
g8xmYnZTOz89+

0ARQ8zMp5zMIM+CCrIKC7CnOKSjIfcDIxCaUxDyWWUA4Y8Lk6r1G0gsKUGsK0khnF
BRga3JBflpBfi6x

6w9/vSokgx8vNbvkaYCC0dPoi5AXCeJ39U8mKMY/+Pa775wZbYh9f/G47/uiO+pK60mEa
BLYAD8gJIAB

Ir6c2ITVGTmny1OaVgA5kwUm8yGqtclNIFAAmtbRZA5VGGNdVFRUPEVYodYLCwqp
908eKuTnx+Xn7Dp/

zsTH97K1/SkTa5uoRF2vQG033+P6ZsEMxg0fv80XL6YUFcUVFB28pmvmF6Ht6KHv5Gb
i5ZNUWKjl7nvG

wtY5KiqewYhhMM6bWWpY2Rj5+O7UuBidn0v0WFiEPaE4hP2BkKCeR0Pa29sbGwFaF
MAS6cJCFCuE7exs

wFRyK96U2ITLCJTKc3J2ZBTko2HI0dHe0tIcuahNVZx4ldhYSJYHjJVdky99qpQlZ3QRa
WUpWrRo/Qd0

6AGpETp8+PCRI0eOHjt27Ogx9WPqx4+qH1c/cRI+fpLwyVOnTp4+ffrs6TPnzpw9f1bpC2f
PXTh3/uIF

+MKli5cuX7585YrG1StXrl29eu3apK9dv37t+rVrWoSva12HNW9owje0tW7q3NDWvamjp6
1vqDthHX0j

3Wk2MNI1nLCB4c0HrW9wQ99AS09fS1f/OulrsI7ulZvaGlo3L2lqXbgOa1K+SFIT8+INrUs
3b17RvnlF

T/eavt41AwNNIyMtY8KasKmxFqyvi9ekfub03uPHtx0/vvUErK56VG3T0cObD+5b5+FmW
VxcgPnq3iym

HOZ7UiHnyscrJDFrGiBn9O/Px/3vhfhfnp4hUP7d2Vhs+u35+Md+1wgOS8pBtPUsqfQBIZB
Wpopll5Rg

Ye3vaR8WIF1amllaetrOwy6zxDmj1CE5T83EPLy0dMvVG1o+Ac6pKWYR4aZhMQmlpckl
pa5hUXahkaml

pTe8gi/Yu7smJKWVlsYUlp43sMwl23GKjbHw9ymkupvSMYXEsNra2kgcP36czCghd4TYn
dy8HCQQJCcm

Jvj5+YDduFpADhllJcUlc7du3Uymi0DWm9o3nF2cLiKxm0TtmfxoTSt8zyWEJ4rQokXrPyB
g9tSpU2fP

nsXy9Okzpm6BpefOnT9//sLFixcvXLx08eLlCxcvX9LQgC9rXL2sAaZehYFV+JrmtWuaV5
UcvakJa928

Cd/U0dbW0dHR09XR19PV15+0vgFpQ8KGxoaGxgaGJoaGJgZGpoakkaCsf7/1jEx1Tcz0Y
VPSZuYGZELP

xAz506xjYqptYnrT2ISwickNExOt+2x6w9hEy8hI09CQRKzxDVOTm2amOhbmuhbmOpY
WupYWOtaWejZW

+rbW+va2Bk72hs6Oxk72BtYWmvo6ZzSvHb6isU/j8p5L53dcOLf9/JltF87tvHB+t7+vHaZJz
Ff3ZjHl

MN+TSuGTCQDeY/1E4e+/7z3WKcpenyEx71f5FBP/11EpQoHJsVZ+XgwmM5vJPGxmf9D
CWcc3Zq+WiQ+j

UC8kxDIqIZ7J3K113TEISS8gLIInJTGcyPWOTzALDU5hMLc+giw6e3hn5qUxmIpOprmWc
xWTmMpmWQUGO

YSElyh6U3U7V0aNH06KifHx8Tp0+efbsmeTkpN/85lfA7d59uz093dPSU/T0dLW1b4aGhh
w4sO/Awf2I

d6lGDAz0yspKysvLvLw8AwL8wGILSwu85R7s4t/RDHtMixatH1WIboFeUJbC7fmJuPXSJ
Y3LlzU0NK4S

vnoNBmYJ0l7XuqappalFWOvmDa2bWje0b8A3dYDbm9p62tp6Orp6urp6enoGBvqGBgZG
4KuRkbHxpI1N

SZsRNjM3NbMwMbMgluaWpqRNzK2MJ2w0YUNzKwPYwtoQtrQ2srI2srYxtrI2tLI2sJzB
+hZWurC5pY4F

YW3CFjdhc4sble3NwGBwVxNLM7MbFubaViCulZ6NNaxra6Nnb2fgaG/o5GDo4mjs5mLq
4Wrq7mzkYK1l

bHBWR+uo1rX9Wlf3Xru0Q+PitkvnVTUu7tC4tDs40IHJLMG0Rc1dM85gKsoY+HECqFN
zi3539km/+PuD

jS6SswuVvT5Dqpyiigc8qZLKCKZl5Q0rC01zq8NXtNyT08+YWSeXVyaXVuq6eh+8po10V
mWlRYCfZlLi

BQuHk8a2+27qBRyU79LUu+zgedrM1iI0dp+O6XELp1NWznYxKWqaOheMTK9YWOR
VVpYTPaA3Jump3VYa

4dpPUxPx65o1q42MDQHRs2dPl5YW6+rp+Pv7ubm5mpmZXr9+9fSZU2XlpT4+XuERYah
VUVGur69XUYHW

KqOiIl1dnRMS4q2sLJA/rX1atH7OqpgiZdYjpSz6ZIV/LCHkvXTp0pUrV4gbxZeJ0FbjytUrJ
GuvXcf0

QN0UJgzAEtYmrK2rq61LxLXa+rq6BnqwnpE+rG9sYECES8CtkbGpqYmpqam5mamFubm
FBWkkzC0sLSys

zC2sCVvZWMDWtpawjZ0VaSQsCNub29ib2dibTtjExsHY1sEEtnMwhR0czcgEcoxmsL2hr
b2BjZ0+bGun

B1vb6lhba1taAb2a5hZaZuaapmZA73Uz8xsWFjcsLW5aW+nY2urb2erb2+k72Os7Oxm5Oo
O7Jh5upp5u

Zl7u5l6ups622mZGF/RvqmtfP3jz2j5NjZ3XLM+/eIH16uUdVzV2h4U4kNM5MQ9OeppUqB
D4EaLgQbFa

1/+J7j//O9YPyFZ2/GyJNUVVD3Eli1VWVclksXLKmRnlFbksViGLIYngtYxZUMkqYrGym
EROMQpUMpOL

ihgsVmIpK4XFQsmESIZ8GSsL5VmsbBYrlcWKqyBWsQk5WGIP0P5EPzASSIVVWVIZ
S1btgyHePt2VR0d

bWD44MH9CMkNDQ2uXNFA1ItT7MKFc6GhwerHj504cbyouLCqCu8fJi580YKGxuWcn
KzTZ04fOnSQWcHE

JqplWrSeDeEcgSgcIqHMfTJRdXFylZcTF8AzVqfKoP3JBAPtq8oS/3mdOXMWEe6169cJ6
BKfy16notvr

WuQnsdo68E1tXVhbVw/W0dOHEdcSNjTUMzTQNzGEAV19I6DXCDYiQluEs+bmlhbm
VpaWVIZW1qRtYEsr

WyvYmrStgzVs52AD2zvakraxd7K2d7libWnvZGHvbE7aDHZwNqfs6Gzu5IKlmaOzqYOzy
f02JuxkaO9o

ACNha69rY6tjZaNtZQ30ahmbXDUxuWpqds3ISMPI6IqZ2XVLYxtWVjdtbHQAXUdHQyc
nQxdw183Ew93M

08PMY9Pcy8Pcx9PCx8Pc1UHP0uSioc4JXc1DOtf337iyS1Nj+/XL265r7LimsTMsxB7TOW
ZZHL9JT5MK

SdUZVF1dXVNTk5qaqqamduzYsc2bN69bty45OXmfbdo0Xv6I3m+XhvfcMykM5oNik8ZA
T5pVw8ayqqam

rKamsqamoqYmIj62sro6Ijq2mFmVXViKrYERUSiQkptdUs1OLyiKtk9PKyrKLi3LK2PmF
JdkFBQWVbET

snJT8hglZDvFNTXoHq2h7pQOkZhRxNbyctSjyzUbGJPqTRELZUiN9Gi9YwI73XMIWVI
ZYWFxEMPSCNH

ue0JhMKAbmNjo0QiAU2pqXWqqPYxSyDOoYjL5XKbmqw/L59/Ts6f+7CtWuIbG8S95N
v3CRCWx1YlzRw

qw8riUt8TmtkaGSCqNbY1MzI1NTY3MzY3NTE0tzIzETfxEhbH0Gwvq4hImAjE2DX1sbG
3s7W3t7Owd7e

8Z4dnAg7OhN2cnFwcrF3cnVwcsVyqu2cXG2dXG2c3GBr0laUnd2sXN0nbeniZjFhc9JmLm
6mq4mjk5G

IK61jba9o769vV5KShSDkYEIODo6QCbrKi7ONTC4LBY3cmqrzEyvWVvfRGE7ez1HJw
MnZyNXN1N3DzMP

TwsvLwsfH0tfb0t/H+sAXxt/bytPVyNbiyum+qcNbx4xuHFI9/pe7Wu7blzdeePqLs2ru8JD7YI
JI03M

jMSUOm2KJKVCXGjNjLxFlimjDx06hHcAlJaW5uND/K0ivDnWmaZMo+aPYjSL9+izKm
oYHxSHy62dYko1

XG4Vh8tSrhGqra2tqCTOSagamzjKsvinqqa6upaNA0ae1uxaDpokyOkZXm4aAR9s5HDic
whKIJ9EsUe

Ig67ppos8L1Mi9azKQ6Hg3PIBCMjIyIxWIEJzi/qHwI0yhxrpgMpHKgqZkQZoDOzs6hoSE
+n49VVMes

2kql0SZwOzw8XfdXhzR6QV/IAx2xdVJU41TXVEVoshdK1CpVEprWF3LI1zSDLI/WuH
FTB3A17yfr6egb

6BoY6hA20jUw1jcygQ2MYWNjEzNjU3NTcwszC0tzK2tzKytzaytzGytTK0tTS8vcvLzOri7
FgKK/v18q

IUZERVrZWts7Ozo4Ozk6OTk6O8JOLo7Ork4u7s7O7o7OboRdSLt5OLl7OmPp6q5Me3jD
Tu5eDm6e9m6e

dq4eNm6etu5edh7e9rC7l62bp7WrhxXs5mHl7mnl4WWNpZunhZunuZunmbObCVAqlYqk
UqG1rQ4AzK2r

7u7uiIryi44OlMm6S4rzjAwJ+tbWVJqaXLWyumFnr+vkbOjmYeqBSNfb0tvXys/fFg4IsPX3t
w0KsA8N

cgwNtPzfNHey1bQyOWeqf9xEBwDer6+1V09rr47W3hvX90ZFOGOqxryI4cZ7ZcYJVwUz
9owKDQ3V1NRE

CerwYwkYOzs7A8xIrzd90r+/8YRGg2j2GRbv4SL+phbpB4XxpxLVbDaXVwcqV3NqK9nV
tUjXcZHAZriq

hk0URiYHhFVWgbh1vFouYSRgnNgQuWXG3iY1uUffy7RoPYPCkQNoIeTt6uoCEXt7e4
FS5CAfpyeohpiV

+L5dURGCY6xiE+bPqZlYxQyAdH5+PuqiDFbRIHIQ7KIkMgsKCoRCIeiLwiWk8vLyqDg
bBZBADqiMqZg8

iYldwiry0QLZD9ERRK0igV6weyiGGXuyL1xDUHuufG3369rV63qIao2MdfUN9IyMCNCa
msKGMawkaml

kamFiTlhINbc0srS2sbKxtbazp6wg72Vg52ptWVRSXfX18lq8rX3z8yOkoilba1t4VGhDk4O
zs6O9k5

2NnY2djYWds72rm4Aa4uru5ODk7ETWY7R2tbRyukEe86OtuRt51tnN3sPbxcPLycnV1tHV
ys7J0t7JzM

HZwtXDxsPH0cvXydPLzRDmqZwY7OZu6e1p7etsAwcRfaydjBycjB2dDC6qZEKpRIhBa
WNxwcDWztdK2t

dbCMjQ2Wy7sx0sbGVyn6GhhcMjG5aml108HJwNPTwgcxboAdHBjoQDjIISjIISTYMSzU
OSzYMcDH0sXh

ho3ZBXPDE6b6R410Dhrc3Gdwc7/ezf3aWvtjotx4vBrMixhrapZ+cNBVcDBm1OzZs4OCgn
CcGhoacLmE

ox4SEjJnzhy8OZCGDjn+aJ8BoymqzWdYGMYNfy6QISTqISpFqr6Bz2vg1zc0cOuxwCphK
sHjEz9OWFdf

DwYjQV71csjaaIRogjQhssajhTI/wLRoPYPCKYNTCTPhwMAAgtHBwUHk4IzGyUWdZe
DxGClQB8XAY2mZ

oCAICnijLgqAtQAt4uDx8fHR0dH29nZsRXmsUmpubkYBkF4kEgGc2CSTyVAXamxsxKy
NrsFU5Pf09GCv

UIUq3Nraih6xSnWKncTOSCSSyb7a2tpQEfsW4zygpXXD0NDYxNSMuKVsamZsbmFsYW
lsYwViYwVqaW1m

ZWNuZUs8HEU+FgXi2jk42jmCnS72zi72ri52ri7BEWEdXZ2YiyztbG2dHG0c7JNSU7Gfz
MoKxL7RsTHY

GewwJJGIo2MiXT2dUtKSunu6MUv19/fhBXZ2dWZkpre0NA8ODSoU/bh+8PJ1c/N0bG5p
EktEXV0dg0MD

vX3y0rJCDy97bz/nmNgQkbhxYKAf+Z2d7alpcUBvYVF2T0+XSNSA/EZBHcZvaGgARiIp
KYJXX9vZ2RYR

4Z0QHyaqX95SVMszNNCUSAY4z4mNU6euTIZUx3NzNwN2gYEc4OIRwSKhTaKhTeJhr
ZIRbVJhrsL+1u5OO

neUIC6NTZobqxn timerjoHjLQPGmgf1L1xMD7GE/M05kUMNDVRPzjiKjhIM0ogEOjo6OA
w29jYWFhY4Kjr

6+ujAvWGo2QWUfi7s0/0N7AeZIRHI8rmnmlhPL+XMNSTy0YkqMwJ8xuJDcinrMwkSjcS
fyiarAXhqE0V

lfkEmmz4e5kWrWdQOJsQRwJsYAbCTYANvATVQEGEluCiQqHAKtIoANig7pwxs6OjA
2lMBZhUgRmslhgF

ZQFF4JbBYIjFYsS+6AvEpeiLJWJWVAdTUQbIRKc4kZFAFI8lMfuAefd3d0ogxz0iDLIx0
6iDHUHu6Wl

Be2jItLgH4KrGWcDbW0dExMzc+DWzNzUwgq4NbOGbc2s7SxsHCztHK2Ih6DAVEDXx
cHZ1dnF3cXVA5Ep

YU/C6VIZMrk8v6AQPHbz8XX39g4MCS4tL8vNz/cPDGxpbceWxifExyfEdXZ2tra2BAX7J
6ck9vb1Aq7x

ibHpGakgcb+iv5ZTExoRXM1m9fX15jFyXNztUUDeK2MU5ERghwiEDX198qzcNA8fB2
mTqLOrPT4hIijY

UyoVt7Y2BwS5MwqzFIq+1tam7JzkyEgfe3v9JqkItrHRc3IyrifpGxnhm5gYjguY8vICS0tt0L
e/T15c

nOvn68ivrww8k5PDEe+GhDiHIItINc4HDw10iI9yjoz3iY73jY7wiQhx93I0cba9Zm52zMDlp
YnAEADbR

O2ykc1hf+3BivI9AwMO8iIGm5ucHR1wFh2pGYZuZmRneKCYmJjdv3sTlkqmpKQ4bEIN
VWFFz2DI7KICf

3KiI6sqGnnXhyvQHSygSCWbypCZyhISFkk4Uzt6pUuY+Xij5A0yL1jMo4A3zXn9/P3gGh
kmlUrATMyRO

aqAOMAb58vLyCgsLcYqBcw/LBG8o+oKdoCMmWBSgPu6FQEoskY+WQVP0S9EXhAZ
TETcDyegRtZCJXUII

i16AcOSjLkVWLNEgVQVbkY++MlejDHUfG9cE2A3ExGiHenVTpa9nYGFhZW1jhwXxp
R87Byt7grgEdB1c

bJ3c7Jzc7ZzcHJzdnABddy93D293Lx9Pb38PH38PX38PP/9cRoG8tzeXwXD19PYJDPYJDP
INDPTy8/Xw

8SooKpT3yvPy87x9vVEpLz+3t1deWMRISUuWyXoKihhePm7uXs4Igru7u+Lio739PNIyUv
r6+1jVIYh9

W1qbGxrrPX2c/AJc09ITQGIOlx0QjDLxMbGhjs4WdvYm5eV44bLk1BjEvghqs7KTfP3tAo
McPDwsWlrE

sJubqYe7RX09B/SNivJLTopE+QpmkZ2tXlOTqLGhzt5Gx8lePy8vFdWLCrOCAh0iwt2ISD
fSPTrKIyba

Mz7OOyHBLznRPyXBPzbSPcDH3NXphp3lBWuz0+bG6maGR80Mj5kaHDPSO5aa5C8S8
TEvYmak5ucHp0gV

HMih6cyZM/Hx8fb29nZ2dmlpaZcuXcKxnFHZZbVnPHL/pvH43w2Ef38q9JBtUmYJW1n5
WRc1mLjkhHDq

/gChmngmT2oiR0L4/l6ofikpsx4vlPwBpkXrGRTohQhyaGgI4ATDwFHQF2cTl8sFDhGSIuj
EVgASJzuA

h+D1wUwAlaIvqqCdnP6e8fFxEB2xINCOKiAiII04GHMF0ugRaSwRNIOjaAeZ4C61S0igy
mQ+GpwsjzRi

bmyC0CmKoTygfmtC2A30iJJkS/fJ2NjUxtbe3tHZxt7BzsHZ3tnV3sWNtLu9i6cT2OjhDei6e
AK6vp4+

fj6+gb5+4Gsw4aAQOL+wCNhnFBZ5+gUEhkUEhoUHhYUGhAT5BPiz2Gy86pTUINDw0
NCI0LSMNLlcxmJX

pWemgL4lZUXBYf4BwT6NggBQnyUtKSQ8KDs3q7+/j13D8vJ1bWtrQcgbHOYdERMQH
ObT1t4qEPL9A92S

U6KIUtHg4MDw8BCMmDg9I764JFcm687KTgwHOKM9A/ztW1uksK+vjZ+PXUMDt6urP
TY2KC01BuWrKkuc

nYxbmiUIeT3dzXw8LQoLM/v6ZCUlOeFhrrExXnGIIdON94IR435TkQDgtJSg9OSgxxjPY3
8rTRcfe+pKt

xVIL0xPmRscsjNTNjNSNDdQzUgKIUgHmRcyM1Pz84BSpgougGYXD4+PjExgYiGOJiyY
ksIprq4cJbzK8

TWMYtdcDCjdZZr6umfj7c7G/OBEFI/GGVtJmq8xr/oyQTCbVOJbKms+6yOEUUF4OC1/
gFBt0tIpnhTS

xKamJsrKXFJUv5SUWbRo0XpigWeAnJJdE5LL5WAYJj3gDRgGRBFWAqhgG3JA6GmZ
k5/7ogo2gYuYGZAP

QoOamIhRBeWRxlyBlkF3iqaoCEgjkga2kUntEvKpcBz52D1MvGiBagehMwVmCHEwcnA
d0NbWhh4h4B+7

gR1AC8pJYYrMzCzsHZycgFsnZwcXdyc3grikvZ3dvV29/NwR5nr7e/oEePsG+PgH+QeEB
ASGBoaEB4ZG

BIVFBoZF5hUUgr4l5eWlekMiY0Iio8MiLzOyMpPT0qpramQyWXpGRmRMVGRMZFOF
vaQXVOdmZ0ODJdX

lIZHhYaGBwmEjT093RlZqVExYXn5Of2K/praar8AT+BWKGoMi/SPiQ+JSwzv6GwTihoi
Y4Ja25p7ejqz

spM9PG2rqjBWvZlZiaWl+XJ5d15eSjTYGecdHOzU2iqFAwIcAgMcGhvrurs74uND0tPjQF/
wzc3VrLVF

0sDn+HhZBvjAlBRn9/fJy8vyEO8mJvgmJfoRTvJPTgoAdwn0poZkpAQnx/uEBtl6uek52Fy2t
TgH+loY

q1saY3nCxBP4ZlpQc7MI8/LkpP3g/KuCy6gZhcOjTD2BKMzggq1pNErVRWRNbnqRFDA
WGBde2OMH+Q/pR

m0ZjP8C0aD2DAg7BSJy/CEMRigCHoCBykI/zGjDDsqqqqri4uL29HfnY2tHRMS0TwJukL
5pCAWASMTHK

gIgICkFfMBIExUQKslI0RVMoQ1EW5QFIapewFQ1OUhkTLzahPJZoH9Un6Ys20SniY6A
XVEYaVVAXuweh

HRHmapFILJFgaWxqbu/g7OzqgajXORXo9Xb29HHx9HX19PPwDvD0DfTyC/LxC/b1DyG
4GxQWEhIREhIZ

Gh4VFhEdHhUbHhOXkp4pk8mFfrEn4tjI6KDwSEZhoWJgoJbLLS0vxy4V15aGRUWERoY
V15b09feVMcty

8rLlvfKKKmZ0fFRETlhILOyR9WTnZsQmRDEK8xTEZ8DsgCDvdpK+IRG+kTGBYK1c
3sOr56Smx/bIutk1

lf4Brv4BLmx2JeibnZ1cXs5AgQJGGqLVxET/8HD3trbmtrYmP3/7wCAngYAH+iYmhmdlJ
qA8u7rc08MS

g9rYwPXzswkKtC8tyenvl1cwGbHRXvFxvgnxoG9AUqI/GfgGpCYHqpeEZqaHbib5RYY6e
HsYONtfs7M6

b2V6ytLkBJawmfHJnIyQ11YJdbyg5pmmSBWQktZ/WjiLMNY4E3Aq0qJF66elEfKJYuAW
uKKup4eHh0FQ

oA48Az7BMxAOOESwi5IPZiIN7AHDgCumAsSjCHxBVrSJxql4lLrzjOkCzVL0BU1BZVR
H1AjEoiK1PyiJ

eJfKR2E0QtXFEu2Dx9gEUfE0gm/wHplg/O3bt9EX8Iy5CAm02YH96Oqqb2jAMiMzy97Rx
c3Th/gmkIeP

q7efm0+Au0+gp2+Qj3+wb2CYf1B4QGB4YFB4MKAbFh0eHhMZERSVHRcVGx8TlxSTk
IRgt5bDHRgY5PLq

45OSs/Py2zs6enpkmTk58UIJ6AvOzMnOzs3p6u7u7OpMTE1kFDF6+3qr2FUJKfGxiTFiiRi
hcD4jOzEl

rqikQDGg4NbVBof7tXe09Sv6yiuKE5KjxBJhf39fUXFuemZ8V3cH4uDs3NRyZiECWYWi
LzcvtaKisLdX

VlyUmZwcmJYWvIY3dvd0ZmTGxcUHCot1CJdTU6Nyc5NB35oapq+PTVtrE6gcFO
gQGuLELMtX9PeC

vnHR3lxuRQUzPzMzqoHPrqpi5ObECxo55aXZGakh6SmB0eFOfl5Gbk5ajgh/Lc9Zm52yM
T9jZXba3ORU

blZYe3sTdbwgHLkHZ38VHOvJYCE1vcSTiG8+yn04q1Oixatn5DAUdAUTAXJQFycyCA
cwjkhUjxGgVu3

bgFsyMH5DqRNywrQEbnK5XIwG5tQAI0AutgKNCIfOKeu0VELmb29vRShUQwV0QK6
Q4HJCQQJVKHqyqRAW

O4byWKIR7B42QSA02AzqoxfsDJpFJpYgPToCran9HBoeRt+4MsBEFRAY4u0XSNxe9gv
0CgixCgz1CQr1

CwoPDikMDosODY8GdMPCYyMi46OjE2NjE+PikhISUxJA0ZS0xLSMhOS0qPjEWm5dX
38/XiBabuvoKCgu

TkxOSUxJRRwM5GMQqGuXguKCpPSk8soy4iHnutq07LTUjOTmlibAuKSsMDMnlVIZhs
uYhsb6qNiQDrzo

jrbePvnI6PDAQD+HW52UFpmSEV1Snifv7UFmv6JXIhUqBvrKmQXsmnIEr0xmbmZWRE
5uVGpqaBWrCGAe

GhqoYhVjhOTy7pychMLCzIGBP15ddVioa0dHi0TcEBXpER3tXIVZOKDoq6ooLGSk9PXJ
2tuaiorS0GCT

VFBVVTg4qJCI6rPSwrIzQuKj3YL8zL09dF2drttbX7C1OGtnec6G+Az4dH5OeGdnC3W8I
Bw54kLnfqkA

DN9LYAmt7yXqhjPebRhunAk/lnDxONW0aNH6Tgw8A/kAQur8BT+QRg6EUxtTIhgMzkF
IYBVbp2ViEsAM

gHZASIRBAYrBICsKgKPU/IAiCiMfxdA48qmKSKM8iIuuqV2a3AcqH3XRI8pjidXJ3UMC
QmvoheoLMMYq

ilE7g/aJO88Af1MTltbW9p5efn5BoT4BiHRDfYMjYP/QtyKCwqOCwmLDI+ljo+MjohKiYp
OjY5Lj4IISE

1MTEtOSUjOTUzNT07NTM7NSs3MT0rLjktMi4hMi4xKj4hLiklNSMzMzcPDgtKzsxNTU
+KTE+OTE5PTU7

PzuvMC87Pys9Jz2bkZ1fnJdXlJuVI5GenZJXmMMozs1hZKZIJadnJyWmxiDGITSJImOD4I
MiwN2MnPj8

orTcgpSM3PjEtIj45FA4ISksKSUsIzMqIzMyPT0sJyeKwYiF8/JjMjLCk5KCEhIDk1NCEhM
DYCA5JSU0

MSEwKTEoJTKoPtY3LtaX/HDXPyHeJy7aKznBLwUI432TEv3TU4NTU4LS00Iy08PSUkK
zMyLzc2PysyOS

4j2DAyx8PPU9XG8422s4WJ+HEQSDvnmZ4d3dbdTxgjpX0JTJe1LB0f1ewruH1vcSToDJM
0f2cPX09Ewu

HxTyKSnXJ9Td0wNPk3Lbw0WVeZKStGj9zEVNIFNPFqQnM7HEqQ3+QdQ5PmPmVFE
FwHJsxRIFkDNZa/Ia

fVpFFFB2T4oqT+VTaar8ZD40uWmyr8nGiao8uscuIhdLJxdPH9+ggODwgJCIgNBo/5CowP
Do4IjY0Mi4

8OjEqNiUmLhUODYuNT4hPSk5KyWfcGpabkYWg3A2IzOvAM7IY6Tn5sMZuflZeYwCR
kFeQXFeQVF+UXFe

YWFeYUF+UWFBSXFRWUlxOVxczCwuqSBcWlFcXFZYVFpQWl5UWl4IF5agam56dnJ
3T2dzyQ9Oz6/MK2g

OL24JLO0PLusIre8KpdRnJqdl5CTl5CXn5ifn1RUmFzASMjPjy0siEfUDSNRwIjPz4vNIxy
XkxuTnROd

nR2TIRWTmRmdlQVHZWaEZ2aEZWaEZmYAsYRzM8MJZ0XkZkcx8mIYebGF+XFF+f
FF+YnFjMTSoqTigri0

ZP+wYFtLwMvDx0Xp2sOthedbC/bWV2wMjubnxMhk3UojxbeRZiqZdPnWxXq+ujJpYQ
wrScWdXGKNzrO

BPnDRR0PIKESIJ4knxK1Sols71Giin3fWrRo0XrGhNMesxJIDy9/P//QgOBIRLpBYTGBoT
EhkXGhUfFh

0YkRMcmxielxSZlwfGJmckp2cmoOnJqen5lVkJVdmJldkJVbkMUozmYU5xQU5xQW5xW
X5heXMkrKGCBq

GbOwrLyorLyYySyrrCirxJJ0FbOMVc6svueK6vLyytKyipJSZnFZBVxUwmTkMtJ7ZN1t7S
35hekIlgG55

DrMir7q6sKqK+Bn0qsrc0pL0nMyYrNSI3PSo3PTIrNTQ3MyIirIUdlVWaWFCESO+BLAs
TCwqTCwsTGIU

JObmx+XkxWbnxORMODcnOjcnKi83KjcnMo80Iy8a0C0AxRlxJYVJZcXJZSVwSnlpGrM
svbIsHatZ6cGR

4Y6+3kaeHjquTtcdbS8hAra3umhlegaRsVzeqRxluxH3iMjxvi+aVaFuDz7PgJLaH0v4XqTur
wF4Xof

JxwSZYpUX18flZiaT6WJozfBUSpnsV3JWhDSU4WS1BKkK5KWeF7imqWFi1aP0XJe/tk
E/b1C/IPCAsK

iQqNTAiJTCChm0gukyIR7yZlJKZkwynpeZQzMkHc4uy8EjgnvySXUZpfVJZXVJpfWo7ot
RDkBEgrWaVV

hMtY1eWsaia7urJmwtWsSnZVJbuyqmaK2RVV1RUsdmUFq6yCVQozq4pLmYz84kxGSV
ZZZV4li1FdXcSq

YISU5RblJ6fEBfm7WzuY3bQ11rQ1uGpncMXOQMNG75KLhWZssBOTEV9dmlZVmlpW
TOCzpDipqCSpsCSp

oDgxnXGflx+Xr3QsYeCWgQSWgG5MUWFcUWF8cVFiaXFSeWkqsyytojwDrmJmsiqyqiu
zq8rTc7PCYqNc

ggJMvb303F00gV5Xx6tONpdszM8xciP7+rqImRjTbZ9c1ieTw733TbMqAMP3EnX7gtaTC4
OGwJek5H30

Vb77J8iHZX9/P8pgSWUiQZVEGiXBb6SRmMycFJWPAIQLVAGidbJ99E49AwlR5Smhp
DJFiiqPikhTLTxa

VHn0CClbp0WL1k9KoC/pfnmvgh8AyPCwuPD05MiYpMj4Ii4rAkEtHxafGId9NyAd20T
AackVOUnVuS

k1eayyjPKyScX1ReUFLBKK0oLK8sYlaVALqs6jIWm1ldU8HmVNZwqsifXGPVcVncWha
nhjSbxamu5hJm

cVhwNVzLYnORqKrmVFZzKqo5zKraUia7qLKmuJpbyuGV8XjlrMq81ITQMH8XNxsDF7
ObLsZaDroa9jcu

2mudt9c6Z6t52lLzpNnVY45GlxNDnEtzo1klKZUIKcySIJKS5JLS5MKSxKKSxMKiBOK+d
FFCEenCwrjC

onhGQSyRKIwrLsbWhJKSpFJUZZVGRUVWZVWWSyKrOrWblsVi6rMjM/NzIuxjU4
0NzXW9/DTcvF4Yqb

0zVnu8u2FucL86IVoC8xUWIqx/RKoHc6fTflfy9hKqf1vUSN2zTUQciBqMOAVZRB0NzR
0YFTgtqENDKp

wlQMjdaQppqQorZSwlbUonqktmIVtdAOhdXJHKo80lQCotJYUsUgKv8RQmGleG30u4IW
rZ+gEBbIeuWY

LHrkfT1yRWBQREhobGRUclRsanRCelRiGhyTnBmTnBWfmpucyUjNLkzPLkrPLkTIS0C
3gAkziisLiqsK

sCytKi5nFVewignussuqa5jsWpjkbh2rrq66jsfm8Wr59TW8OnYdd8KcWh5hDp/LrefAdfW1
XF4NlnX1

NVw+m8uv5jSw2PVVtfwqDr+ivrGqqio/IS4wNMApyNM2wMHUz8rAy/iGm46Gs+YF5+t
nHS6ftLt4zPbS

UevLRy01jlpCoeZpfj0vwV/Izq0pT2NXZbKrs8HUsrKU0tLkstJkpCkXwwRxE4tB6JLksvIU
uJyZxqxI

r6rKZrGy2dW57Oq8GnZ+LWIOTV5xYWx8rFtwkIW/r4GXhzbQ6+mi5epwxc7yAjb193Zg
2kXgC/T29PbI

MMz3S4WaQ59c1GxL6/tKOXyPE0XKgYEBlJFGDkAIHmNJ5UDKQzchIEF4TW2CSHQ
q2U11PTQ0RCUoTaXv

vy9lo7Ro0fqpCdfpuGDvlsm7ZX3dsv7gkOiIyMTouPSY+IyYxIyohLSYpMyE9NzETEZSV
mFqdlFaTnFG

bnFmbnFeERO4ZRQR3C0qZRWVvxNmsksra8qqagju1nIqONxKb11LbeSA+7Ws3n8mnp+
bQOf09hILBvq

OQ08ynWNE26oq28kzKMSAm69gMMTcOpFHL64DsvaOmZhUXpSYmh8tH9MsHuEl32
gnam/ma6nzlWXq+cc

L520P3fU9rSa1Y19lif2mp/ca3pqr8mZvTZX1V0NL8f72/Iq0uurs+vY2ezKjGoikE2vKE8tL0s
uK0su

LU8uKUuCkShjppYBupUZcGVIZiUrq7omt6Y2n1NbAHM5hZR5dUVlpYmJCe5hIZb+fobe
HjruLtc9XbXc

QF+rCyVFcf3yjl7Msr0Ihnq6ERzJpz94pUJN1k8uauKm9aMLJwOWAC11ex+rw8PD/f39g4
ODWGLkW1tb

kTkZxU4VcSJ1d6MYqlCopjTZLOpiSWWiGBqZsR0IxWjRovVzkIx82KpHjqCsv0emCAuP
i4INi0/MjoOT

s2NTshIz8IKyCwDdtNyS9PzSDEZZTkE5o7iioJRVWFZdXF5bwuSUVXLLKrHkILO4FWx
uRW1dJZdXVVfP

4vFhdn1DDb8Rrm0UcAQCrIBYJxLViYQ8kWDCjXxxY4PSDQ3CesJliPkNll6DqK5BxG0
U82B+I7e4ODcx

LrxJyJXUs5jZyYn+7oGWRt4GWu6alxwvnrI9fdj6+H7zw7vM1LabHFI1OaxqeFjV4Nh2w5
O7jM7uNb18

yMnwQkFKAL8yo7YspY6VxUUsW5nFqsysqECAM1ZWkVZeiWVqORIAMysTrqzOqmJn
13IZHG4B11sI19UV

UebzSyqYKakpPpHhtsGBpv7eet7uN73cbrg7XnWwvVxWHN/f20nMp70yGRn7kg9e3XcH
VIWaiJ9EFK6V

kzSt/4AKCgowwqAjkInwNyoqisfjIXIFWfPy8rCJiokpfLa3t5OHhRByAGwUSE9Pp8pMFZ
CMTVIZWSwW

C8VGR0dRBIWmCsWwRDFIuTe0aNf6poXrcZztpPtlckw4SbFx6QlJuQnJeQkpuUnp+ak5h
Wl5xeBuel5p

JoOZU8jMJ+4wk+h1hYzOSUV3PIqLpNVR5iNSLe+iltfzeNXT0B3grtCrlDEBXfF4noJYb4
YFIfulIge

EmGjWNAobhRQlggFEoFQ0iCS8BvFdQIRr6uzpSA/q7KsYLS/e1TW1sytzI0KDrQx8TG66
X5Dw1njrO2Z

o1bH9pkd2mF6QNV03xajfZsN92/SP7BJ/9BmXbXNNw9u1Dq0Ue/EdhstdTfTy9ySJH5lJo+
Vy63Oq67K

rmJIV7KymaxMJiuDWZVGLMFddjbMqsmtrs3j8Aq59UV1vOI6XgmvvpRyo6C8qiotPd0/K
souJNA0wEff

x/2mN+jrdN3R5nJ5cYKir7Ovl/hP3k88dSXrfeCpK2rynSoET1higqZErQLUmKApYkPkrE7r
R5aHh0d1

dXVoaGh4eDiGPT4+fmRkxMTEBJkZGRmAqKWlZWZmJg4HCiOcZbPZ1LHAYcLWxM
RELS0tVCkpKYkghZK+

vr5IoDUNDY2qqqqOjo6YmBi0Vltb6+zs3NjYWF5e7uPjg0ZQBmE3ErRo0fo5qBdTutL98t7
BqOik+ISs

pOS8pJT8xNS85ExGWm5Rel5Jem5pVgEzp6gyr4TFKK0uKquhot7SirqyKh6zml/B5lfU1A
O9LE5DdR3B

XeCWQ+BWjCVcJ5aAuzwQVyppaJLCjVJY0igVC6RiUZNE1CQWN0vEzVgSJlfF0haxpF
nY0i6WShvSkmJS

4yKYeRn9TYKxkf6Gohw3XU1X7SsuWpcdr5yxOXvMQn2/mdou04PbTferGu/fYrhvk8H+j
Xr7NujuXa+z

a63W9lXXt6+4umP5zUPrdY6rXjmySczOFdcV17EZrMpcdg2DVZtfVZNbVQPiZlXV5MBs
Th5cw2HUcgs4

9cV19SX1/DKY31DOIxMCYXk1OyszMyAm2iEsxDzQz8DXQ9vHXdvDSdPJ9kp5SdJAX
3d/nwJhTa+ij3zw

Sk6weIpUKMQ+KMzFVAkkIIAaMRPiMGqV1n9CDg4OAGFxcbGOjg6CVKA0MjKyqKjo
1q1bmpqagCiQCQDj

oCBCBW6Tk5P19fWRRs7Y2Bg4zeFwUN3V1bW0tPTatWtgLeLp7du3o4y2tjby0ezu3bvHx
8exCX05Ojp6

e3szmzcBwUFqHyaPOC1atJ5tIdLC2U56oE8xFB2dnJCYk5JWkJJRlJTOSM0qysgrzmCUZu
SXZSHqLalm

ILKLyomQt4zFK6uqL2c1VLABWbWiai4sZNcJa/kiToOIKxTXiRDgNsF8aTOWgpbWxqZ
mQUuLsLVF1NZK

WdzWKmkn3NTR1tze2tzRAre0EybTzeJmkaRZ0NBY4+1pn5UcnRzunxrqUxQX2I3LFBdle
elddb561kHj

pOVpNeMju40R9R7cbn5ou/IBVdMDWxD4GuzboLdnne6uNbrbVmpvXXFzy1KNDQsubV
mgc3iD+ub5Gke2

NPFK6muKODXFtdySGm5RNYdRU5dfw8mt4ebXcBmc+kKYC+42INQ3lvIF5Y2CikZBZ
YOgkt/IbGgsF4kq

OHV5ObkhcbFOEWGWQQGGfp46vh46ni5aznZXK0pThhTyAcWgYmCgf1DRN4CJta+fG
PJ7UiH+HuiEKO5i

Wp8UAilqSf3BUqQBYKoYrSfX1OF9hEDf4OBggVQKKLa1td24cQNBcFlZGQafIjHgamp
qColiSBUKhVZW

VlWuF0cEBwhh7rFjx/z9/W1tbRHvAq6enp75+fmoZWRkhCrXr18PCAgQi8UoaWZmduXK
FSAZUPfy8kJh

6rBS+wLR+0OLFq1nWAQYlB7sHxiOjk5B4JuWXpSWWZyaUZiWXZSRX5JZUJbJKM8ur
AB6C8trSO7ymGw+

k91QVSNk1QrZPAmbJ66th8lPaIVSkrsIcFtgQXMrLGxrh0Xt7eKODklnp7Szs4lwR1NnO9z
S2dHa2d7a

2Ua4o7UFJjLbmtokHZ2SsFDPpBj/SF/HSE/bWE+bVB/7LD/HbB8HZ42T5sf3mh/fY3J0h7
GaqskhVbND

28yJparJgc0G+9br7l6js3O19rYV2luWaW989+aGRVobF17eMPfilvlaB9eseuMPQS6mTY1
VnNoSTI0Z

p76Uwy/m1BfU8vKx5PAK6xqK6hqKeQR3yxqFFQJRpUBYKRSyBCJWo7ASOSJJVX1DI
aMgIjHBNTLCKiTQ

2N9H19dd29NZy4mk7/BA7+DAEGZVxRAA3N8/0K8ghvyeVDDtToqaeSncYu6eFCZ0d3d
3uVyOfKrMNAHj

WKlitfqEAgaoBFX9WZVycJ9AJSUIVVVVGGfqJ7XT09MRpCLYRcibnZ2NtLW1dU5OD
o4CCvf29IIXQ1hC

LBYLSEYCUW95eTkaQWtYtbe3B9EBWgaDgWZRC8wWCAQo4+LigtbQeEtLCw4Bmk
X577XDtGjR+skKpzro

AA8NDA0PDI3GxKQkJ+enZRSIZ5WkZhal55ZkMsqyGOVZDOK2M6OMXVTJLWfVM1
kNcFWNgMVByCupqZfW

1Es4fFhcJ5DUi5oaJM0IeSnuClraYFFbh6i9A+id4G5nM+mWLSJt3Uq3d3e0dbUTxmpPZ3df
l0Rab6B3

KTnKK9DRKMROP9DiRoDJVc+bZyxP7rZU36W/b4PhgY1mR1TNYbWtlmqqsNnBzcb7N
+oBvbtW3dyx/Ibq

0hub3r2xfpHWugXX1s7VWD/v+vZ3r+1apr5h3su/U5G380HfGi7QW85tKCWJW8hrBHRL
6gWIMMVdoYQl

krCEIsICMQFggahK3MRqEBQXFEYlJbIFRViHBpkEeOsRsa/zDcS+IWUpI4N9Q4MjmE8
HhgFgQE4xMHgv

woFUMOdSomZeLEdHR8fGxuZs7LB87733MOM///77mNyjo6OxCQeJmxFC9StS+rRHiQ
gZFKrSGCJBsHd

7u5upNEFKIL9/hykHOuHCKMEUYnJ1QeFQVOMSJFVpxemxnxSk6sPaxb7RjX7JPtJixatn
7hwjuM8Hxoa

hYeHRkaHRsYSEjNT0xmZWaWZ2WUZ2aWZeWUIeWGgNx/oJW441zOrGyrYgsoaYTV
XXMNRqq1v5jQ01xFu

4gma+KLMbmlrY1OboLld3NopaVe6qbMLbu7sbu7qbuVpmXB3u6ynXdbdQbqzT9bZ19Mp
7+zs7ero7SYs

7/TxdfH1tAr1sghx0Au01PTWu+CmecLsmKrFUVWLw1stD281P7yFsNomy8NbrNQIW6o
Rq3q7V2nvWH5z

+zIt1SXXN797bePCaxsWXl2/4MqGBde2LLq8dcHZLQuWvPq/oT62DfwKDP9Z1wiXI8zIC
0obhOUNQmaj

iLBAXAX0iptqJM21kqZakaQGForZQkm1uKlaIGYWlcYmp3hGR9mGBZsF+Rr6eep5u2i7
2F+vYqaPDimG

Mb4A3DCub4gLHVzuTJUKNfICOCC4HMISaQBy4cKFWlpa58+f37Jly/Xr19XU1AIDAz/4
4AOq5DQhE3M3

qqMiBWzEXsi8desWluAxWIsEYq+enh4kEIpRhKbqUolnWMrBpkWLFq2nRQODQ4oBwg
NE+Ds4mpqWk55R

mAnu5pRl5pblFBDcJR62KmUBvaWs+nI2n0JvVa2IQi+X38prbK0XYNnMF7cKmzuEze0w
0Ctt7ybc0SXt

6Gzp7IZbu2StXcCtbMI9HX3yjj4Z5a5+eVe/rEch61bIOvrlsGxAflnjtKudfpCLobfZfQ/dM45
Xjtic

3Wuurmp5bKvIkS1WR7ZYHN4MA7cWhzZZqW2mbHl4s8Gu1To7VujsXHFj27LrW969un
HxtY2LrwLAGxYC

wBqb5p3fMn/b4le2rnqnkV/RIGbzBBU8QTlWAY3iJiwQFwhEJNRr5QN9EpbuJJmrljKgQ
kGE5k14ubK

krKEIFSv6Ci7sBCzYD9jf099b1dtVwdNFjNjWNE3qBhGwNs/ONA3QHzkO+2PAs5AX3A
U+NTW1m5ra8vN

zXVxcUHizp07R44c8fDwoEo+KNQFm//4xz96enoeO3YMxEVJtINMNAgkI+Hj49Pe3k72N
op4miqApbKJ

Z1rki6ZFixatp0GYkYZHRoeHCY8MY2V0PD0jPyu7BPTNyI3PZjBziypzi6uAXkZZTXEFt
7S6vqKWQC/M

rpNy6ltI9LbVC+BWvrhN2NQhaiFMRL1tXRR9mzqU3G3rlrfLettl8o7evo7eXspd/TCgS7hb
0det6AVx

ewbkHQq4l13POXBwp5OVto/tTXfDC65aJ2wuHDA/sctcfZvlMVXLIwDwVovDADCCXQ
B4M6JeKvbF0mjX

Gr3tq3S3r7qpukJz89JrG5bAV9cTDL4CAG+af3HLgh2LX/7jL1Q62ngtHXy+qIovqmgQlcM
kdwkLJVUi

KWLcGmkLp7m1TtqiNIHhZo6ktVbSwiplJqWl+UzS189Dz9Plpovd9aqy9NHhgZHhMYwy
sIrwFx4iPiq8

JxWgcVKABJYg4u3bt8+cOQP0RkREIAKOiY15/vnnBQLB3r17qZIPanh4GEDV0dH56KO
PcFllaWIZVFQE

EqurqyMCNjc3d3NzW758eWdnJxKurq5Xr15FR3p6el5eXqiINHXDU9ncMyS8NIwArj/wM
j98QBguWrRo

Ofq/FWaeD+APPnr//Y8+eP+jD9//8KOs7IKc3NLs3PI8RIV+UXV+cXV+CbuwglNSxStj85m
1jZW1Qlat

uJoj4fBb6xrb4Hphe4O4o1HSIWjqFLd0Sdu6J93cIYNbCO7K2nrAXUC3H+jt6ld09fdTlg0qJ
i0fGpAP

KXqH+uRDfd1Diu6hgeDIsKtXzzta3nA3v+prfsVL76zz1aN25w9YqG+3Orbd8ug22IL60Je4/
7wVacsj

2wgf3ma0a4Pe9rWwtupqrc0rrm1cfnXjsqsb11zdtOTqlnevbF14acvCPUteeevFFzjV+WJprVD
KFkhZ

QmmlQFIhkgK6hMVN1QhwpS21zW3c1jZe84SbWuua2uqa27kt7bUV1WkZWf4xMQ4RIR
Yh/iaIfb1cdVzt

NauZWR99+AGGmRpojPAHHxKeKhWwYVIULYAK0HfPnj2hoaFOTk4g6NmzZ1H0ueee
q6mpoco8KIAT8TGY

iuXHH39sZGQEtgO0ampqwKqVIRXyQ0JCEEaDxEgDuuR1NQ0Pz+fulmNRpSXZeR97G
dG1HhiAPGWx8jQ

okWL1n9bmIs+JH1vUstrKLSzJLcvOK88rqGIAvcXs/NKaosq6Uha/vLqxolZYxSHQO0lfnqC
dLwJ6OwXS

TmFzl6S1u6m9RwndTnlrV29bTx+429Yj75D3w519is6+/m7FQLdCQZkkrtK9w4O9wwN9w
/29w/3y0SH5

6HBQWMiFCycNtE47GJwLttbyM7roeOWIwyU1j6vHg7TP+V0/bX9qn8OZA3Zn9tuc3md5
YrfZ0Z0majtg

U7WdjqcP26jvN9m/w2DXVsNdqjrbNI7btFpj0wqNrcs1ti3TUH334rZ3dy15ddZfng/1c2yS1o
qJ28hs

cTMi3SoJsWRhCUtbappbuS1tda3t9S1thFvb+c1IdPBbO/ktnZwqdmZWdlBcrFNkmHVogG
mAl4G3q66r

w83qimzlsD5cKmDDpEgEE4EaaOHi4oLEJ598cvz4cSw1NDTq6upQgSrzoEBQ0MXAwA
BktbCwQOEjR44g

8/r16wqFAkEzGgGJ5XL5+vXrgaJTp04hFkQVdNTS0gJCoyL6VTb3DAkvCuNA0ReDQIs
WLVr/bX38ySeY

jj76+BPMSvDHWMnOLcrNL8/JZ+YXVTFKagiX1RZX1pexGphsYSVHXMWVVIG+XC
m3oRXorRd2NiB3i5h

U5eouVvaSqK3Uw63dPW2dve1y/oR8rbLejt7FXBX/wAC356BwZ6BAcry4cFJ940M9Y0M
9o8o+kYUvWMj

fWMjVnY2VzTOXj13wNn4oqfRRT/Di3YaajE22vle1tnO5jGmWr6aZ9w0jtufV7M4uddUfY
/FiQPWp9Ws

zhy2PKVmceKQ6dEDJof3Gx7co79/l+7eHTd2q17dueHC1hUXVJde2rbk0o5le5a+PutPv/B2
NG2Wcpqa

OdLWWmkrW9JMcFdMLJFmNxGBb11LO6+1g9/STrqzobmzoaVL0NbT2NxVV1WbnZ0b
Eh/nHEXQ1zzAy8jH

Td/NQZtdmascaVIYXoz4NN1H36lCnEr98Qc/Pz8tLa3a2lqUVvLkASnr3LpF3VXmcr1AjkQ
i8fDwSEpK

kslk1JdTGQwG4trm5makgWfQyMbGJjo6Gi2jLtoBj6k4+FkSTV9atGg9ZZpO3w8/+SQrtyg
nvzw7n5k3

Qd+Ccs50+nllrDopp6G1Tvik9CWiXhK93YpBWD44JB/EclA2NNg7MtQ7qnTf2HD/2JBid
EAxOgj09o+O

3tTV0bh8Wu/qcS/La256Z/2ML/mbaMTbG4QbX/W+ctJGfa/54Z02pw9anj5odvKAyYkD5qc
OWZw9Apud

OWJ04vDNw/uv7NtxYcfmTbNf3/z2G6rzZ+1Y/NaBle8c37jw/LYlGrtWXti1auWsv/o5mw/2
Nbe08Zra

OE1tNdLWapLBbGkr0jXIbG6va+msb+tgAHcJdwlaugUtPcJWuQD0ZXFzsvNC4xNcoyNs
wwItptA3hxxh

pWamL9gwowBCqLGxMSIioru7G+To7e0FI7GJ5Ox0jY2NUbU++OADRLHEg0bkvWiAB
8Sl8iHqZixWqXw0

i8Rkm9RWiFp9NoQXiHHAK/34448/pUWLFq3/tj759JNPPv0Y/pj0R59+8tGnn2bnEb8bmM
Ng5hdXF5TW

MkprC8u5JVX8supGZo2okuAu4WqelCNo44k6+OLOBkmXUNoN9IpbeiTtBHrBXQq9bT3
9HXKFEr19BHp7

BoZg+dCwfHiIMojbO2GCuGPDA2ODA2Dw2NjA6Jiunu6l8yc8HQxCXA18TC4FmmrkBj
q4a589uHjW5tdf

3PDyn1Rn/XP3vFkHl8w9uWHFxW0b9I7t0z22T0cdPnBsw+rTi+Yt+edLb/3+d6/88oU3fvert/
/8+zkv

/n7N7Jf2rZx9RnWJxp7VF3evWfnWi07mOv1yaUt7vbSdK22vIbSxJW01ktYaSRvStchs7uCl
dvHbuhtb

u2BBa7cQ6G3uEbaAvj08Ni8vtyAc9I2JtA0PMg/0NvRx03Nz1KmpzP30kw+Jgf70U8z7MB
LTpAI2/Psi

cE1GeI8QVUYZD5IRIehLZQK3WCITOdSmZ0lAL642gF5c7HxGixYtWv8lfTrhTz779JPP
MB998rHSn378

2Wc5+cV5DGYuo5JRzGaUceBCZl0Jq6GcLai0FVXVSWB2fVMNv5mLwFfUyRd3NUi6A
SJRq1zUKpN0yJuB

3p4+Cr3tMkVn7wDB3X6Cu7LBYfnQCNw3Okq5d3REMT7aP+GBW4QHbw0P3RoZGh8f
GbulfeOGsZ5mgLtF

oJOur/mlOBd9q4uHts1/ddUbL61569Wlr/zT7T/8dtbvfrPor39Z98YrOxfO2Tbv7VNB153fuUI9
8+o9

yxavef2N+X/42+I/v7zq5Tc3zZ67bdGCrQvfWT375c3zXj28du757csv7l23bNZfQ3wd21sbWj
v4TR11

TR2cpg4QF0aCCyPqBXrbewQdMmF7D+G2HhHo2yITtvYK7vruYKiwpLYxCSP2Ci7iB
CLYF8jfw8DDyc9

Dovx2acfYazJ0SaMxDSpgA3/vgBXZerhohisjAfJiHBSyixSyqxnSFTcD/TiYucuLVq0aP2X9
Nndu5iD

SH/2yd1Pp/izj+/ezc4vyS2oyCmoyi+pLSjnwkuUVvNLqBmaNoJijZvGkMJvfXMNVqRN21Iu
7+JLuBmmP

oEUmbusVt8ulnb3NQK+sn0JvhxzoHSTQq1Cit3d4FO6foG/f2Kji1j0P3B4bvD06dHtk+Pboy
Pit0bFb

jnZ2Jrpafm5mLiaXfUw1HK+rH14978CqBYc2r1LbtmH3+pVrF8579/W3lr3+9vLXZ62aNW
vrooXHtm48

vn3T9iUA7fwNb89b8a85S1+as/b1xarzVu5fuWHPipXb3l20Zva/VOe/cmLTosv7Ny589Q+R
Qe4DA+2t

nfzmTl5zJ7e5C9w10NvcWQe3dPFbu/kdMkFnr5AAsEzULhO3ykTNeN3yxly5ny9hllYkJad
4xkbbR4VZ

gr5+ngaeznrc6oK7n354l6Dv3U8+I8Ycgz9NKmDD/42UEJ5JyiCRIDLrGRLQ++GHHwK9u
Nj5/AfpLmla

tGjR+neEaQRzEOm7n37+2aQ/+fzuJ59/nl1QSnzHt4hF3HNmCuHiSl4Zm3raWVLNa4KB3tr
GVq6os55E

b2MTYkC5pL1P0tHb1NXXAvTKFRR6O3uB3iECvQMUekf6RkZhxdhYP2XEu7fHFO8pP
XBnfPDO2PB7IyO3

yb/TNH4rITbmpsY5S70LGsdUrTVOBBhrh1gZhdmbW2ie1dc4eUptz9Y1a/Zu2bN70/5ta3ds
WblJdfW6

ozt3HIZV3bho8YrZs5e+MXvzu+tP7Dy5b63anlUH9q7ae2jd7gNrN+1evmTHwjcOrnxn38p3
Dmxacub4

3pTUyHbiESpecze3uZvTBAADw911cEs3v03W0NEr7OwTdfTC4vZeSatc3CwTtvQK2voE
ojZ2WVVyarpP

XKxDTKRvSL9xgLeRu5M+h134+d2PPr/72Wd3P//0LjHmD87hj4p9ldj8kaRsdCYpwUtKmf
UM6QnoS+F1

qidEJu/PokWLFq0fIkWjk/T97AH6ZjFKcog/LVmVV0LRt4545Iqgr+gefRtaakj68qTd/H+bvp
PonaTv

EElf4sGfsVtlhQVv/uPPB7YuczW5Yn7+6JXdm83Pnzi3a/PedYsPqa7bu3n7+hU7t6y/oLpRa
9vG61vW

nV67XPXInkMn9x/Zt2nH3Nfm7tiotn+X5poV59esuLh6yclVC3evW7hxy+KV+1auUFu1CG
H04XULVVe8

88vnVA6pbW/tbmjq5jV1c2FpF+Gm7jrktPTwW2UNbflG9l5he6+ovRdxvqRFLm6Si5rkgpb
expYeXhEz

MSXdNzbWMTTrSOsjPyN/b2NUR9C36/O6Hj6Gvko1TBFooUxOPSv2IUrZ7f8vKLFLKrGd
Ik087Y7i/mEHI

vPuAJ0qS/2IxY01atGjRenJNmWs+v/vF3Ul/9sXnn37xBWLfnEJmTIFVfmltQXldEbO+pLK
hvFrIrBFX

cqSsumY2v7Wmsa1G0FYn7qpv6gF9Bc1yEXCEeLBdPkIf6p4zDPT2KAZlA0PyweHeoZG+
4VGYRC+WxGe9

igdi38H3Rt7/7MOBoSEuh+vj5rps7qzje9YZXNqvf2qb8eldmoe2bVk4a/Wcf7314t/feWnFmo
VX9mwL

37M99eTh7H3bPBfM2qm288TFo+ffnb189fIDZ857b9zhv3JTyLwV7n979eKLf1sz65UFK+
Ys3LF08dG1

S49vWHJ4/WLVFXNWLpndIKhuxUvp4Tf38EjXwy2yhkkjxm3rFbb14rWKW0HfPmkzjHS
fCCWZrIzUdP/4

eNeocNsAX+MAX1NXB/26mtIvrvvgeQz455g/O4SoAwwP6GLwAM5T0+FGlZOz9Un5AS
kqZ9cRSVnuKhVeN

wfz444/v4n0+XVNOh+kmDxYWn1MLUsp/7glQV6Zo0aJF65HC/DExv8xAX8S+WQXM3
GJWHvFNX25BOY+k

r4jJFIfUSqq4zdX1CHwJ+hKxr4SIfUFfxL6iNoS/BH1b5Qoq9kXg+4PpO3JndPy927k5OVpX
NA7v3GKr

f9nkyiHLy/tuHN54bOPSXcvnb1n6zoq35695Z//WFVZb1qVs21R2+jD36O6Eea/tXfLWyhV
vLVr8xuJ3

3th05XqM2pni1TvzFm9M/Ncc01fe2L143tqNS1buWbHsxIaV57asUlu36IjqqvUr5zcK2S3dg
qaehqae

+iYZj1iCxLKGSbf2NrbKAWBRK0Fcir5NTb2S1j6xtJPH4uSlZQTGx7IFhdv5+xf+Jo52ek
+EX1JJMif

EosP3r9DGOmP7tx51L3iHyYljh6Q8qYzKWXWE4t8VOupFnYSY0zR98vp+uLLLz9/iLGJ
XHyhLKRcvV+4

VsISB5JapUWLFq2HiZpJSH/x+ZefT/rul1989uWXWXIF2QXMvBJ2fiknv5TLKOMVVzaU
VYvKQd8a0LcJ

sS+7oY0j6iRiX+m92Bf0FVOxb09/m4z4ohH5vNUQLFMMYqEGewcJ9PaPjBGeoG8flrfH+t
9TWgH6vj8+

9tGt4dsj7334PquqctOalcf37TDXOu1ieDHGxcTXWFNH/eCN44cObVmx9MWtc0XD29z2
b0lbdFWklMH

WIe2hi18devhLbvP7ty7bdHK1/741sXTrmrqyWu2Januy3l3hdO7C05uX7XvyKZdJzdtOrNp
9emNy8/v

XAf6ammcHB6Vt/YIm2WNzTJ+s7yeWCLklTdOmop92/vEbX2Stn4pRV+Ev4BxU1d9NTc
vMzskIR70tfH3

Qexrbm+jXccp/fLLT4m5eWLMH5yjVYaGbw0Njw8Njw0Njw6ODA2ODA+Njg+P3Boe+T
F//GD0AZFfO1JK

mUVKmfXEUIZ7igUA48oD9P3888+/uqcvJ/zFpHGMpqzi6HzxFap8qVynKuEKikpSQstYoi
i1SosWLVop

0+Sk8yXmDEw4pL8gTEw92QzyqaviatAXgW8hk19S1Qj6MmsklbXknef61prGdo6ogwh8y
VBR2EIEvkCv

pAP0lbf09CHwhbt6B3v6h3sUwz0Dw7JB4mnnvpExuHcEie94//gYZcV74xMeA3pHPrw1/v
HtcSzfRxh2

e+eWjdYGN8w0T4U7G6X42Mc6W7vraRueO31275bzBw5c2H/9/H6H80diTuyNO7s3etcy3
a0Lt+ifOn5T

bc+pjetfUvnfvWtOb15ttGWj6yn1uG3rrVWXnTm++fh51QMXtm47v2nt6Q3Lzm1bc0lte7Cv
88BgF/E9

IpmAuM8sJ7kra2ztFUxY2NYngtv7Je390rb+ppY+uBnLtl5pc3cDm1eQmROWkOAeFWEb4
Af6mtnb3ORx

y7766jNibp4y7NOKIu+90yMb65EPdvc6evp6ZYo+Wf+QvG+4TzHQp+i/75f4fyRRv271oKit
1I8/KLMe

Kar8MCmq1qNF1fq/L49LBIS/CFKn0BcHgLqfaYukoBcZ1cnstgXX979jDx81MqXn37yC
XG+fPXV53eV

TaHxixcvgr5qamr19fVU5qTQ46effqpcmdAnn3xC05oWrZ+37k1BJIAJ+mbIFxM/6FvEyivj5
JfXMZj1

RROxL+iL2LcKAG5orRW2c8nYF/RtJAAsm6Rvc3dvKwlg0BemnrrqGRySDY3Ih0d7Sfr2j
Y0RUS9J34H3

xuFBwmNDd0ZH3h+99dH4rQ/Hx+6Mv3fn9qWzp3ydrPQvHQmw1o53swq2MPQxMjQ+f+
7q4f3XDh+9dvCy

xt7r6utPnlc9a3xCe/U/3tLYvtns9F6Do5uu7ly37K//2rtQdc+7Bw+tOn1w5anDq04cX3P47IYD
Gqr7

NFS3XVbdcH7rKiV9vR37Fe3tMkmLTNgsb5zml15hS6+Icmuf8rYzRd/m3qYWuQRjwOYV
ZuSEJyR5Rkba

BvqbUPStryun6Ksc75mkMjz2de/Ax8O3Pu7DAA0oCCuGexXD/QOD/QMKhfIH8p9eKRH
3BHoYHR+mH6v8

GPKDi9QHtF8T+urrr7+c0V98QTwfx2+sn7dgboOwgSr82ccf3f3yc6Q++fRTMgcF0YJSIyMj
165dU66Q

+uyzz0BcKo1O7969i8RE14SQgzIAsHKdFi1aPzvdNwt99fVX+CebUZJTVJIVxMoFfZm8gg
p+cVVjKUuA

2LcC9OU0sXjN1fzmmsbWokkXFfs2NHULm2Vi8nNfaSfoS37fl7j5TD7zrBjuHhjuHhyCe4
aGZcOjsuER

KvZV3BqHB+/cGoLfuzX83vjoB+Mj74+Mfzg6DgZ/cPu9929npSZG+Lm6mmvaaZ9x19cItj
CwvXLJ7Nw5

/RPHdY6euL731NWdx27s2W927JDBge3r/v5bY7XNNmd2GaqtubFn9eFF7y7748unVm1V
X7bp7JptF9bt

vLxx75XNe65s3nFly9YrW9df3rbm+oEtO1fMFXDKOjuJryu3yMXNvcLmXsFUt/SJJ93aL2
ntl7b0NzX3

NxMGgPuamnsENfzirPyIhGTPiEjbAH9TPx9TW9CXx8RcS47zQ6XS2fupYvSLLsXowPgt
2cCAXKH0JaLe

AUS9Tz96IYp5T7NGR0dBX+pzX3LMH0pf0l9fvHwBS/WTx4gb0Xc/u3zhvIm5mXeA/xZ
V1Q/efx+bLl+8

FBUZuWHDBkAUpNfR0SHKq6uD9OjuzJkz3t7eCQkJ6E5LSysyMnLLli0oKZPJVq1aZW5
uLpFIUJ4WLV0/

Y81AXyr2peibNxH7gr5l1UKKvoh9WfVK+tYT3zgivqYjArXaYBnxlya7+4i/dUWEv/0Ufbs
UQ10Dg1MA

PNI7Ngrfi32JP7IxNnQbge/YFPreeu/9W++PD/q72Hhb6xhfOGB39bjbzYump44ZqR8zPHbc
6NhJA7VT

BoeOW5w4YXFc7fTKhUcWv+l08bD9uf36B9br7tt4TXXDot//ds/cuRc3bb68eYsG4a2XN26
5tH7D5Q3r

Lm1adX7T8huHVE/tWDPS26IYaEf03iIXPSI9iQ99m5t6pc1ySbNMWNtYms2IAAn3DI2wC
A818vE1sQN/6

J6Cviwfh0b2oa+Djzv4BBL69iv6+/oH+foK9gO/gwKCSchPq75/2+/y0HiOEp7dv334y+n6N
2Pfg4QNf

fv2ltp52v6IXJ4W2pqZQKsYmXz+/rMxMJMDgL7/4MiAgoLq6Gu0bGBgg8/r16++//35QUF
BZWRIWISq6

/eqrr4KDg0tLS3HgZs+eTW1CJpWgRYvWz1Iz0DczryiLUQ765pTWgr75ZXUF5fWFTP5k
+FvJbariNbEb

WmqFbXXijnpJFxX7kn9sskfc liPtIDV3yVu6eyn6Et87mkJfCsDykXsAVtwam0bfUfjOyNj7
47fvjH1w

ezjM2ynOx9bmmrrBse16atuMj+2/uW+33sFDZuonbc6ctzlz1vHCGd2923a+9Q+LY7v8bp53
vHBYd+8G

gwNbz61fevjdUv//DsN1fWIdDW2rL+8ad2ljevOrV15dvWyM6sXnVqz4OSGRSe3rxzskbS
3N7Yj8J2B

vsLmPnFzn+R+Swnugr7ypma5tFkmqhMycwtjE5K9QsNtAgLNPTyMQN86Xvnj6fvrXzn/6j
davSNf9wyO

9Q32I+4d7B8Z7Bt9/70PxxkZHQV9EVx0dHbdu3cJEL5fLafp+Xz05fRHsxifE7j+4LzQi9Pip4
1HRESh9

XUNjYHjw7ldfJicnpyQ137n9nq62TnBQ0OHDh/18PhoHfb/44guEuTg0APC1a9d27tzZ3NyM
eBebAGmE

xXV1ddhqaGiIPrAb5IcO90TuFS1atH4+mjYL3aNvZmEV6JtbxgV9GSSAQV/isecJ+rL4zRR
960QdfGmX

oJn6zYHuSfo2d8pbu/vIm88DCOpG AJhiMEVf+SgF4BHQl/gbk7dGQd/hO/Aw0Dt6Z3T0ztj
4e6N3bg+B

vvE+ tq46Z27uX39t28obuzdp7tyis3e32bFjFgiCD+67tGHlqRXztXas9bh6yvfGeVeNkxbq+3T
2bjmz

bsm5DcsPL5274dW/Hlk29yi8dN6xpfNPr3z35PL56kvfVI8xW33d3NO7Vva0cFtaecQfj+wI3
NQrnGIR

hVuSuMAwIhKgt6m3qUkO+jY3A8A9Yp6oKr8oITHZNzTMLiDA0t0d9NWpq3uC2Pd/nk/
6n1/ay4dB31t9

g32Dit7hPlycjI6PjMt6ZL29vZjQseZr6wOGqYSSKrSeTE9O36+++uJPf/7D+x/eoYr9+je//PrL
L65f

0ehT9GM9MzOztLjExtqmkU98JOzu7I5VVYUjYmxs/NFHH128eFEmk6EXbGpra7tw4YK
Pj49QKMSqi4tL

ZWU19sTMzAxIpnBLcZcSVmnRovVz0gz0Tc8pSM8rzSioyCyqzinh5JVyYQC4qKKhjPibG
5IKjqSCK62q

J8NfQRtX2M4TtfOlnYJmMLhT2NIlbuuWtPc0kb8zSNx/lis6+hTtfYr7GTzYM6S0nPxhQcX
4yMAteGjo

NuVhePj20Ohwr7ejRZC9ofONUzZn9xmrbdHevVFr+8arWzZc3bLx8sY1p1cu1t2z+eqWFfZ
n1fT2bjY7

skdr+4YL61dc2LDq4KI5x1cuPrVu2cm1Sw4sevvvgondgtcVzTq5YeHrlgrOr555eO+fc1sWH
Ny1o5BR0

E78TgagX6J1uCr0T3JVI5WKpXCKVSSfo29zcl6kXVzOKkxKT/UJCQV9rVzcyjG1s9LvcJY
t8Xnk994ZdO

vSPfyIZuKwYHRhT9t3qHbveNjgwMA7QIdidx293dTT1jTD1sPF2Dipk9RUNDQ8rUz0nU5
77U37r6htDX

33zz1YyWSEQ3tbXIMtDXdvY2An79pQvnB0aGsB4RGZmSnCLr7lbdssXfz3/Dhg1lZWU9
PT1XrlzBVtAX

wxsXFwfoXr9+HdzFsdudZUtYWNi6detYLBZWNTU1UfLB562I3mjRovUz0n2z0NfffI1/Ur
PzU3NL0hjM

zGLywSsi/AV9uYXM+jKWgPibGzXiSq60kidl8Ym/N8kRtBG/8ivpbGjqamjqbGjuELZ2idu
7JR0y5fPP

sv623n7Qt6N/AO5SDHYpBkDf7sEBoFc2PAT69o4O9Y8CwMOT9B2E3wOAB2/dVrjaGvl
aabtrn3G7csTm

5C6zw9tN1XYa7t+hs3OLluqGG9s3WB3fe33bquuqa9b89Xcr//jbo4vnn1mzatfst1a/+NcNL/9
zz/w5

p9avPrF6xck1K0+tXXVm3arz61dc3LDs4sbF5zYuOLt18a4Vs9jlGV3djW3ETWZR0wNu7p
dSbuqTwNJe

sZRgsFQqb4Kb5NJWuZQvYReWpIamBoSEOWQE27i6G9k56HPqmN98gwn/UbOryvMkfft
GvpEP3gZCQd/x

vsGxvqGx4ZEPPvgA6MXETUXAPB4PyxkJ2t9PkHpoCLgln5Um72CTifsAPDSk/JoQwkEq
8SDLgXll6lnR

A/SFZgAwrkC//PLup59+TG7G+YDMbz7/9JOvvvj87ldffEEexa++/Io6mkRiQqiG+BUJcBQ
JRNhUR8iH

yCJEGplYTi1JkpcQVYYWLVo/G903BYG+mCkS03OSc4pT8svTCiqzitm5pSSAie/+1hVX
8ok/OckWMWtF

FVwRiwp/G1s4glauqK1O3FYvaQN9BcTv33YK27vEHd3SLnlTt7xZ1tsi723r7ZtgsKJzgHD
X4ADcPTwg

GxkgfmN/ZFAxPqQYx3IA7r+FZf+HH495u1oEOhj4GF6yv3AA9LVS32V7Yp+1+n7jA9v1
9mw2PqhqdSH

7t6N59a9e2jh7D1vzr67rsH5i1Y9/d/bH191sZXXtszf/75zRsvb92qQXjLVdXNV7eu19i86tL
GJec2

Ljq7eemBdfPrqxmt7fXNPYKmXqGUtEQugMm0iLR4grtiiYyyRCKTIgKW9kibe8Sgb0Fp
WkJqYHC4Y0Cw

rYuHke09+mJ4HzrBqjz/QvILv3ToH/6mb/D24MDQ8ADxpBUSAAZAW1lZqaKiIhKJPv30
06NHjyLYmiH2

HVT09/eCvthCbBwkviqMnOHhQVgx0N8j6+7tlY2MoOoQuEu1ACYBtNQv+yKztbUVAA
fmfx70hXBIKE85

ByZuBH0FEhNfFPuS/Jcg6Odf4wyZUu9+TSUoBdRp95MnN9H0pUWLFjmJTJl5SPrgpWal
5JWkMpigL3nz

uTavlEP+yUnipwbLWI3l1Y3MWmEFV1jJk1Tzm9iNzTWCZq6otU4Ct9Q3tTW0djS2dcKij
m5xp0zSLZP2

yJpkcgC4tbevrV/RroD7OwYUcOfQQOeQontYAQDLh8Hgw7Rwf6xwb6xgb5bAHD/7Q+
GYiN9fOz0vYwu

u1w77nThkN3pfXYn91kf32t+ZJfRQVXjQ9uM1FT19m+5uWvj9W0bjy1dcnTp8oMLl+x6e/
7mV9/Y+c6c

w8uWnVq37tyGDec3bri4eeOlzes1tqwDfS9vXnF+47Lz29buXbOoqbG6qYUn7QZxldwVyw
iTqwAwAV0l

fWVicY+ItFjcLZF0S6TdkqYu0Le2oCwjMS0kOMLFP9jOxd344fRF+t6qygsV EJ/7Dg5/qxi8
Q9CTIO+w

YmgEOJXJZGBGfX19enr6hg0bgBCAE6SkvkhzT0OIZYdGR4dA1dGxwffeGx2/BcQiExzv
HxxSjI4Oj42N

kMuxrKysiooKsBb1AFqQCZIUm7du3ZqMicl2nxHhBd65cweXL1988cW30/XNt99+Pen7zg
fyAwPiX6S+

/QZG0ScRdVynMXXqpgdFbaVFi9bPRvdmnm++/fprYob5FvRNzS9NK6xIZVRmFFZnF4O
+xOe+iH0Lmbzi

Kn5pdWNZTWNFnaiSJ2bxpdWNTaBvrbCZI27mSlrqpC385naKvoL2LlFnj7hbJiHp29zb29
LX19rfD/pO

RW/X8EDXsKJnZEBG0hfuGyXoKx9X9I73DYzJykrSHUyve5treuqed7t20uHcIesTe82P7TI
5ssPwILi7

WffAZu29m67v2KCxdf2RJUv2zF2w4625O2fP2z7r7T1z5x9askR9lCqTalaafWbf23Ma1Fzat
ubhp1cVN

Ky9tAn1Xntu2fuXsV3o7JQjXp9JXJBOIeh6gr5ygr6RbSV+gl6Bvl7SpS9Ig5RaUZSakhQVH
uPkH27t4

mNg5GHB5zG+/xYSPEcZQT5oacyQIEU9d/ep/HIaGvh0YeB/EJOk7qhga7VcQQRvi0bCws
EuXLkVGRmZk

ZFB/2VFJIUkNDQCxgLUCVzAjfc+/oPL+B7cGh/rvfv7xnTu3xsYJ7pJ4Juh78eLFDz74AKB
FGkvgHEKb

aBnQpSJjKvxVNv7T179DXwK9NH1p0aL1I+vezEPR98tvv41Ny0rJL00trEgpUNI3t4T4g88
UfYsq64sq

ecXVfIq+VfUSVoOkulFal2iqFTZxxE1cSTOvqRXhL2UCwF0AcI9EJpPK5U33ADwZ+8L9
oC/C354hBcJf

IgIeHZCPKnrG+mVjvQPj8srKXE8HQw9zTXfdi67XT9udP2x5Yq/xkR36h1QBXa3dGzT3b
Li6c/2lrWvO

bVxzev3a3fMWbJs95+DiZXvmL9y7YOH+xYsOvLv48PKl6qtWnFi74vT6lec2rbyweSUyf
H7zmuOb1qpt

3TDQ194hb5J0E6ydkb7KpUwk6RFN0lfUJRZ3SiSdiH2IDZK6wvLsxPSI4EgPvxB7Zw8TG
wcDzj36Pmj1

IKvyq18k/foFh+GhbwcVHyjpOwj6jg0Nj3z00Ufh4WFxcbEmJsZI3L17V09Pd8bYF/TtkXW
O3xpOTIpJ

SIyOiAz+7O5Hv/nt/2jduJqSkhUVODv73tI7SDKamp2dnZ9fT06Ovre3h4gLIWFhY2NjYp
KSIYBebF

YjGKPUsAfiR9oSmnwT36UtglRPxD3q1QFn+cKKCSle5p6qYHRW2IRYvWz0b3ph2Yom9
MSkZyXklKATOZ

wUxIVE0AmLj/nF/GKazgFVTUFbF4pTUNSGDzxawGMQlgaalISgEYETDFYCoCFnR2iS
YA3CzvhVv7etv6

+0gG94O+MEVfGeURhML9XaN93aPy7sH2ji6BnZmmn52+083zjldP2pw7Yn5in56a6o19m
67uWqexY63G

zrUXt685s3nlifUrNffvPbR8xd7FS85uUd2/ZMnedxfvXrhg14J5exYvPLB08aEV7x5buwwlz2
6CV53e

tObopnUeNua98rb23mZx10DY3SjsaYAFXXwi3d0IBgu7QWlhLMaySyjsVFrQIRC2C4Xtl
mmnhE/QNych

PTI40tMvxMHJw8zGwYjDq/j2WwzqvUGeYuWUq/Kb55J+84LDyCBB36GBkSGQj4h9x0
bHxkJCg5OSEyMi

wjw93T/44H3ErwhNyh0qobIz3cH+vp6QF8jY73Pv/jEwtIE4e9N7esfffy+xpVLCkWfl5fHvn
17ZDIZ

6Ovl5dXc3GxmZpaZmYmQNYehITAw0MjIKCQkpK2traWlBcT62cS+IHAwpvnflqWyKCl
lFi1atGjdE2aG

ezz4hrzzHJOcnpXLPHWVlFeenF+RVsDKLKrJLq7JKanJLa0tIOlbWFVXwuYziY9+xZU8U
RVfRAJYUiOQ

AMAUfafcgu4Qgr5dXYiApT2y5skPgPuICLhd0dcxSLhrsL97sB8Aptw93N850ts1IutQtN660
2djpulu

edP66gmri0dNTx80OLZbc//mK3vWn1NdeWzdoqPrFp7dvvriro2nt6xT37hOfcMG9Q0b4Z2
LF+56d+Ge

pYux3LZg7o5F8/YsW3hw1bvqG5af2Lj85Mblxzeu2r9uZVyof4+8tUUmFXY2ArqCrnq4sRP
mUxZ0AcNC

WIQloNsuAHfhxjbCgjahpFPKl/AKy3MT0qOCI738gp2d3C1tHEw4vKpvv8WgUvP5faNNr
hJS+e1zSb99

foK+xI8cDSiGRuCw8LD4+NjIyPCQ0KD3339vYLC/t1emUPSTuJ0qJX2HhxXdPe379u3y9
HLdvn1LR0eL

xpULoO+Nm5pnzpz67LNPNDWvdXd36+npeXp6gr4IrF1dXaOionx9ffv6+q5cuRIUFIR8cP
dZQi/0BPSd

QUp4PiDl5kdKWZSUMosWLVq07uk+HID0jUpITewqSMotTcgpTcwtT8mvTC9gIQLOLEI
QzM4r5+QzuYwK

LgCM8JfJFVTUEa7kCcHg6kYxBeDJCBgAbmhtF7QjAu4AgCXd3U3dPXBjxwYbpbLW/
rkrf3ydkVv5wDh

7sE+MBjuGuzrHJZ3DnfLR7vH3uvNSAoxunrC+OJhozMHDE7s0zq07fLu9Zd2rz+xaanq/Nf
2rnjn+Jbl

auuWrJv96uJ/vLj4H39f/I+XFr3093f/+feVr/9r3ew3Ns59a9O8tzfPf3v7u/P2rlyktnbJ0Q1Lj65f

dnjDim0rFovrazp6mlt6m0jWAro8LBS66vltPH474cZOhMKCxs7Gxo7GxvbGhtYG0o38lgb
CzY3CdnG9

mFdQngf6BkV6+wW7OLlb2TiYcngscmwpPYS+v1NJ/t3zTqND3w0NfDQ8RDwtNTAyibh
49drVQUEB4eGh

H338wejocF+fHAwmH5AiHryaluB3cGh44MOP3uPWsZtbxLffG21tk1axyjMyU26/N1ZWX
sJv4Lm6OkdH

R/X19WZnZ1dVVSkUCktLy7i4uK+//jo8PBwMLikpYbPZXV1dQ8+cEN+///771A8bfPfEog
7Pg1JufqSU

RUkps2jRokXrnjAzfDPpb7/79uvvvotOTEvKLkzOK0vMLUvKI24+U/TNKmYT4W9ZbW5
ZTT6TU8TiFVfX

I9U2lnMbKQBT9GULxLXCafefQd+OxrZ2YUcnSV/CzQiCAWDiLjQB4LZ+eYdC3jkg7xq
kAEy4a0jeNdQt

G+7sH2pra6o10jqlc26/7qm9N4/uun5A9cLOtSc2Ldn17ltr3vzbgr//9rXfq7z9I1+ve+e1fWuW7
Vu9

Yt/qlfDa2W+ueONf7/zl/73++1+teuu1te+8uXHB7O1LAeAFB1YvOrB68b41S66fPjas6GrqE
EhlQkF3

fWM3D0tEwI0dDQ3tMIFhpAnuUm5vaGyDGxH1NrQKGlpgobBNXC+pL64qSMYKDYnx
9QtxdfawsnE05dZX

k2NL6b7RJlcJgb4pv/uF062R7wYHPh4eGkUIOziiGBhRKAAb709JSFIo+kBX0HRjopx6eepC
+ZCbzxPPA

YC88MjIwONTfi+scGOyDB4cUCHzHxxEBEo9rjY6OIrSIHrMCgyk4ITS8e/cu8pGDMhS0
nhnR9KX1MxPe

dU/op1PTdvIR/ukKO3+PB6Av/pmIfUuo2Dc5vzI1nwAwYl8AOLu0JruUDQAj/AWAS9j8M
k4DAEwyuJHV

AACLCACLJByxlCtp4klb6ptaGlrGtvaBO3too4OSWentKuLjICJB6GJZ6HI8subenta+7vb+7
g5Fd9dA

D+nursGursHOdnlze7d4aKDd3d5E87Salvr+64d23jy8U2PPht2L31zxyp/Xznpp/ZyX57/0uzk
v/nb5

Gy+tnv3a6tmvr37rjTWEX1/15iuLX/7b/Jf+smLWK6vfeX39vFlbFr+9c9ncPSvm71mxYNvS
+YlhgW3N

AnFrgwCIJehL3HYWdBHorW/j17fV17cRsS8JYz4y+W38hlYYEbCA30xZKGgV88S84qrC
R9IXmhzwe28b

ld+qJD2vYj0+8q1i4OPBwTHAcmC4FwZBkYaH4EHFMomhwQHID+lNEXnzGejtHxtHFI
xahEdGiNvRxB/c

IH6uQXknGcWwBI0AbSpnqpA/TD7wrFx/VvTD7jzTovXT1LSbbI/2U/jJyE99/59Q017mN/
g/MjY5IT0/

IasoNrM4PgsAZiblVaQQHwBXZRRXZ5aws0rYOaU11C3oIIY9IuASdj0YXMbhM7n8Kr6
A1SBgNwprBSKO

UMwVSeokUp5Uym9qamhpFra1CVvbxG0dYsTBnd3SLpmki/g2sLSnu0ne2Sxvb+1t6+hvk
w13yEc7uwfb

2nqb+kdckVnZ2eLgFd3ar/amd37tA7v1zqw7czGZbsXvL5vyTtH1i3Zv3r+7hVzti58c/07r25
Z+M7m

+YQ3zpm9fvYbq994Ze1br62ZTQS+a+a8CfpunD9LddHsHUvmqi6es2bOrLZGfld7i7RNwm
9p4LXyBD2N

zf1SfjuvrqWO11qPZV0Ll9eKNMMyrb63ntfB4zXxecwOvSVAnFfAINza2iOpEvJLKosSMm
KBIB58gZ0c3

S2s701oui/qLSZQmxvy+N4zKCycqBv1KxGFR8OzT8+eDArUEAdrHVgn0p9JKekb7Ep7/Uk
88T9CUqkn6m

PsH9YaLpS+tno2lz+pP4vvnov62f+v4/uaa9UoIPETGJ8Wl58ZmFMRmFsZklCdllJICZKYz
K9KJ7AM4t

qwV9C8lvHyECLq3ll9TwSmt5FTwAuLG6obGmUUABuE4k5kkk9VIJv4lksLS5salF1NYha
geDu8BgSXeP

pLtLKuuUytpb5K1N3dKmHnF7f3N7X2tTl7Stp6O1q6u1o2d06L2bl3RO71Y7s23LyfXLji6ft
2f+64eW

zzu6bsnh9e/uXTF3++K3ti9+e9vCd1QXvLN13tub57y16e1Z62e9tumDNzfMeXPD3Lfwz3tr
4/y3Nsx7

Y9P8N7ctfmfjvLeXvP6aiMOTdcolze2itvb6VmmtlM9p4taI2bXSGhK9lLmkkeBxm+q5TQ1c
KdAr4oqF

dYQbG5qFXEEdRd/gKfStmYG+06XyKxXHX6ncGBn6dnDwy8GB9wcHRwHYgXv4fDx9
laLp+xDR9KX1s9FP

nV4/9f1/ck17pY+ib3I+M62wEvQlAFxUPRn+FlbVFbHqSmrqQd9iNre0tq6CV8/iN0wBsKh
OTJgnITDM

l5AAbm5ubGkVtrYL2ztEnR2irk5JN9zR1NPZ3NPRlusUtbdKOru6+od7Bm5X1kqCwtJzc9k
+rsEn9xw+

sXXT8XVL9y+cdWDRW+pr3z2y9t2DqxfX7lgz7K5O999Z8eiOTsXz9u+YM72+e+ozpm9
5e1ZW96ZBQBv

nAvugr6z1s99bf07r2yaP2vtO2/Nf/m1XRv3+HtF6+o52DgGx2cWC7tkHGIDraSW21xb18IB
bgniNnNg

gr7N9RwJ0CviSqR1kiausIprCwC4XiLgCuvKqktA35AoH9DXyd3Kys6EoO+3E3+g8GH09
XBS2NvUj41+

208883xnaGh8YGiE+IvMBE0JoNL0/TdF05fWz0Y/dXr91Pf/yTXxSr8hTdI3LDI+Njk7Np0
RmZofnQ4G

F4PB5C3oMgrARARcSDi7mIqAOYxKTmEVXFtUXVvM5pRxCQBX1tWzeHwSwEKukD
DBYJGoXiyB+WQo3AAG

t7Y2trUKOtqEnR2CjnZJd7ekq0fSJe8evFVWI7RxDIZ57kUVlZdUfvGq2lFtPW3rvVt27ly+Z
M+7c/Yt

mqW2dM6xdUuOrFt6ZN0ytbVLD65evG/5gp2L54K+uxbN27Fgzo5572yf+7YqgmAi/J21ac
HbmxfoBn1X

v/WPNW+/uvT1V2e9+PLrL70z7+31L/9rxdzFO195e+3KLfvdg0PL6thEgEuEuaSb67jNXG4
TAt+GGIFD

jVBYKyK/3yyQ1DQKaxsb6pDZwCmtLklIjwZ9vQKdHN2tLW1A3+qvv1X+qYaH0ndk9Nv
RsW9Hb70/MDRA

PEA1ND04TJj8XhH1bSLie0iTHh4it0wXVZL4O88TJiqSnrH8z0jUU1c0fWn9DPRTp9dPff
+fXOQrpdBL

mHgVoRFxMUIZMan5Ecl5UWkF0elFFIDJh7AIAKfkM9MLKMEyAmbnldfmV9QyKuGa
wqqaouqakhpOOZdb

weUBwNX8BnZDY21jI0cgqAOAhUKeSATXi0X1ElG9VMxvIvJbmvhtLfUtLfy2tsb2rubeo
WK2YPvB8yrP

//1//t/bv/7TonfePXT4tM2Jc1Znz+nsVt29Zv472xa/vefdtw8sn3tozWK19cuObFhxZP2Kw+u
WHVqz

5MDKJXuXLoJ3L16wm2Sw6rzZG0n6bl74zpbFb696+19LX//bsjf/OfelL738xxf/+eJbs2atev2
tdf94

fdXfXl/+4mtLXvh/r/zmL69Zu/gwufwybm1ds6BGzK1qrKnk11byudWCxmqBsEYoZjcSrm4
QsBv4tQIe

m19bzi6NT4sKjfbzDHBy8rAxsZKq4bK/wsh+Q36d6yHvEZVe4rlkRf+QTDEki3+VCOHtM
DyIIJcMf4dI

I+olA18i9IU+y3ufkKks+YCftWeYv6+o2PeBX1mgReunLurKfpon/17bE3padcr/N5rWKeVp
u/dYo8pP

UeQr/XrSxKsICY+NTsyMTskNT8qJTGGQAEYEXBSfXUJ+B6kc9IXTGBUAcHYJK7esO
q+cncszmOy8itY

RSx2cTW7IF1TXIPLrOVUcLiV3DpWfR2bz6tpqEeYCAxzBI1cEeE6sYArEXKlkrqm5hqR
VNjVW9/ac83I

RkXlf//3pQW/+vPcV+eqvrFg7ytz9mxXM9lzxGDX/rPbt+1as2jexoVvqS56c+eS2btXLdwP6
G5YeXjj

qsMbVqitW35o7Yp9K5bAewDgJUQovBUh77y31s+ZtWH+7LVz31g26x+LXv3rotf/Puulv7
74//74txdf

feWNd//55rI//XPh/3tp3h9fXviHl+b/6g+zfv2nN//wj7ePnNVIZMnjt7TUikS1IjGrUVBZ38DiN
7Ia

hCy+gMUXEqv1vOqGOhaPXV5TFpsaERrr5+7v4ORpY2xpwOZUf4mRjX+u7v4/PHhPKn
2DfbBiSKGgfg1w

UDEyNgyPjg3ByvQo4Y1vFhEaGRmhElNE/D3nqX5c+ekivr408Ze0lOD6CUr5Yqbo9u3bH3
zwAWLfyd/7

o0Xrp68nAtXXxG/nTM98Av8fIO2J9p/0o0v+H+zqf0ITL4qiLwmIoLCYyISMyoQc0DciBe
EvIzq9gPwM

uCghhwJwWUp+eWp+WRqDmVFIADinFAxm5ZZX5pZX5DErilhgMKukmlXGBpBqmJya
Cm51JY/N4nNYDXWI

FGsEDbXChhphQ61YQFBNIq0Vt9S3ymol3RGpBb/665vP//GN5/846/f/WDRv9cHn/jB/zc5r
G/fp7j9t

umzDnm07d69b/u76RW9tmPfKloWvb1s6Z/fqdw+AuxtXqZH0Pbh2+YE1y/avXrpv5ZI9yxft
XLpg66J3

Ni98e+OCt1e+/dri1/6+8NUX57/617mvvjK3/7wpz/87o9/ffHvr83+6+tz/vefs3/397d+89dZ/+/
v

7/z11UV//dfiP/5j/ouvqvym3+evKTvH5FaK2yvk3ZU1gsq6uoJACvpCx7zWXxcYbCZtWUx
aeFh8X7u

AXZO3taGFrosLuvLb776kvpD/Q/Br8r4eyPj742N3749duvW2Pjo2PjI2G14ePzWCDx2axQe
vzVGbgJK

RsYnhJDufpF1JzwJYGXpx0nZBMngnzSAla9nihD4fvjhh9T3fZWjTovWf17T/tb3E2paLazO
9L6dmLsJ

Y+ukJzMJE7+V+dUX8LR80jOUp/zll3fJBLqgfJ9+wIuiflVzqjAfTnY3Zedn3h9yH+7LoX4G
dOLC4jH7

88OOwn9e014UMSIBoVHh8ekRSdkhCZlhyQBwbmQqGJwfnc6IIwGclFuamFOcnFtCArg8
o7AisxgMhpk5

ZXBZXgWTUcksqKwoZVeX1rAicyrLuCwmj11Zz6nkAVT1rMaGaoGgRihiC6U1ola2sLN
OqmjqeX/F5sMq

//PSL/88+/k/vPXawi0qv5v1z3k7Fm08t3z79S2H9eau3Km6+9D61SvWLkYU+/LGea9sXvz2
9hWL9q1f

cXDjSjV4w4qD65bvW710z8rFu1cs2rVs4Y6lC7YsmrNx0TvrFsx+982X5/zzz3P+9ec5r/xl9it
/+ftf

f/f7//317/70h7+8+tfXp31u3+88au/vvaL/335hf/3LwD4768vfWX22ldmr5+zeMeLryz/8z8Xz1
um

amTtjkuNynohs64BSxZfXFkvQgIArqyvLecwo9MiQuP93f4/e2cBX9WR9mHquLTFtVDfCq
Xubttut7vb

bvt1uxUqW91ii5SiRYsTgltwAhQtWrR4hBAhRAgRIercje85804ON0qgyW0L8/xeDnPmzJ
mZM3PP+5/3

5tx7Jw76fkz/rt/+d+svW3MKuHN4lfD6swJgNebFXgw1UtJOp6QmJackK0tKSj59OjE2wbI4
LD4xHktI

TDidlIglJifFE8qVAdko+Fk7fTo+MTEhKSIRHUo4fS6QKEii/eTkVEXaHxN9PQ5wReSjvjk
5OdyKBoMT

QHlcf8J52LBhEydOPHjwYFZWVnp6Wk5OlvXr0aqYvDilmKwRQXalEk7hpZuenk6CzNx
cS1CVZefkZKRn

JGdlp0+b7hoZddz7klecmJvLIaqytpRUWqU7g7hyYn4+Zu2RphiJ9PTUrOwMcrOVwuSwX
bFi+ahRo6RX

QH/UidaaQNIk9u3bx9WRxm+og/lclPQWJME2IyMjLS2VC1eXLzdjbl5+FsaFyJYcq7DqiaR
tpG9yRRiJ

rGzGLY9RHT16NAWony453uaPP/447dq9AnbpsAz44sWLJ0yYcPz4cTnkXBg6rk5bXr71a
hjnOn3KnEVT

5rlPmrtk8rziU+evmLZgxYxFP85cvHLWklWzl66es3S127K185avW6A0eNGqjYtXb1y6bpP
7Txvd129Y

vgFb/+Pmjat+3rxm+8/rdm5dt+vn9bu3/fTLtvW7d2z4ZdfmPdYHk7bs89jnG7h5j9du76Nuy3/
+vPvw

1/7Z9YnnP7y2eafa195+Vf2brm506633/6VG/dta3/Hqnx7/oONzX9z9zCfNb33m4af/9sxTTz1
0F+rb

6pFbWz5yW/sn7rjt+fvueumBu199uNNrj9yL6D57921P3XEz9sxdtz19522P3X7TA7fc0LFD
q9taN765

5bW3tb3+5jbXtWvVqFGja66pc2Xtaxs0atWqfqs bajdrd/V1bS+r2+Lyus3rXN++QbPbrm1xx3
Ut727c

+t6mbR9sccNDLTs80PrmB+o1ufHR5//Wvf/wfsMnfDdswswla3/ed3jzXg8u7aedW+b9OH/C
rPEjXIYO

+KH/1z2/2rLr5+x8Xtm5ObxC1U+2F5l+AUONtLQkZSIFlpyUnIAIIsPJiadT1DY5Cd1NSkn
GRCeRSfW/

IxxJsA0Vx5BzDlC4MqBSf3TpBX0xDpApnsuor8HJ8KpDa5988sn4+Dgc7pEj/vqA5XPFC5
CwETHLEaFS

aQt1SKMOpUkarXr88YctXc5IPnkywm3eLBbuaBKWnp4i/11JKSUzM7PSlZ5RZ1ZGZgpnk
U0Z1JfCZKpd

IY+zKC+qjFLm5IhUW8hqQBI4Cumhl5fX3LlzkTRyRNhkK/IMjrcekik1q0azs7LTMLVLvt
V5JavoqLU

sPtPATmdWpXlqnFgBZDr6ekpX5rr2Bxpntn//+9/pkuTIIZA/QIHAz8+PROfOnfUBp2Jdgm1I
Av0b4zrN

dc7CyW5LJs5Z7OrmPmXesmnzl01fsHzGohWzFv84Z+kq1Beb674GAS7S4PWL16xfsnb90
nUblABvXL5x

04rNm5dt2rR2547N+/dt3Lt37c5dG/fu27B7/5a9h3YfClqz7eDEWe5T3FZ/02f0zXe/1PGhfz7
6TOfb

7nqtTsPbr6jd/sq6N9VtclerW5+5pvE9re947fbHP7z7+S/vfOaz5re9dOcDf372meeeuP+OR29v
/fAt

ze+/sfUDN93wsPVZ3puevvu2Zzve9uQdNz98S7sHyL+x9UM3t3vgpnb33NDyT62b3tLiupu
aN7ql1XW3

tWtyyw1N2rRsWL/hVVfVuaL2tQ0btWzdoFX7us071Lz+hsvrtbq8bouajdrVbXxTvca31Gt8
W/0mf2rQ

9K6Gze6+tsWd17XE/tSo5W11m97Y8uZ7G7a8rXG7O5vccFe3/sMHjZmwac+e6YvmT5w9
1dVt2sAfvv+s

61c//7IjK59IImapr23F1JdXbJGxBMbS2WZw+2RmpGdlpGWlYyTIFUvjRacsXVkJJO0jxS
wzM4M74VIm

O5sIwLr3Cgocv3jMYKhG1N2dR2LMmFFHAgNIZOdkFBTkREdHdv7o31OmTejW/avMrJ
Qbb2o3ceKYd955

08V13OQpk/71r//DFw8Z8r2rq+s///nPyMhIeTiTRK9evfr27btz5849e375tm/vr77+/FhY0PWN
G8ya

PTX02NHRy0Z8/d/PScTFn3r3329PnDj26/9+ceZMYXZ25pkzBYMGD5g+fermn/8n7Hho4
NGABx+813Xy

hJiYE399/RXavfyKGpQZM3bk7DkzvnmS7r66GMPDR06+KWXX0g4Hbd23ZoFCxZs2bJ
1lqxZ11xzTVRU

1Pjx4+neJ598Qsd69OhBZNY9e/eVK1cmJCSgZC4uLl9+aVXy2WefTZ8+ffXq1XIJJ06ceO+
992bMmP7N

N1/jxj777ONZs6ctWDgnNy+D0TpwcE/vPj0+/c9H9HmS63g3t1lffvVZTm7Wu+++Q1Xe3p
7z57sdOLiv

X7++X371eW5edkFBHtvuPbpNmuTy9ddfL1myhCEaPHgWHQgNDY2JiXnttdemTZvWtm1
bTh85cuTkyZPf

fPNN+rBnzx6GkQ6ju/SKdEREBGWcTrFvmiw4U8hrZczk6a5zF02ev3TSvKWT57tPW4juL
p+xcPnMxStm

Lflxrvuque5W7IvNX/ETHvouWrVhyRrC343uP21atn7z8g1blm/4efmmbau37l67Y9+qrXvWb
D+wetuB

AaOn9P9h6pBx897u3PvBZ96694k3W930RJO2jzZv/8wtd71+x71vNG39SM0GN15Ws+VV
dTtc3/qBOK3u

ubbt463v/Oufnujc8YWv7nr28+a3/7ntbY+/9NKfX3ri/sf+1Orhm5ve36HIPW1bdGzT/N72rR6
6pe3D

t7Z78OY2ZN7XvkWnG1qQeXfb5re3aoz03tgM6b3+9nZN/9Sh+U3tGrdq2aBeg6uuqn1lbW
Lglu0atLyx

Xoub6zS96aoGbWvUanZlvVbXNGyn7AasZqMblXW4un67Ooh4h3uva33nta3vbNyuY7MO
aPDtTTvc3fZP

9zds2eFv737U+atuQ8ZN6Dt0+Jfde/38y76cM2dyeW1Z3+LJCocrK/bdv2Wqb7qD+lqWnq
Wl11F9sVLq

m1rKjPoa9TX8BuDc2SKlffr0QieIvtLSk8nu2u3LpOSY3Py0yVPHxcZFPvRwJ0r5+nlOnjJR
bUZakpMT

u3btIjX861//IoFIoGcTJ05EY15++WU3tzlDhgZKzcvKyc146+2/F57ByZzZsGHt6NEjEN0e/+
uamBiH

Qx/8/YDo6BMkoqNPdu/RITLw/gf/PuTj5Tp5IumePbudOInwFLz9f29m56Tn5GbOmj39ttuI
hJ9/W9/

oc9INQuCHTu2r/tpHeWPHTs2c+ZMEsnJyZMmTbrjjtIIGPkHDp0aPHixShxYmIivUWPg4
ODn3nmmfDw

cAJIS3sLC7/66iuOUnjEyOGhx4KfefbJsOPBObnp+QXEtVko8bDhgznK6uGOO29dsHDu+x
+86+Gx/+v/

WkJOn9euWz1v3tyxY0cT/DOYhYX5iYkJvXv3ZHh8fX0RV27wTZs20TSLg/79+9M37vq3
3norLi7um2++

oRL46KOPFi5cSBnSeAOCZjc3NznkdM6qb2GR+o52neYyZ+GkecS+i1zdlkyZt3Ta/KXTF7
jPWLhs1qLl

s5esmLN0pQgwga+D+m5SArx5xcbtyzdsW7xmy5K12zfsOrxi44HRU5Y9+uK7dz781wef+7
/b73/1tntf

6/jIP+948O9N2z9e5/pOrW58rnHrp9rc/HK7W168ss6NV9ZpeXnt5lfVa9+iw2M1at3U/Obn2t
71uhX7

Pvcl6tvijleatLvvlT+//+pfnHnmq4w2P3NLs/g4t7m7dDOvYppmSW0t072nXDLurTdm72zRF
em9ufu1N

za+9scW1t7Rucnu7Znfc2PKmGxq3bF6/Xv2rrq59Va2GjRo0a9ug+Y0NWtzaoMXtNa/rUKN
WcwT4strN

r6zXUImrK+tira9u0LZmQ9S3Q4Nmt9ZrenPN69rXaXzjda3vwJp36NS0/T3N2neynp1ufmuj
Fjc++dLr

//roq/Vbd9vqm6+lt5T6ZmWmnbUinU3PSEtDULOzsvNyE1OSU9FibrWs7PTMLNHd1PQ0
LA2zSU9NS0/h

Di9uKcU0+pKE+x8BNuprcCa2+i5ZsmjhovkqLw+Z+eLLj1NS4woKM8ZPHJmYFP3Sy8+k
ZZw+5HNw+gxX

luA9e3ZHVHr2/B8vV16077//vkgX+pGUIS/gKOih5//BEcN8Fr4Znc7j26+Poeiow8PmPG
VMLi+IRY

yowePTIo6AhlRH1RU9Iffvg+sa+b22zCys8++4RtQUHuiy89z6r++eefob/vvPMWTuPjTzoT
XEZEHJ82

bcquX3auXPkjl/PKK69QLXcTKwASrAxQ02HDhpH28PCYP38+0TmdZBd5I6Ckt7NnzyYt
l/DFf1/Ex8cX

FOSPGz82PPwYxebNnz102EBWDwwLfUtJPf3Ms09EnTj+bd+euXk4vEz6/Pb//ROh3bFz27
LIS+nSyVNR

Tz/zJALM6TExpwYNHkiF+/fvX7p06cCBA0NCQlgiEHAT0eIR6fPzzz9Po1276sXHG2+8Q
f9jY2MffPBB

WZFTThr79Fs6hDPUDnWnqxNnzJ81dPGH2QraT3ZZMnbdkOgI8f+nMhcvmLF4xZ8nKOU
uJgK13nm31Xbx6

44KV6xeu2rBsw/YFqzYvWLnTx01e/+09/u6H37zzgTdvuecft9zzxi2d/tb+zldv7vi3ux/5v2Ydn
qrf

/L6Gze9v2eGplh2ebX7DUw2b3lfjmtZX1G56Rd1mNa5o+tgL7//9373rtXiofad//OnxD+985v
NbHv2g

TqvHLq/X7plnnv+sU7PduqA+t7XvnmnG1oS+97dumnHtk3vbtPkzIzN7mjZ+E8tG9/Ruslt
La9Xbzhf

d2Pz625q2fjmVo1va9P0tnZNb2x7fYtm9VDfmmWuqVW/UYMmbRo069Cg+c2ob90mt1xet
yXqW6NW08vr

IsAtsCvqNL+yTour6ra8mpi4QauaDVuh1g1b3FS/WYd6TTrUb3pz/aa3Nmh6Z4Omdzdqes+
1ze5u0vqu

FjfcVadR65XrtirpxQqK1Nca5+Lqa4W4xUxiX9SXFNkr5DYzMTk1Jy8f9UWJRXrLV98kB
0s26mvU1/Cb

gH9ni4cfOHDA4O8HjRg5bPacGckpCR9/8v606RP79vsfR2++pX1BYbaHx97xE8YQ0nXu/
EFOTvbQoUPG

jRv3ySefICfIAy9dBOPVV18lwlu//qexY8e4urp8883XCOfwEUOwlSuX9+vXt2vXb1xcJqSk
JBHgjhs/

pmfPHIYf1APMQ4YM5hTiyLi4GOLiRPHI2yhocFvvPF39PXGG9tT7JFHHyKyrFnrasr/4x
9/y87OpMCk

SRO3bft5+fLlaC0rgBkzZoSGhr700kuurq61atVCtzp37kzm559/Pm/evISEBPXe8owhQ4Zw7
ZxCYv36

9dx3XEVkZOQ777wzaZJL3+++pXVC2yFDBq5Z+yMRP1I0yXU80f9XX3+WkZGM+rq4j
CM0LyzMI7rlWj79

z8ebNm/g6ujP1199zjWqgc1hMTFlyhQi2iVLlqxatapLly5ff/01ek9br732GsPVsGFDmqYbjOe
XX37J

moCeT5w4kWIpKSlcC11ieNXy4CxUXv2Uob4jJriOn+E2cfaCcTPmT5y1cNLsRZPnLJo6d9
E0tyXT5y2d

ZUXAK2Yt+nH2YiLgNUUR8PoFP26Yu2ztbPe11geC1+xY87PPvz8diO7e/3jnux98/5aO77S
/4402t77a

8uaX2tz65xvueKVOk07XXPunes3ubnrDw61vfrpxm4drNrzt8jptr6jTDPV99Lm33/3PgIYtH2
h+87O3

PvT2nU9+9KcnP+3wwL+ubvLAVQ06dOp03/OP3fvMPe2fuL3VA9abzK3uaUu826KY+rZq
TNR7a8vrb25x

/S2tmijpbYld2rbpLW2b3NC6UdMmtevXv+qaWlfXrNew3vWt6jdpX6/JjfWa3lKv6a01G91
wed0W19Vp

ZqvVXWbO1izq+o1vbJek2satah1Xeta17WpdV3bWte2r9XoltqNbqvT8E91Gt7WsMlt17e4
7cpaTRe5

ryuuvgxyJdSXIV0mlm1FwXkFBZncjrl5+YVnUtNR5iwyjfqeF5mZmQgwLsBZ95XBYL1dr
FMK9TdgySIE

3rJzrD92kkCbVSSnCxcWWkllsg28aAU7BzhFyiOZxNOSSfwmQmslrd+KpYDy7FZaKJQC
8rYt+6zy2ZL5

yivEslavEEWOqii5WOcFxz4Ico10WHalq5IGyaeMYybQEHJc1PM8YncxtSuCxCmWJLNPF
6xc67fnrT4X

XU6hGjRdbYn67XalwyRAHSmG6qx1Ikdlas4lQQpUM7QiF3sO9Z0yZ6EYGoZnmL9sxxvl
aPCsRSvnLF49

Z8kalHiO+1q35etnu29Yu+3Q6+90a3nTczff/eat9/zrlo7v3nD7my06vNr0huea34g92/iGx2rUv
uGK

Bh1qN765Uas72972RO3r/3RZnfaX17nh8jqtsBq12zRocc+fHnr9zkff6vjkvzs+/dFtj3dudfcbd
Vs9

Xq/Jnc2atHzo7lufuLPd03e0fahDqwfat0GA7+vQ+r4OreSd547tmmF3tWt2R9umtxPstml2a5v
mt7Zt

fmu7Fje1adqhzfWtW9S7rtFVtWrVuOqaK5T6tq7XuH3t6zvUaXxTw5Z/Quavqt9S1FcJcPOr
MERXrJ5W

3yvrN7mqQdOrGjSzrH6LqXu0ubp+u2vqt69Zr33d625s2OTmy66+ft7iNQwpLxf1zjMjLFZsd
kupr3rn

OSMzIzY+zsV10uSpUya4uIwdP37KtOlEwEyU/bZzMfVV0suRIvVNTk2ztiqTM6gTGUKG
y1Di1NRUnbpI

MeprcD7yYmOLZ5ccVIG0cvRWjnz60HL2Sp9EdEIL4RLKwQvYQeR0hZzEPS4eHDUiWs
WUdlrQcm6eVJJv

uZSMNMIXCm2pZpq1qwjHP/rowwMH91FKjnJIKZwWBkz6qQ8q6BK3lXQMbLVT12YtF
KTzRZmisjIa1id9

1S7pvJxc6+9pWG5eVnYOSwEK03kKSM00KgnyZcWgd5VRlXyOWQtnDNA2jDAlyAFJqF
OsAZQeUkZ6bs1E

UWFVxOqtLlvVjHUIYvLjvlz2iPGTxk2fO2HmgrHT502cuWDSrAWTZ1sm6ksQrOLgpdZ
ngue5o8EzF/w4

a+Gq2YtXz1q8dsHKrXPcf371ra/vf/Kd5h2ev+FPf2t1819bdPhL8/avNG33QtMbnm3W4emm
7Z+o3/ye

Gte0vKx+66uvbduw1W3Xt+t4ZYP2NWq1uaxOu8vqtLm8Tuv6ze6o3/zumo3vaH7rU/c99/6
9z3W+9dF/

t7jzr01uev7aFp3q1r2+XdPG97Rv9sTtbR+7pd1DHdo+eGPbB29q++DNbZQAAt1R/AG7ZsV3
zu29ocWe7

Fne0a/Gndi1va9filrbNO7Ru0rZlo5bN6jRscHnNmpb6XIOnQd1rW9W5/gZRX8Lf+k1vqtmw
9eW1m2JX

aPUV0bUN9dUCrKyxla7XVII0y6vrtqzVoG2969rXuLzhnPk/MqRMMFYkvRiza5m8impwY
zgYUqFt5A8j

I6OisnNzEE/m6v0PPujWrTuvMus2Et0tKb0pKG5WdlpySgIJVDU29pS6G3OSkxMpEx8fl5
h4muK8EBEk

7h/SbFHfZPVVUCR4aXKU1zE5cvQiwHryKjPTifeVwaAp/ZKT16H6ml9LM8DOtIXBEclR
xx0PsSffckxQ

iJPRflwZxcR0WmmnnSmUWd7RwDF9FumJoLPOgRSTtgTJYWtlorgsI+RCii6nyGeWa1Y
N9jLinOjuKnSW

dboFCcdMJ6IvHxP15YKHj3UZN3XO+BnzUd8JM+ZNmjV/8qz5U4ps6pwFmETAU+cunjp
3yVS3ZTMXrZ6l

aN1c9y0Tpq+888G/3n7fXzvc+UrDFo+2ve2V69s+fV3rp65t9STb69o80bjtY41a3ndVw5tq1
Gxxef1W

Na9v26DlzbWa3FijTssatVtdVq/d5fXbXdWofc3rbqzT5LYaVze/4rpbbz/z4/95ZO7n37vhnv/
luzm

5xs07Vi7dtPr6ze6uVnj+zu0fvjmGx5EfTu0feDGNvd3aKUeuWresU0z673otgTBLe9u1+ru9q
3vvKHV

be1a3tSmWftW17VuVq/p9TUbNriidu3Lr6l1Nepbp1GL2te2q3N9BwS4buMb6ze5sXajNlfU
bnpZrSZX

1rHfbS5haC3W5Kp6jZVdf3W9xpbVbXJ13aY167Wo07BNjRp1Z7q5yytJjbKtvpYVFllp9d
W4uLggh1aA

q+R206ZN3377raurK3JiKa4j6alZWeksHhMTYwvP5MrTCjmWjGakpCRxi+7Z80tcXAz1Jy
cnUTMzz0II

LCWWLVvGAtBqKDWWVHHd3d9atpCmQkpLCVgTsD42MZ4l7z2BwAqVfcvI6rBhdVKG
zFDrLgj1RX8tsP66M

YmJWuvhZjtjFyjQpUAZWjUXorIqgTJl9s41My0mWcy3lmZwodm50dxU6y4EyM6sfx0uwe
mCp75iJY6fM

Hjd93phpbuOnzZ00w811ptvkmW5TZs2bMnve1DnzpsyZN9VtwRTLfk6avXDirEXT5q9et
HLXSJdlnR59

+5aOf7vjgbcJc1HfVje/0LDlQ5a1sKxB8wfrN7vmoa31ri6RY0rG19Rr2WdJjfUa9bhyoatatR
pXqN2

c6Lhyxu0vqpRW+tx4qY31bi6MZk1rmlSo37b2x/72+2P/aNB6wdqXntr/WvbN7muTavrmt3
YpFmnG254

8KYOD3Zod/8Nre5t1+KeNs06tm56d8um97RqZllrNLjVPTe0vqtd69vbtrypVdP2La9t1bRu0
+tqXtw

6np1r6pVpxbqW7th81qN2tS67oY617cnCK7fpEOthq0ur9X4sprXX1nn7LvNZZIW36st9b3+
GrG6jWvW

a16Xi6pRe6bb0vLU17Zy1XfatGmILrEvuSySURGWxjNnzmREtRhWMCzvOefkZjRu0mj8h
DHvvf+u6K68

HzVs2JCgoEB25R2kkSNHhoWFobJ+fn6JiYloLTXLm0Uk0GD14jhDH4O9RXkogyG3xb
RgIrRRRU6S6Gz

LNgrT7EoJmali5/liF2sTJMCZWDVWITOqgjKINk328i8APV1tHOju6vQWb89culiVrdKq6
8L6jvLtrmT

57pNdpvr6jZ7ktsc1/nzJs1bNHPp2jnLt450XdGk3VPtbn29zU1/a9L6pXqNn6jX5JHm7Z+p1/
Te+k3v

qd+0Y4Om99RrfE+d6+68sk77Glc0rXF1s9rXtm/Y7OY617e7vE7TGrUR2saX12t6ZYPMVz
dqWeu6NnWb

3lDj6oY1al1Xo/b1Neo2qdGgecP2dzW79cHmNz/Y+saHOtz0cNuWdzZr2KZDkzYd23W4v/
0N97VtdV+b

Fve2btapZbN7WjRli93bqsV9bVrd17b1PW1b39Gqxa0tm97QtFHrxvWaXVv72gY169W9pla
d2jXrNazd

sGmtalZD09d17bO9W3rNW5bq2GLy2tdd1nNa6+ofT0qe3V91BehLc9U1HtWfa+vWa9p3
UYtalxWc+a8

xQ7qKx/5LWmO6mu94SzSC6gvujhsxPCBgwf17t17yJAhBKNjx44tT33VZ4VT+3z7P+ayx/
+6paWndOve

ZdDgAfsP7B0/YWxExPERI4cTPRPs/uUvfxx8eHBMTMwnn3zy7rvvBgUFTZo06euvvz5w4
ADbpKSkN954

45133jly5Ihu4aJAv+YNht8ULQIVoosqdJZCZynUG9dlKpaNVb7EWb8e6YmgsyqCMo59Y
7eEkWn1v5xr

qYydG91dhc767ZFrF7O6xZUPGz1hzORZY6e6jZkyd/zUOS7TCX/nTpo5F/WdpNTX1W0u
0jtp3txJ8xZM

mrd0ktvK3kNnNL/pubA3v96qw98aNX2xccuX6zR6pHaje5u2ebTOdX+qe/1t9a67td51t9W99
vY6DW+t

Wa/9ITVb1q7f7vrmt13b9KZa9VtcWfv6K2pde3ntRlfWve7qBo1rNmpW+7oW9Zq2qXF13c
vqXnt5g+tr

1K5f48qa1/btG7L9o3b39mi/f1tbnioXdsHWja5vXmDVjc3a3tXm7bo671tWt3bukWnls3vaY
H0Nsfu

a9Xyvtat72/bplObNne0anlz82btmlzb8rr6zRrVva5BrXp1a9auU7tW3Qa1GjSu1ahZrWtb1r6u
VZ3r

W9Vr3Jrdy2o2rHFN/SvqXGvJan0d3VZg19S7Tlvd62rVa1y3UbMal109a94ixxeT+nPv2fecx
VBfEd1i

OgsuLi5M0eTJk0f+MHLgwIGIiplIMuorBUDJyln1zcpKf/Sxh7755svVq3+kxXvvu2fevLk9e/
aYOXN6

dPTJ9evXubhM7Nu379y5c+Pi4oYOHSoFzuvfvz+tREZGJicn9+zZc+LEiVRaUFDQvXt3Vf9
FgnrFGwwX

EyJgJex3SIkeljbbSSo/WbY5lnG0PyiOV2SpL/+Nd50+1nXWGNfZGIlxrjPGT545fsqM8VnN
jps2c/z0

WeOmz5owa+7U+e5jps2fNHfVmKkrb+30eoc7/9721r83bfNkk1YvX9/imetbPNaoSaf6191
Wp9GN9a7t

UN+yG+s1vKIWnXZX12xRu06rJk1vadr85roNWlx+Zf3Lr6p3xdX1Lr+m7mXX1LmiZr1aD
a6/tnlrrGbD

6y+rVa/GNbVr1KpT44ora1xTk8QV9a5Fs9vf+MBttzx2U7t72zRp37Zxq1tatbmzXbu727Xpq
MJcrBPW

ps19bdthndq1u6tNm1tatGzfrGmr669t2rDutfVqNaxbq2H9OvWhwbV1G15fu2GTa+pff2W9
RpfXrn9F

7fpX1mlwVd2GV9drVLP+tdg1JCpjobXVbdDksitq16hx1crVP50dX2WIX0Wob9lMnz6drau
r66hRowh8

d+7cmZ2dTVqOCkpWRH3TkpJOZ+dkfPTx+7Tx7/f+1XA67q2332QtmZ+fM278mO3bt06e
7JqcnPT5558T

6aK1S5cu3bJIC4HvrFmzUN/Q0NCcnByC7FWrVm3fvj0gIED+xnzRoF7xBoPB+Yj3q8AcfS
K7pSIRxtH+

oDhekVbf70eMGTxi3JBRE4eOnjR8jMuIsZNGT5xsmcsUy1ynjpk8ffz0OS6zFrrOce89aIL6l
olODZs9

VKPm7TUuu6nGZR1q1Ghd47IWNS5rXOOyRjUuq1/jsno1amD1a9RooIx0HWXo0zU1alxe
w+KyGjWuUOnL

a1x+1WVX16xx5dUqX3GZKsP2cspg11x99bV1ajWrc831V9WoVfOyq+tdXbNhZqNal597
TVXXXfNVdcr

a3z11U2vuabJNTUub16x57TXX1Lv66ppXWodLS1dcZpmC/zDVNE1IK7ah+thlV1TOKKns
cuu6Lr/ymrnz

Fp4d3zLG3LJy1ZcYNCEhwf5DLCoaGxtLJrGvjSpIxGy9cZ2Tm5WWnrJ+/Zqs7LTY2FNbft
7k5eUxZMjg

Awf3+fkdRoxdXSe5uc3dtm1bdHT0mDFjqJMgGN0lsXv3bionsW6d9ZVy5KPQJLRwXRR
wOQaD4bcAr1ex

OfpEdktTooyj/UFxvCKtvvs8D+31PLzX03e3x+E9Hod2H/Ta6+G996A3+ZZ5+WD7Dx328
A3w9Du64qef

3RavXrRi09zF62cvWDdnwU9uC3+aM3+128LVC8lfunrhkpWLFv+ILWa78MclS1YtX7bGf
enKhQvdF8xf

umjRMrbz5y9euHDJoiXuti1YvHT+oiXzFi7GSC9csmzugsWLl65wX7Zy0SL3+fMWLlq4e
Onipe5L3JeS

WLjIfRHbhcsWWbb8rC34cfHCFYsXLI+yaNmSRUuXLI60eOG8eW4I0KKFC5YsXrxkCe0
unL9woduChXMX

LZ67eMl896ULlrkvWr5sobs7tmDpUrH5S5ZUwihmmXSeq5g6fYa3z2FGWY+vDHmpV1G
56oscjlOMHz8e

LYShQ4empKR04VWogtZb1sS+SC+Wn5+tPulLsYyCggy2mPq6jpx868c0rJ93IMZVJ1iPPa
Pu9qNV1CV1

kk8xKSOHLgL0+Bsmv1cK1Qd/9U4Rv6c/VV4wXELFZjtE0uXhWExKWjnyeV9d5A+GfU
XWHOerLxzJVl/b

jckTQyWgGpmQkZcnZ2blWdsc/ikoYOOYLr6joLlc6/dv8/IK8/ILc/MLqVGsQH0eVhWxLD
unQL8Gz1ZC

10p0UMraJr0T04dpKjs3x/qJIfX5bOufOkwt8gdaOVNVXZhbWCCWU5BXCbM+++o1JJT15
Mki6QrGiDOmS

tnLV93xAXNLtL9xQzz9bSqyMdnk/FyiaJFAFukumVGejD18U6PE3GH6vsOS1hUREI91fr
76ssH9rCaf1

Cuz8kL+myfNZ6ssz9Dds/DE5OwL8h2ggwGzFuCo5XEwxiszK1xp21qSqSmKdoJ4GJIHsY
aTipooWmfVP

6aN19KICKKk76/CYMU2pahRSgurEdGmVb1t5jys7mL4IMU6xcUwXYVdsWZWOL6AvV
vjL1o6DtfqWg2iS

oOsohT58UaAH32D4/WFFcOrbhmWXBDncgL9GNaVOSVxM6ssF5eZlZ2VlsJXvuiKLK0
WGuVIp8UeEgeBK

RHTFRIfAliXJUZIsmFZ5StzR5IxKwauC18U5TaMbttqtdFtSGAUnsLfVvFiV9nVh7Io5Qvl
ScluRFa+g

VGXFrSrV1zajvqXR428w/F5BPKxQTgVzliR4x1+pKL+PuLCk1ytu54vjKdbg/D6u8dfCVal
vJD6rRphc

quOuw8Vbrw6sKlv4wUqg1Pfc6NIXAudKxx0v62yF9uFiucCOmC5zNrQ9pxWdXxqpsZhV
k/qqN5zFykE0

SdB11ElfvijQU2Aw/I4JDAwcPXr0vn379u/fn5CQsGnTJn0AV1GWH6zYOXK6h4eHYyW/
EWf9XVmG760k

FLa23t6eqWnJpDIz05EfeS51+/bt6ugfmKJ3ks8OjVA6R7C08Veoo5x+TnTpC4FzbXm17WY
FpGwruW9b

qYwKjc15nFFV6guIqDb1MSRH0we6gggtCV/G7R82KwfD7Zffu3WPHjsXfnThxYvLkyTEx
MWTa7k+ekSTB

1g6IHRPKVerCkJOT8/bbb5NDaCj5Kq62zpXCUgwcY0fJp62srCwSdrXUxs1una9+uYEj8iy
nSIM+iwKS

pja7Qikvf6ZVX8ssepnfufMHuUW/p0SBgoKi39nNy8+3HieiG9bpxDL2LhVZxfKsU0aNGh
kbG616l0e7

nEuZzz//3DpHdV4Sht8aZrGESWZZOJYqYZXmN1Tfs2jBqTT6tAtCV/G7R82KwfA7BQk
ZPHgwCiS748eP

T0507t27t7+/f9euXe+9996kpKTu3bsjbF9//bWPj8/IkSPlc4OIzdVXXz1o0CB3d/f4+Php06a
Rjy75

+flx1rFjx6iE8r169erYsWNKSsoXX3zBWV26dJk9ezaZERERU6ZMkU8YwocffjhkyJD+/fu
Tk5iYyCFC

8L59+w4dOjQ4OBhtnDzZtWvXb3x9fZBAF5cJ06ZN6d2751//+hphW7169QICAr799ltqOH
LkSOfOnbf8

vMXT8+DQod9Pnz51yITXb/v2fu21V/38Dnfq1DEoKHDlyhXvvf9uamoKXoTLoT+pqWm
DBn3fq2ef+PjT

H37wUVZmTpdvus2ePXfGjJnx8Qlc6YGD+/v06fXIow8lnI5buGj+W2//k3H77LPPuGQ66ev
ryxB9+eWX

aWlpXMiaNWsYB3spYHA6xaROmWSWwrFIBXZupJDjORWZUV8noWbFYPj9IoorawQ
vNjYWGZPvvJs5cyZa

iI6iJQMGDBg1atS8efNmzZolESr5bPv167dz507yOQvVQZbQIeSc8hMnTuREDiFLFKNwj
x492KW5VatW

zZ07d8SIEZREvJcsWUI0iZItWLCaHDC3t/Dw8L/+9a+0FRQUhJaNnzAuLT2lf//vFiyct+6n
NZOnTMrJ

zVq+Ypmv72E0m5pff/31xYsXBwYG0vOtW38ePfoHN7c5y1e4z5gxLS4uZvYcLiQBGcb3o
d8fffQhUSzK

jXDKN+699trCxcujomO69GjJ7Fv167dQ0PDRo4cFRMTQ8eGDRtC9Lx06WLOGjiw/2eff
RoTE02faZe1

C0sEEps3b/b09Bw7duyOHTsYB3IMvxEl1a4osxSORSqwcyOFHM+pyIz6Ogk1KwbD7xek
9J133kHniGJR

4tTUVCI5b29vosnHH3+cGPS9994jyOvTp8+BAwc+/vhjgkU5kTRbwlnoQnXQIbRW1Bf5+
fTTTz08PKiK

SgigP/nkEwJl0l5XV1f56vgZM2b07NkTrYW///3vNIG6e3l5odCdOnUi4vz0P5+6uc2lJ6jvyB
9GIJ8e

Hgc+/c/HK35chrKixJ07f0AHKMb273//G5qN8rFiQNR9Dh9CYt3dl6C+SUmneW+2zz33zC
Ef7/c/eO/V

V/8cHX2SkJp1wzfffEP33n333zNnzK48ndT5w4/GjBIHHLz7lz1ff/0NMTqV7927h1C7U6d7
YmJOvf1/

b73yystewrvvvtDLEGIzhmEr7/+mpHkumQVwlmG3w7RuUpwVhPLsWqgWtT316DF6qJ
Dj7fB8AfBfssU

UZw0aZJ8850NR0VaOCrFHCmtOuS8/fbbdmxdHhJMI4RKiy3YVQ8D6V9BcDRqHTd+jPx
6KXKvesrE9qBl

WEZGurQCdFLSqaIpkmPDWkSn1Fk4KpWw3rFX3awIOc3gdBh5XoeYNWUyX07Bbq4c4
yWhzKivk9DzYjD8

oUBd2Pr4+HBvIk6iTzZSpjQUYytSmpOTExAQQETr6+srsG1lnittAbG11K8oW3ox2vH29sz
JpRXxa5ar

VfnWaVLVhSE9rwyqh+dAFzU4G/2SUInfE3RHmVffJ6HH3WD444ByiI6Wp5floUTnbHl
OZytVWQJeKjKW

wmw5ZLdlVWFRrvrm5+dkZWWQENHF1Ndf5KITzqO3pbHXAYY/MrwG/vjqqwVEobOq
Dd3MRYced4PhjwZK

dr7qWwIJeS0lLarEMWFDGUfZ07nlqy8mclvCfr36/srTDb8bmMff5VSqflWkviIbklDfkGEh
mYIUq1p0

1RcdetANhksYR1WzhFVh70rg63hIUZH6YsV118qxHJvB8LunbPXNViQrSGRIZaWmFvv
WKIEUXbpKkZov

PvR4GwyGstBSWwbnUF/bjPoa/liUrb7ILYjQShr9kF1BFEWXrlKk5osPPd4GQ5Wi3sEqw/6
IaLUtSTH1

LfpLnrZyDv1BB8BwaVG2+qKvKSkpUoLwNykipiUwkxFF6QQqD5Bml632VIwmUWxKO
lM7UNV68yGAaDFXI

JSIyCDBbCW0dJNYyMIURx0xrVBBtlW/4YyDzpRZbFpJ50VOu+ubm5h46dKhu3bo9e/bs2
7fvNddcc+TI

Ea0kC11UQfnU1FTUF51GrVFuyMnJQWXJpCq2nCKSTBnyOSpnkSMVXtzo8TY4BX0Tq
z8lllocUyFforApx

Tnl9AZXm7N88yzHalual/vPtjzPLS7oE6i9fmVypeiBLxsdWWevH/IQO/K5dtlws6H2DA/bIS
AIk/6Kn

3KeuTp482axZM0T02LFjISEh3A Y33HBDdHS0alkuVIS8NY3uMnCUFN11Sw5H9+7dGx8
fj9aySz4FJJ4W

baYAh2yk/osPPd4Gp4CnZswTEExNjYmKiLjoiT0RFVNoo/EchsizCw8OZRNFmPbsO6ssCg
wRbiY/B0Xdz

CiWUPz8POAGTFQE1VECe/Lx+nrKykGLSB92nS4Pzul4pzEDJ7qVDueo7YsQIb29vFHT
YsGG+vr4kPDw8

Ro8ejVPTJYqDcCLV3bp1GzNmzIIFCyTkJZ/yo0aNQsLRWnZpkqo4xCuShFrYWvkivWAp
1cWIDLfBOchr

jxVeXFwc68iLilMnTxRZFBZdjkkU45Tz5tece6GcKAsEODY2ljnFY8jkKt09axn6h02t31ITh
4NLoTBb

TqRatjHKKoMuGR8Xy0tHQeRQARSzCjuUd0TKJCQkOL4pmKrQ/u63Rjmn84ARrhjKUG
2JC5Rzy0QKcCJr

ZT29lwzlqu9rr71mLd5ycydNmjR//nwpTaY+XApGkBF/5513pCSvvDfffPPDDz+MiLiYOHF
iz549P/vs

s/3790+ePLIHjx7cVH369Bk6dOimTZvkXBs9JxcdMiwG58A6mtcV/g7fR+QURSAhkOYV
WCKzkshZgs6K

iVEyoSssgS5RISgEEZ7Ij6VCJ09WcCIHsFOx2o5HRR4/YV1YZMTJ2BiMdHip2Fd6SM0S
UIKKNsuAGxZO

RUFtiYr6UT5UUuK8iq/lhjKloTYOWSFpqe/oEOLjETIL6igs14jUiewd9vX1Oezj5+/vF2BZ
Kawjtvn5

+fpiAf6Y35EA/yMBR4oTVESgQuceDQxQRiIQK4WUCggIoIHg4GAu51T0KSzawawpL4
U9ZTaMAwsDa5Ki

onSWA1JeBk1nlQ9lqMQuX2lOFb0ALeTmkgqJr9RRvWujzysfKcN1hYaGIjd6Ui8NylVfQl
5PT09e8WvW

rFmxYgVFefUMGTJEHy4L1JdVHqKLYo4fPx4pJbRFZadPny7rmm+//Rb1ZazZveeee+bOnf
vFF1+gTBex

6Nqo0TY4iRLqC7zq2KI61h0fHY2zqKR3KA/OFQ3D70tCHyiOtF4BnM65Xl5euHVx01oZ
ysFX2WF/bUeC

gwKCg/yDAv2OHvHw8T7g5Uk6MCQ4MDgIPRBDFWwNEP0ALQ6l8EWD/PyskPf81Vc
kEPFmcKw4MiaGVQUJ

GWcpc74wtpyOIyrvzUxbw9ASWTPqngGk3V2/7PI65O3tc0hbMbxK2CEfb8zzkDfDiHn5H
Drkc8jHx2fP

nj3bt2/38PA4RIYDDBTq7n3Yx9tX2WEfIL48OBevyBRERREVENYw6UdyiospYDCkxtUJ/e
XUB18XrecyY

MYsXL5YcR6S83jkXVIXaMVZ25eekqH7KR0qazjDsvB7HjBnFOkcatyuUMo7pMpGj9IdX
XValv+Dz4qBc

9UUgGzVqxCt+165dO3fupGjjxo3J1IdLgfRy73Gzccr777+/bNky4lpuclXV9QXIWemZ8+e
Lb9cxr30

wgsvUCdekhO1QF3UqNE2OIkS6itKwE2OayCHXaIQbngbSyfPE87iBU8rVEvkiNIAcESk
AqhP/QKFy/h

UUmU2Po7WFGeNgTYp8j8AwNDw49baSvmO2yphzJLRLyVSbqUHfY+awf2H/A46GH5
wyL1te2cMBR4ABln

TuXSCEPJZ6BkCi4A/AYOukL11SPP0p+GaB2l3Lp1K+lfdu+WIRJjWNBLZYcPM0KlzMf
P55DfYW8/X+yQ

ny+nBAQe+Xnr1pE/jAwJdd13YL8lt9Z4Udiqxxpnf79DAZaRoLzj1NgmwfWBgweCQ4NPn
DoRHXsqOoYw

8aypkT73W/xcKdc4ePDgkSNHkr4wpB6mgwUo08RwSeWVRHV7BQjzPz27/9dwum4Ey
e5s+IiIsKpnFIT

TZ0HnMI4GfXVIIoMx9VXXz1w4EBCXhIyOvpwKRCYnJwc+X0xppaoF6F1cXHJzc116Yc
YT5w4Uf54zJxx

I7H4HDVqFIcoIPp0caPH2+AUSse+OBpISUnZvXt3lagvoAoPPPAArZDWFSnkqCCtVwC3
Ax1DfYIMHVXW

toBSZh8CRy//YeeP+g8YeNjXLzAwKMAv4IiY75FA38Cjhy0jEeh75IgvMvB+EWvXMnX0
yNHDAUd9Aw4d

9PL28Dpx4tTJ6BiWFYiDSK9YxXC9Ir3yl85vv/22f//+7DImusT5I/EZHqM89aVyGmUumG
sSRJlrFQws

sS/DYumuMgS1yA4fsoS2DPP2P+z174t5+/sGBAViP4wd47Zg/pr16whwi1eiLMDPW5kIcJk
m72bv8zgQ

hPpGl62+chXWf+XDFTGwzzzzzKBBg+TVdQFQD1tRX8b2gtT3hDI6bNm0aVPi4mKM+l
4A5aovN4+MRUhl

CK9+1p5pRR/hLRMEhgKcIgl5q112OSoSS0JycIL4LBJ2mYsea7ANzkJeb7yGcVhyb4t3Rnp
XrVpFzpEj

R3AluJ7z9T6ObN++fefOnStXrqR+qao0umgp5HkmzFbfotjXUU99/X0tC1Bb2+yjvlZ8ZpvfF
19/PX6i

y2E/Iq2Aw8Rlh7T5efv5e/tjJKwcHen5IkeICHri7+0XYJkv5rXPw/OAqG/sydjY81JfBZ49Jh
x9MyZ

M/HFXF1YWBg58uZwOVYRKAQ6wZzighUyfw6luPJGFLSw9Pzx5Ur165b+9P69QmJidt
2bD8rvVy9pamH

iwyhPYR5FTMfL0t9tXn4Htrn7TFhiuu2Pbv6DOwXeCz4cGDAkWPBjmW8A3wrsEPKfi74
YXs99wcfD42K

PhETF21ZrGWxbEV9rTfOK3opyuCimg899NDHH38sg6OPnQ/SClXJO89Wq+cD/Tx5KjI6
GouiM/HxCcS+

Awb048WrLsSK4OmY9NZGemsjZWx4kaAXvr6+xGx6Ui8Naog2VAnlvYfMbaNTGRksYJ
Fb1FfvK8zffQ1V

C26aMRf1FV9juWYPj3Xr1q1evRol9vHx4bbXDuVC6dOnz8SJE7t3746u66wiRbfRucUR3c
UJYfH08NTJ

g56efkcCREctSXUwO+a1cw4f0eZzxPdQoK+3Nr9xUyf9MHHsjv27D/p4eR4+5IWpv3ceOn
T40CFfZYfJ

8TzsrYyED+blQ4nDh70P+3od9vU+fHdfQU8P7xOnosMjTxD+ijjYdi4ssbaeAo6PW/ft2pU
rVyQlnQ47

fowc5bXLsnPBGOKXZU7L9M5Shvk9GhR04OCBjZs37di9CzsVH7vX44BvYIAYwnnoiL9
3IOZXNGLYYe+j

P15F5n30MON51oL8vZUdCg44HHJEzAcLPXJIGWm/0ED/EMtI+JYyMg8H+XsGHDpw2
MMr4FDgsaORpyJP

xvCK1HYq2rKTP6IYCv1n4BP6iSrRLQGdY3DtfEmX9+qqGDmLSIDf8PBw7gVyKo+atciT
0WEno4+rlRar

W+LgqNBjwXRK1LcCdCVFCYGrI6b38vJCGvSkXhpUpfoaKkCpt8Ep4Kllkcddzb0NmzdV
JkhFfdmK+pKp

BKNyEV1ZUCf1b9iwgaoQYAG9B9WmheyWgELxsZZZJMSfjI9Zv33zfj9vj8DDB48cOnjE
2+OseYkdxAK9

DijbH+i176i2PUFeu4ts6PSxbuvdvSICth7es/eoF7bvqCe2P9B7f+AhbN9R771BnnuKTJ3lvee
o975A

GvXxCDjs6X94r9fBgz7eJ+JijkWER+EkT+oHZyqDUt9T4eFhcXExaekp7u5LBn8/UEILFPn
l2LnHf//+

/cwpi/icnByZX0dw5QS+CMnxcBYMUVGnTkaePBEWGXEkNHj4mFETp7iKEcK0mzJp7
BSXsVMmjpgkqNmHM

1PFio6dZRmLslAnjpkzEKDZ2mssYZWNnTHK0MUVGesIMV5fplk2c4Tp+huu4mWdt/Ex
XjiqbNHbq+B9c

Rv1ycPfxE8fDoxA9bRFR4VEnIqKiImSUMPUQln7ynIuCsLAWewtcrI0a9fNDzmLQgoODq
VPqqTR0NTLq

xPGokyFRJ0LpplUj/T9BrzDrbwRwoVKQaRUsWi6QJpPWjylIhISE7Nmzx8S+hmpBj7fBK
eQW/7wvt7qs

9LnPITeIIJzikDh3rZPniWgtDgXPIm5FhFwUXTkZC9k9iwohCSdVRHmK6Cci7mRUWvyP
v2z0PhW097jv

zhCvXSGevwR7WBZycFfigV0h+3eG7t+BHdu//dj+bcf2bw3b/7PY8f1bjju/frGxLxIGtUR4/n/
DYHX94

20mPLVH7t0Tt2xK1V9m+zWIn9m06sXfjSW0bTu7bcHL/hpMHNp04uCXSc1u4187j3j8H7
NvpfzA0NjI+

7XR0HJ3lEmw7R/RrXfAp64/FqAhrGwLfhNNx8QmxyjWH2wLjYJWK3nbv3i1/97XR03zG
+qIGBl/m98iR

I34B/tZbzX6+++z09CO5/3rl9z/598gdabz9fTz8fDz+fg/6HDvp7F5nXwQDPs+bv5eHv7elnGQ
lWJLbt

D/De5+9Vwvb7FSt/wN+bYrYdCLCa2Hf44L7DB/xCA3bs3+kb5Hcs4tjxyOO2hbPIVXmc7g
eHhAQFBxPA

n/3kkjydkwC+iRQFGzdOl/kLEYSICCAau2qKknYcSwkLDwwLPyolbB2Q5Wd7YwuWhx6
TnNcDgl2ZRnB

3AFpumHeeTZUF3q8DU4B9U1LS0NfxTUDvozb29PTk62/v/+OHTu8vb3ZBa8KwS/oVHF
wIh4eHnv37qUG

EIIV4GLYPVAE6WIctMxTGan9ngf2eB8IPBX2lw//+c+vPvj3/z578+sP3vzqg39+9f5bX733z
6///Sb2

33ff+O+7//jm3b8r+9s3777e5d2/drG2kvhr13df7/Zv7K+WvfvXbv96rdv/vdb9nb/+753Xe2L/9
9de

2l7Dev/fX7S982rvf73a+91Xe//7L73e+2vPD/7W48M3und+q+tHr3/0zj5/771eB0LDjwUFB3l
5MURi

ZcA46JSFp7e3l6+vz7GwkEM+XhgjgXl6HhTz8vIoZecYf9i+fTuBr8xsafVFnsWJsyRiJYUR
sh+PiCAI

jomPy8zJ5kwsnlFmcLcMwU5Z/KV5SnLLWG5Z/Jyz+RjqkxBtoNlnckvYdlWyYI8ZarmY
uXZVTIUmJdv

FchNzU7NysvKP5MvVqCssIwfiiGDrIT+8i3vAahhKPkYGodKZ9pwU3AiSCUpKSlZ6psHp
aoKKGqOys+O

ZVZ2SuGZHPpSUJCbkZGmjmoOT/2O6AMK6QntCqQRXU5hrUxJ6eolglFfJ6HH2+AUuI2
JfYINcc0SlkkQ

LLEpy3N0UYpJ+QpISEjANZAQx2EnqLzMpToNne8SPik1OTkjNbsQ7295c0Qi78yZPEsq
LMs5U6hev2Uk

Ms/kZ5zJSz+TSzqtIDs1LzOdImfOoAQY7j7rTG56QWZGofW9coXWuVmZZzLSzqSxi8fP
PpOTdYbryWVL

yUzL8qgz6wwnWOek5uWmZOdwhTn5hWkZWfHxibhertjBisH44EOp3HEwk5MTrca1gXh
tKMzPz8nOzmRb

IHMOaILxtCuXRiUNpG3pJZbasGFdUnIyAhwdExOfEB8eERF/OmGi66TEpCT8PUrMKX
mWeFgU0GVLBZHG

s6YUkfwC9FJktWKjWMXllfQyn6Rzs/Iyx0wYM2fenGU/uksfNm3ewKHyRgNlSk5OJjREJ
kUCyZTRloS7

u7u8ONIKPmU2btyojlvIS1FETnJsOIVR5UYYNmyYPYlSGzgOuNrSSamhIIfXDjlnpcb/uJ
AzfA57Onzb

tj6RrV2V1CCKKzllwg0rJS8djPo6CT3eBmeB38GnILfyDjPSK292iRIHBARQxnYxFYBHK
OUUCjMzM4hD

JBqx9tXCXzydXSeTXurEsikoLEhNS83LzyMhOZzmaOTaJpDpiPWNw3n5BQgH0Kijm8v
PLcjLzs/NKiQS

Q83P5BXqrWVW4xK1iIYUnsnMzkvPtPxsLlpdaOWIUWuRCXKiZdIOidy8HEX2URRxyuo
rk/NIS8K+CPXl

zNYY6vrKASeO9sioskuCrdQApJHeqKgoIqdly5ZnmjRJfIly6kJ9B38/ODEpkfWWz2Gfqd
OmxsTFbNux

NTQsJDr2xJqfVvk6cNC4hKbaQ0Thr+qJtTa3AisasovJ5lqJbV53PqigrNa8wd+jwIfTcxXXirl9
2bN26

xc/v8KzZMxYumj9u/JgTJyO9vb0mTpzAYmL//v3z5s1bsGBBz549t27dOnnyZDJnzJixfft2xg
Tlee+9

90gsXLjQxcVFhig0NPTBBx9kWJDVuXPnLl++PDExkWFBp9WAWcgAonacyyG23bp1Y7
hmz57t6el56NCh

mTNn8mKmrWnTpqWlpaGamzZtmjZtCro7adKE+fPdqIMaXnn1mc0/r162fBGzTR/WrFkbE
hJCo9TMuZzi

4eExZcoUbjfS9hcml0DOAE0Qi9v3ziWCUV8nocfb4BRYaDPm8ndfiY3++c9/Eq3iiUjjaHx9
fbnn8Tvc

9ucEB9SISxf84IGDB7KyMjMy0nJyiMYs8SBB0CWNUj8VUu2ePXteeUVqV8Eyba8XMv
yLVPfzJ+Xg8bl

nMLykhm4uMWm66EK+t35YsZXg6f3+fb7/93w8/DKXRHOsP3Gn4dw6xxT9mZaWjg/m
52cmnE6xu5eeN

GDrkzTf+Vqhl3yOFtBuPqGP9TNItgSKjOCS0zOt4MY6bJUWeSkhkpJBZ6ztlKmTMbwzZ
yQmJmRl8WrH

O1thbsLpOHYJdhk0tnQvJubUxInjp0x1dQyYKoDxt6Mi2TpCDhOK+jLRTDdyxSzY6hsRE
Z6Tmz1i5PCD

B/a/9OILbm5zBn8/KOCI3yuvvpyWnuziOj49M3n+gjlWPHzWrIuyrqtIUCs2ilVcXsXTVKv
Vly4P/n4g

2/97520yP//8P6tW/XjyVFSXLv9NS08ZNRk63/767x5bmMUhPIsJrZs2TJr1iyuaPT00cjhv/
71L64a

NUWu0DwUDp22vrzL25sBGTRoEEtMJHD9+vWsOYYMGTJx4kReaOQwSgymjBvwQiEf
9ZWGOHH69On//ve/

t23bhlLu3r3bz8+PRSqVU2z58mWv/+21jZvWRZ0I37BhHfnM+9jxwwYO7jV+wmjC4lmzZ
v/zn2/Rh/Hj

x3MiSwQ6+fLLLyP/xNb9+/fXrZaCYkDHuBwSOvfSwKivk9DjbXAK3MYs3vHleCKcDi6
MQMF+CItdf39/

ikkgdU6IRXANS5Ysjo+Pu/mWm/CV3/btjeiu+2nN+x/8m+2SJYtG/jBy69af8Zu4KgLiLl2+U
d2Qtbwl

RcWtGHjE5Jx0+XtkniW9diAqnt1SXEcb0L9PZkbSypXunl77vu3bc8HCuf5HDo+dMCosIvS
7Ad/2/a53

4FH/H0YOd5kwPjQkqJAqz5wZNXJEHiuGvFx2rRyrEesPjaKgdseIbLNzczKy1a+hFOSxFD
h7THVVEOmV

c7NzMmbPmbn/wF4vr4PTp08hhps7dxZFpk+funbdanYHDR6wd+/uxYsX9Or1P+QZAZ43b
y75ql2pviIY

T24fZkHv14IJRYARXRg1ahRb+fuvmv2I9evXEavNnDFtksuEdetWR0efGDFymJvb7OjoqG
7d/ztj5hTv

QweLuiH9qTwlRqU87GIFWTKzrDn69OnFzltvv0nOe++/u3r1yqioiK5dv0lLTx09+gfWMQ
Tx4eHhe/fu

7du3L8vEHj16EEHK++pLly5FFzmd0Hbs2LEocUJCAsJJmnFgmfjhhx9S7KuvvkJKUd/Bgw
dTW7hx44hu

7TGUhMS+3BcoHznyjcJoOdFz165d0VHORbIFFD/88IN3333nWFjwkCEDhw79HuWmht5
9uuYXpNesdWVc

XOzw4SM+/fSzsLDjaPmuXbvo86pVqzidLUvePn36VDCDwFGjvobqQo+3wVngWfC/Eer7
BGJjY/FfeAFy

JDBiea7LVYKFixb4+vogvcjGp//5JCUlqe93ffz8DqO7JPChuE5XVxekpe9336qIseCTTz9Wp
1bkzY8d

O2Z9u4af35atW2fNmzt/0cIFixax7f1tH2W9+yj7Vtnbb7/117+8KnbvvZ3+8uqrDzxw/3PPPfv
MM08/

++wzGLvPv/D8k08++dRTTz3z7DPvv//eZ5/959NPPv722z7v/fvfH37wQc///W/RwoWLimCt
IK0zOLpD

iuzc7PSsdBK5BNDWCqBMbKHKz8hIGTS4f+GZ3G7dv1m6dBGRN6PEIAwbNmTFj8sIh
RkZdBepJjgmGiYU

ZvvxJ53V6VRyDofLKgpRqcAvM78oE1tBfZolPCLieGRk+ImTUeqBIGuhIO+Eo/25eZYIJs
ar6N8ylZPN

bl5+Tg5lHC2/XMvlxIJzl8/OzWKbV5hLj0rjQ5If0pYkUhzodaKkHWhUOLCZbd0phRmrNj
qXAcoYJ9C

GXnLJ1n9/KtdXsqoaixkVw6xKwIVyvrzQUG++qNLfpay3MTE0+okjZwr6KzyoSfW3OTkc
HuS0O1cGhj1

dRJ6vA3OQmJf1Fdi33379hEVIb2kyTx06BBIKuMd8A+4GwIUIrzQ0ODdu3cWFORu+Xn
TypXLCYI/+fSj

+fPnJiUI7Nu/B9WJT7CUzOew9+Qpk9jiVRzNx8ebiLB3715vxfXWq0Wgpq/+5dWX//znV9
QuW0kUR7L1

wZdeumxRx99/PHHXn755RdffIHMhx95+NFHH3nhxRfuv/9+1Jc6H3vssccff9xSa0unH6D
8yy+9/Jfy

efvt/v06T1jxnRPz4Op6akFZ/KzczJz8nLyrT9vO2JFvYwchmihvoVncry8D0ybNmn/gd0BRw
5zmcfc

QqZMdfX1PcTqxMVlAsuUzMx0cry8PDiXIWJwaEXN0mBL4v6gs5SWF1hJnJzidWC1A9
U+Pv7E+f5+h6m

oV2/7Ni9exd9YI6YHQy5xRJOxxZZnLLY+ISYuHjrOS3rUS2M5VmV2qnYU2zjEmJJRJ2
MjI45VdS09Vks

bfpT4md/kRApSqLTCQmyi1KSI7sC6RJlvq5IicU4UfJZZgGvf7bkef6zDCXfPt0uA9b5Dv2
Bog5Ypciz

sk/HUXdSEkd031RB3Xn6XAKi29LIO1KcS2e4Z/XsXhoY9XUSerwNzkJW08gtrkRcAxAb
kYMe46R1uXOB

+iq9Qc4z5UmlrOw066mlM7kcnDlrKon8fLwG8YFEcrY+CaTzjh0L/uijzhK5FlkxXnnllZJZlc
YSWEvF

/4IqP//881aOxSvqoEYKVAZKfvjhh8TleflcbBmBlHak0C83z3qDWpTYvna1Jfq3Lt/T8wCh
p7wZgCpL

ZnGkfEVkZWWlpqaWVl8bSw2KII0TJ/BFdKOjT0ZGHk9NS5I/6DJNmAS7WNFDZ2Ln6E
NVYYXLefqJAXXt

vwpZf1jLQ4Wk7YGSBEhanVEMpM4+S7DLl0aXUPeC3A6glmK/9ipskGfWUnrn0sCor5P
Q421wCjiRzMxM

lvlh6mt0iI2QW09PTyIkHx8fgqRdu3YRUbH0llV5hSTFqHhFvkQiLj4a9Y06EZ6YGJecksA
uaSw6+kRS

0mkKsJrH7xNIRUaGJyefTktP3rp185tvvvEysefLL/35zy//+ZU/S7rliHv/jHCyrSQvvvgi5eVE
ajub

pnYrGraPvizV2q04Qo6dKWmB09m++eabmzdVzvmC8pUiJTUtedbs6W5us5JTiHdOpaaxyo
my3tePj05K

io+MCouNOxkTc5KYkqCI0VBPnZ9CFNPSrWiHNGOFcVSmQBSUBLWLjtwWilfx1+m
MJApbzgjuqAWWzEn

T52lloo4cTIyNvZUdk660jnMWgaJBqsTLfUt0mApYFkFD1vlFeYVfU635KEKTMrTovV4
XV52vrUWKQ3X

IfJm4bhDugT6jLKQo1LM8VypikNs7bfxSbBCtU6rBFIPNTmYQ4+Lz47jIeukIvTh4ICGRQ
C6y9TrrEsG

o75OQo+3wSlwS6O++G7CXMI43DdeHu8sYgwHDhzgnmf5T2hVMUXvhuE7tIMuCuPKN
kJkO03JhNMx//zn

G6+88mcV775Kwo5K7QT6h0xa4S9ZIUDOAimvd9SuTjlgZ1JSytOcnGU3JwVAclBfEv/85
z9RXxmB4mRi

CxbOy87J2LBxbU5uhnoL+oz6k2o2+lpQkCuqJkFeUbRkDV1WVvkZmZrrkkMDwuXJ3MB2
0JXrg6Knxy0wf

R/V+cSjJFMvffcePHz969GiEHHVHfa2//0aE7di5dcvPGyg5b/7sAwf30AjqKx1WvbLA/8vW
3X3pAc8D

v+z9BaUkB7m18s8UZGRnpGWmEbmST6Ytw5itsuWZIKeeEtm5qG8ZF1JazCpAn1MJLM
VT2CeyZU4ZKMaZ

O4J7REpeAI41A2I9QB3SucXVI0lcunSpPPAI48aNmzFjBtPXv3///fv30x/Jv3Qw6usk9Hgbn
AWeBbkN

Dw+Pj48PDQ3dsGEDgRHgffDUfufx1JXIN2xjV5mlJZWwfFwMioY999yzL774gmgcWvvc
C8+TkDhVxZ/I

s/NAXJ999lm2svvcc8/RDRLPPPMW7szdF6PQUkKV69eydqCBczKlcvYJ83g2G+rljbHY
XHMt99sZKYk

Qc7u3bsRee6aoKAagcpKSkspTX2A28eDML07bzc1NRcAnCXxR37DjoZs2r//yq/9QbMDAv
lu3bnJftuiL

Lz8NOx4ycFB/f3/f7j26Dho8cOXKHxH+adOm7dixY/feXz7570O9+/f06du7R68eiSmJ3Xt2I
4H0SnMg

mnph6punHkEvE5Goc6JLF8fOdywjaZBdGySQhYm1tCz1qwb6hHJaKYEuqtBZCp1VCmaq
e/fuW7duZTZ9

fHy2b9/OXfnmm29yyldffcUsVz4Wvzgw6usk9HgbnAVOnPtZ3pM8dOjQ5s2b582bRzRMb
HQBT13JHy/P

19LTU9AwW33vvPMO9gg+77230yOPPIL+PfTQQw888AAq6Ez1pbkXXngBoX3sscfYJdJ
9+umn77nnngcf

fJAEW8fOpKdbzz+XonDp0sX8RxBJKLlm7SoCX/WWABQTV9sch6V4vgXB63333UeMi+
iOHTvWy8tr6NCh

/fr1W7hwITPFPN0iXRpRX2Z23bp1yGdSUqKj+q5es2LGzCkuLuOQ3jVrf5w2bdI77/wzIyO
F2H3tutXu

7kuoYcWPK6ifpcb27dt2/rITud2+c9ux8NCRo0euWLUiPS9/ab1gcGBEgrn5Gfb0ivKWrFJG
U4k4QT1

rQyrV6/mFmCxUnpNI02A3q8QXVShsxQ6qxSff/75qFGjunbtShkWWJ6entu2bXv77bfZNep
rqEb0eBuc

BZ5FHu8kKmKNP2vWLHI3i12cdSWfulJOA7U4Kxil9aMCW7VqpZIwHWJ27Hi3+v/Vhx5
6sFOntvfeey+S

/MQTjyv1reiB5CrEjnfpaForabj//vs7duxI4u6773ZUXzy1HovirFy5wvqWj4LcnNxMBgQBjp
NLZiOL

IRLmOCwO+cwShwonT568c+dOpoZK3nvvpbZvxfXWihUrEAkcdMWxL2fJ3xR69epF/IoY
qwd7T5EdHh62

4kf3rOy0F158Jikpfu26VcOGD/7ss49jY08tXrxgy8+b+vXrO3bsaH9/v+HDhw8aNmJd08PX/
3C/Ad+h

vmERYZMmuySnJvX+tteIH4arptCY0nYO1Ie2daK8d56dBkPNLYD+LVmyhAVHBaNaVa
g76CxMk7e3NxrM

MmvkyJEkQkND7XeeKaBPuzQw6usk9HgbnAK3MZ4FvxyvPiFB1IsTx5WnpqaSSaTl4+Nj
l6wAKaNgz5Jh

Ud+yzHbHZ23IkCEImAjeXXfdRcT54osvPPfcc+QR8iK9RJ8PPmjFvgSgtkhXNzT3/PPPv/T
SS4S5xN+v

vPLKnXfeyS7RJz1kTeCovlyCHoDirPhxOReI+pLevn3r5i0bPTwOZGYSKDuqL0J7DkMDC
goKpk+f/umn

n+KC9+zZQzokJITYd8yYMStXrvT3969YfZFeFDc8PFwWW0yu5ERGRhw5EpCbl22beqs8
hz4rs3rIy0TN

lEXR9xhb5Bfk5VFSHxUrIMcK8fnfwdRXmKnvLSvHiOfYIrokMjIzUB2uRZC3VgSd9athP
EVWbaRyGWe2

9sNN9IQuSRIBdUSjs341ujqF3FBpaWkl/u6jumy9U1X8jrv4MerrJPR4G5wCNzNjnpCQgCP
GKW/fvn3N

mjWenp54ZzIrH/uWg+2RbSubjz76SEmvJatPP/0UsSZRLzr37LPPyOd0Sbz88ksk5EEn5wiw
etDqlSef

fJI0IS9a+5SC3SeeeEKWAqozlk537tzZWoaUpGD5imWol4vLBF9fn/nz3dSzZjhYeX++ojEp
E9w06yTm

hZiM3S1btpDD7u7du0ngl6laStpYnVBygu4SPyHY8gN2gYGBR1DdgACmePfuXxITE8RKf
MRWdsWSkk6n

pCSlpiWzVR9AtR7GtkuWZ+rcs59wLQ85mpycTIKXIpJAwKZNjrrVyON6h2FY/2SQICzs
rLoDF2SowLR

p43OqjpoFN2VPnBvyicOBA4By2IWBeyrnuBLAKO+TkKPt8FZcCejtceOHcOJE/WGKeTr
pXDQe/fuxQEx

L3iBCpC5cyA9LT01LT2F80pZui5SBDE3wgavvCJm/fVXEpKpdtm30krwrGeSS2MVqjqoU
ARYKndMkxDF

ZX0gmWyBCKkEaN6GDT8NHfo9seCYMaMYDD+/wwgi44CGEQEztMoyrQd7KgTni74y
XGypVoad/MxM61z0

IS0uW9RXCX9J7yx/XABWWmwRFUZe3vNAIYvembD+WMDiQMJfFQHrAL0o/LVCdg
mRSahTzg/pW2noOVsK

kJDAV8pXE9Ko3ikLuwCCJ2pnIwUEnVWIUC0vHhkBRsNGjtIf1fIlhFFfJ6HH2+As5PO+tk
eWCiktAgzE

wdz2uHtbTipL0e8llLRSeHt7I2BKzkTJtOKSI6om8iYaRwRs/VckhNWHEn4L0V22oriyJYc
CqK9dDLgQ

GSWbAuuPlyJXOWHHj7m7LwkOPpqTg3Rx0NqWNyaOSFUIgGgAOZJGfclnF90lwSGC
ObbMqeWkS4mEPHUI

7zbLXNtbgmE03dZahNbWWn3+WQrQXVYPxL6sruQhMk6xz9Wlfh1cFK86veN07KGTB
MNLf6wBLT6k1Qct

SkIateFWBVZL9EcKXCIY9XUSerwNzoJbnahIvDDeOUG9C00CH20/8wxy/0u60lC+hJXB
vn37RNvQNdkW

iZz9+VpL7WSr/u5r5QgqXS2oFnUzdglUQQ3Sq1Oqwn69e2WUHFGX6DgC2uSv48izsnOj
6rEmS2nx2Ude

kWG2FMAjO6ovOUWt613mVP7WO3To0C+++IKolykWMQ4JCZ4/323EyGHU9OVXny9
bvhrxjYul4dzIyHDS

8QmxJAjWs7MzEd2NG9fv2fPLzJnTCdwJ6JOSTqO7YcdDOSU7G+WQdvWVFu2eA8feco
H2JTs6YmXl9fE

iROnTJmC4BGG2t1zAjTdt2/fbdu2kebe7NKIC1Pm7+/fuXPn1atX05/fcHB+E4z6Ogk93gZn
YHIGvHZy

cIJ09Clc8P79+ydPnozo4pTRY/z1r/u7b6VYtGiRVjCLVx9//HH1aVpL6p566qnHHrc+7fPoo4
8+/fRT

KuC0n3kuqYXVAWJvf7IIPaYb8ndfyVT5Z/vAhehLqjT49Eoi5ffs2b1kyWIUF31KTDw9aZJ
Lamoqy5f5

8+dz76QUPQ3L11F9gbQd+27atGnMmDFJ6mNmQE5ERPiGDT99/vl/UNOxY0evXbfay8tj
wIB+fn6HXV1d

Ek7HdeveBcVdunRxalry8OFDOYpRfueu7bPnzPzmm699DnsvXDR/6NDveSE5vmUqrVcG
u7ckuLrqfue5

TGja7gadZ5wZKHFKdr4TYB4HDRq0e/duxoEF08CBA3v37v3RRx9xqEePHqzAzmtgLwK
M+joJPd6Gagdv

wj1ckJOTnZiYoL4RPg4vfODggQULFuCU5eGXIOeO6OLVRpH6ioxZ29tuu81Kqdj3oYcef
PZZ5PjpRx99

BOktUt+KdPfAgQPdunXTO78C1B19feGFF9TPMFiR71133fniiy+qTxy9ytZeCqh1wKsXoL
7nBfEl6rju

p7WhoSHM2r///S6Z/fp9N27cOC6ZUCk5OVIEgi0O2IEwSDOnqG+c+iHnIUOGEPsyv6K+b
NesWb1165Z3

331n/4G9K1eu6NXrf6+88nJObhaqTIw7a/YM2l++wj02Nrp//++2b9/q6Xnwq6+/2L17V+DR
gJE/DF+/

fl10NLq+8WJSXyJOtnTGye/0y192mSMmiBUAQ9GvXz90lzQyzCxLsUsHo75OQo+3obrAu
YhZ0ovhYWNi

Tp04GYn3DA4+OnHihI0bN+CjxU2fz3ddXSDycSMRVBQXqXv44Yclrn322Wfvu+/eJ5988
rnnnnvyySdQ

O/XO8zmiXql2w4YNH374oc66UGjooYceIt61mlSN3nNPx7vuuouE9Xlf63sxLdShV8v/xqtf
g54v+ZNq

ZGT46NE/REVFkI5PiHVxmfDZZ/9xd18aFha2c+dO+QE765wirAqKiFYkJia6uLisXLnSnm
JAgIOCAjMz

09HdwsL8I4H+7u5LiHSJemfMmEZzmzZv4FB+fs6Uqa4HDu7jpRIXF0M0HB4edvJU1L59
e6h/1qyZ27db

b5b+euTvrHrH6UjTnp6eS5YscXd3R30l32kwm7NmzSICHjFiBFo7c+bM0NDQpKSk2bNn
M31pFf6O5EWJ

UV8nocfbUC1w02rRtQ23TkCDh0WAFy6a37t3T7wqoTAellDY/rtv9dGnTx8lYNZfdkncd99
999xzz7PP

PfvkU08++OADSNI1LL73YqdM96ruuXnF457lcdL3q8znEo6+//ro+cJ6gqQTd999/vywInn76
afpAl55/

/jk0WFYDuqjFq3369FZDWswzWhpYFvrwOXCcLy4nhxj0m2++ZqYINw/5eHXv0RX9Q48
/+fTj5GTrh/bs

cLN0E6K+hFORkZG4crYQqzh27JgqotuSx56x3DzCrUy26qigy8hHp3SeRrdIHxh5a5tXYH2
kNk8SeWhq

BUiIyVkk8AM0LP1AbTY6ywF9QMhplUSfXApap2IHxZWnzWXWBH1AobOK0Lm/Dvup
ZhZGdEnSJDLV100j

wPRfMi8RjPo6CT3ehqrH0ZWL5SuFyklOPh0dfSLqRAThIpp51wT1/fuR4eHHvb29rTOLkI
qqFlFfoehJ

q1efeOIJJX8WRfkkRX0rCnxB11tEfHy8q6urPnb+SB9YCrCVXekJWzpTtGvRu3cvES3HB6
mUKJSBPlwR

WgLF9MVYk5hPSOrre0hEUalgIencvBxiWWq2CpWaLHaR2KCgID8/P1ZU/v7+Pj4+hHd7
FPv3709KO2k

Y7wG1EeTrG+BFmOXTMmXMsg/Zv2SVVyM+shSmcTHxsbFxsRbRuJcsCxcK++Hy9+nJ
R90fQqd5RRojv6E

h4fLqkWtXixI2+isqoMLBxkH+iBbQIIZXSHMRMa2JF8iGPV1Enq8DVVPCfXFTePT89LS
kyOjwjDc8pFA

vyOB/r6+Pt7eeOaDXl6eW7ZsYa2NLxC4/6ucrl27vqx+wo/tK+rBKrZgvcusfltQHrZ6Sf0a4Is
vvkia

fSurHPTIFgfhGTRokC5xPtCQfK5XJaxfFaRLJNm++OILKm39NRrr3r0bGqbs7Nc4MHplog
9XBGWkNsz6

icb4hBgmS36uMSMjhW10dBRKjBCigkqYYkR9y0TalUkUL49DF0kLCjqKeEsxpfS8PMrD
eg+81JqgimHp

oVO/HfRBImC2spRRS5qSqHVSFSPVq15YqNWaNa0E5WyZO0lcOhj1dRJ6vA1VT9nqm5
2TjluPiTmBK1cO

nYT1u+thx49FRIRX8psmHTnf8vJFVyCxJqGkMkHSHJT3peV7nu2jZSMdKJMLLeCCrVMf
ErF1rZVDUMfj6

66/yrW+osN5sFY8J1ruZxXHMI3Q5UAkFrK+8wNR8ncnJzVTfF41HFv9LfGyFQSKEFasv
WisfOmKyg4OD

2WWy5f1nwmLaCgoKxFRbZ18nhYX5OblZqWnJycmJaekpWVkJZRNu2qY8FU0zDbOrUr
wDhQfAc5ceZyCXQ

OssfrpmeZGZmMIPIXZr1Cq5qaG7fvn04QxL0hG4AiT179mRIZRH+VjDLFyVGfZ2EHm9
D1VNCfTHu4fzs

nAzip9jYU/Iz7/K+IrGUeo8xOiAgQJ9dbSgle0TVit7JLYIcQk/JJgCVImpbNrre8jmvB7KkA8
UTorh/

QX0lo8hougrkphUaJn6GeCiPwGq95yL/hxrtSjvPOOXKacyy0DUly1T+8QTT1iRsvq6KxK
o7+Ytm/bs

+eXkKeQ5fMyYUZmZ6aGhwaNH/7B37+5ly5dy+vIV7hydPn2qh8cBX1+fI4H+JCZNmpiR
kRYefnz8+PHW

52LzLlwVpOeiPcR2CAxpQQo4ARQOJJGQkDBo0CA0GKfk+GdgJxASEuLn5zd48GBZhX
Tp0mX//v3Tpk3b

uXPn2LFjL0EnadTXSejxNIQ9eDFH6dWGnyXYRWtxu4Q4OFy8MC6VTDy1E555VgKme
eWVP99//3133239

xhFq9+ijj7DL9t577yVRVeolOPdzPpCF5D/77LP333+//Jovu48//jjdeO65Zzt2vPvxxx8rS30Z
Ujgr

GFpASqEPV0SJ+bIkdtbsGf37fxccfJTdwKMBb739JqEqOb1790RGK35DEq0VuR0zZszcuX
PRFXRX1Dck

JHjr1i3f9u0dEhL0j3/8bdWqH0eNGjlkyGBC27FjR69evRLdnThx/Lqf1kREHB80eMCOnds
Q4H/96/9Y

pc2eM2vYsKGM58CBA+W6CgsqtMJyxwQQG9QX1QESVEuOPlb9yEDZuLu7M2LyDJQ
u4aylQGBg4MaNG0kw

p1u3bj106FD37t3Z/fzzz2VMVKILBaO+TkKPt6HqKeHNLZP3FeMTYsPDw/CkbHv27LFv
3x558OrkyZP+

/v767GpDCZgF4gok7r77LhVfWjz88ENPP/0UKvjkk9YYPG1SV+grx5T+QZXdGfs1Xmn7s8
cdQ33vukV8Y

vMuZ6puVIVGgPu/L2sjb25N1EnOHQLJ4mjdvLnEqwSvipU8ti2gFbt3FxeXll1/28fGx/uRbp
L5r1qyW

DzKhwaLuLi4TaG7Fj8sQ3WNhIbS1cuUKXiTjJ4zdtHkDMfGAAf2Skk7PnTu7V6+eVMJIq
nasn+qt0Cr6

WynSgt7IV3rZ6uscSk/KjBkzfH19pRtsJaGPVSfEvI999RX9QXTp2IoVK3bt2sXwBgUFDR0
6lGWTLnfJ

YNTXSejxNIQLJQUYD45Pl2dckdsPP3y/73ffurpOwiefOGH9jdD+u2/1oQRM3s61eOyxR9F
aEkporV8Y

ZPepp5569NFHRQKIWAXoeitNcHDw999/r08uDvEuolhsCcHp25/u+Nodd95Jb++y1Fc+ca
SIF3SN1QCL

pIyMNELPkT8MZ8pQX2Zz2DDrZw1nzipwRePQIwauULA9rjtWTVqw5/Pz8mFykl9iO/KC
go2npqerx6TMp

KUkbNvwkzctWNNocnliMiyvEF4zxMHkqK+ijKW8+qqWuLVr15LOz8tHXOUjRhUaKl
Y2iC7Sm5mZiQbb

AlwZREH1TiUoXd7etS5E/WAUQSevf2JfeiKHQK2dziL1VC1hYWGbnm3at2/fpEmTGAE
CYNSX/PXr1zMg

TN+I5ieN+joJPd6G6gIBFrPUFz+OW8eBEv7iXnHuAQH+AQEB6pGcU7/6e54rhegW8sY
WfVVvNT/6yCOP

ILrPPfcc8SWZDz/8MLEvx52m virwfZUOEHy/9NJL9957r2Q/8cQT9OGBB+5/8cUX1CNgx
dA1VgNW9JWX

PXvOTDe3OUFBgSNGDgs7Hjp27Ohdv+zasGH9rFmz0I4yoyKZNbbWB1kU6C7+XZ7AA
mbZz88X6eVlwIuB

lwTxtG1Et5yN3CLDHCVHonAyVZCamZ9vfZCXNKBSsjEdfrY3N48czFLcPOsalFWEqjZ
LamNXi175yFl6

pxJUUF713/pTKwVEdLkRkpOTrT9pFyFlbNR1l4E+XAmkvPRKZ6k+0C4TxyFGgwS+kT
Q9iY2NpaQ1tZcM

Rn2dhB5vQ/Wi1RdnIh6WKliYBgGOjdWfMiQkKv13XyXBVYwWLiV4SvMsn/+edIiySD
5bFE+u0x56L5W

ggreeXbAkuEXXnhB71lYXeM/Yt+izlhl+E9fUtXDZOnHrAoL8wlME07HiVKqHEtf8dfnVF
/mFK09cuRI

eHi4/Mqvbd08DqIC1E9Ei7hSs5h6mMvy+bxCeJ2Q4CgyzBbdRZ8AbUAMRJIePywZcUDk
xC5QMZQUvWFL

mhw5sQLkRC4fdFaFlFeeTGIRoANcKts09QOacgmcog8XQX6Z6MMXRIR62WCElnpI0Ad
y6Ib0mRzKyLRe

Ihj1dRJ6vA3Vi1Zf/ClhDbp7LCzkSCBRr7+v72EvL6+9e/fu3LmTxNatWxOJiRTygVFcfNW
CoFqfoS36

yC9b+eU+2cohZfGSL75ofeKW/ArQ11gheNIFCxagl/qcklhtvfTSi/TBSlkf6rX6I2n+qQL2532t
8sr+

rC+p6mGOTiCNiC7iF58QixySZu5YJjE1rJPi4uIopy/PAdy0gPqyqGLBQYLyrLGsv/oW/d0
XIRctt18b

mPxhQqXVAfWUAB3AkATqlHwbckQhLhhHFSQtTVi9dwq6EwpLBIWIieypi9PookXo0q
XQh88f3YwD1KZT

6nXLDOqdSwajvk5Cj7eh2iAND40AAFCpSURBVLGcbG5eTlp6Kq48xvqgURQuPjr6IES9
+OioqCj7776V

dyhSvvKo2LFY4FsCyZctCqcKVoSut3w2bNjwwQcf6NLloxoX9DvejvmEvyhw0Z+r5cirOM
dKolx05clX

86VBBXWqCOSKLTIsu44TIWlgQgEBXrly5cSJE11Iib24crahoaFhx4+NHTt6z55fiHHIRCH
f+vgvZ+fL

lzmTYMWGZHOWNFoaae7CsC5VCbAMka7Rudjturi4MGLE94ShkuMcWEWNHz8+PDx
cdmfPnr1nzx76QH+4

MYmDWfrIoUsEo75OQo+3wRkU5ufnZWVIElqJFw4MDCTePXbsGK5ZwiknfOLoo48+U
gKmIcQEO02wS4Kt

vAuN4KkjFaHrLYvz+rYNWkRQ7fecJS3yr3plfdsGWznIv//+97+6mUqg1abSyEmxsTFhYW
HInhKpPOaL

NH6ZTA4ziVJSTrGRHFFf5vSTTz7ZsmWL/H2BU0igvpu3bAo7HorWrl690tXVJSOqwt19y
fgJY0eP/mHE

yGEEuxs2/ORz2HvkD8OXLl3s6+vDKYMHD+7du3dERMSKFStGjRrl6elp9fHXQT8d1Vc
670xoUdSXkUF3

Z82axfCWCH+rG1a9bIcPH26Pw7fftu/f3/c41dffeXkzvweMOrrJPR4G5xIZGQk8S7uZvny5
UuWLMGx

4qYRYLyz4/c8S+Eqx/F7nhG8p5566q677kLh2H3wwQftnze4//77yZQYtGJ0vcWp4MHm8n
n1ueesTzqp

L522+vbMM8/ce++98gjYww8/pDpzNkTmQnRj1YC8C8p0LfiwYOfOnZLeunXruHHj+vXr
N2/ePASYKZNp

YivSJUgOc8osUyY5OZkoCuWWSWeWmfF9+/b2/a6Pn9/ht95+c8vPm8aNHZN5yqSc3CzE
mN3Q0GAkGQFO

Sjrt4jJh7bo1Xl6egwYNQp+IzBYuXEgTKATbXwn9/A3Vl6WMPOJEuyTGjBmzYcMGOs
NlSgGnwcz6+vpK

jIsYu7nN7d27F+kvvvicZbEqcglh1NdJ6PE2OAu8DC44Sf3Wur+/PyvuNWvWJCQk4Kydqb
4it7Lt1KmT

pWYqAejxs88+iwRemPpydRf8Kwu0eP/99z11/bC/FXk//vjjLAhU917t2PFuZ6qvLUVIHeGm
vOvL7Myf

P5+c8PBwdAJZITJsS6gvoLVC37596SpBMGlbfTdt2ki8i/ZNnz4Vi4wMR2UJhd3c5sgvCRI
Br/tpTXJy

Ion169ehvllIJjt06FCECsGw/hhM62VZJaG3v23saze6a9euadOmjR8/XtY9zoRFMK/2TZs2L
VmyODY2

pkaNGkuWLFrx4/KePXuw7klM1H9fuHQw6usk9HgbnAUek3goNjYWxWW5HRAQMHLk
SNI4ZTKd+AuDWsYI

fFE1INCUB7p45JFHkL2HHnqIo0rwzoGuVz2icsG/MCj9UY9ev0I3nn/+OZp+6aWX7rmn4/0
P3P/SSy/e

d9+9JTpTreorT1QhdRs3bkQP9u7di2SOGDGCzAEDBqB8Xl5elEE8yLEk1wHJYTIFFEWI
HBkZSTAnDzyL

+vr5+VIG6VFCSR3W13qoXUFLYLb+VUGrTH6+9YhWpsN3ILPNYD87K8PBMnOys3N
zcvJyrY/4KmVTNZUL

IYj6ypPG1MfqsAJ49bIQuQC11ucXhxZpF2RxQ4Xyxr5sgeZKoM8sghzpD4V1VoXY5Uv0n
wQrqGXL3BkM

1kDyPaPqoffC+IRYMqWUsofo75OQo+3wVlw/6O1hEF4YR8fn+nTp6ekpOCpwVF9K+/X
zpeiX9e33tp9

+eWXn1Q8/PDDSB0CTJpDaPBzzz1HovLq+yt/XV/Ulz488cTjdOyhhx5EhIkEoLvKp/74Y2z
/rD/vq9cN

1fPr+hoZ/z179mzbtm3fvn1Llizx9/dft26dp6dnSEgI6ksBVIF4cMfykhCYUHQXzfbz89u/fz9b
avDw

8KA2T08Pglr5DcETJyMJgiUtH0KLi4sh3rK+q+NUFJaWnhIfr3+WX8XSFgTBp6Kjk1NTE
IOSsdPKrERy

UkJiYvzphLiE+Nj4OBZ0vM6K/URfcaiKalkl8JrkclhPENBXAAUoRmGWefSBTuuKzoU+
vxRUCKxO6AYj

RrUCN0WZ6NOKsPtDDbpEhTieQityCQr6yKjGyPe/RkYdj409FXUigqnB5LNnaiUki6GLH
KO+TkKPt8FZ

oL7c64REREIREdzbUXhzdo8cOYIDwjtLBFB9zJ8/H6mrANE2SSOHaGHFIC1du3bVOxc
K6wC20qgk2Eo3

1K6VgxKrXf2TiG5ubjqiqQYYKCIkYDpEYkmzlUiRBPcOHpxd0iK3JOw0p4hYMtdMMR
6fBLt4f84KCrK+

WIKhxVvBefmYZBKB5eRafxMIMs7NyyH2lZ4I5LOff4bShXkOJjmORnxnhXjlQFVcjlysh
Q46GbkcWidN

Ao0kkJXOIb8MtGHLwiZUFAf95J5ZBZIW5ObkZGK+hbNi9hFLsBGfZ2EHm+Ds0B9W
WaL4qK+sg0ICAgM

DCRBYIQRVO+6VRdz584lohWUsJ1FVA04JDmVUD8qQXUHAbbaFZX98yvWX3lVji7jT
PXFIyMDaWlpBFVM

GVtyuF/krUsOMZUET/h9mVYbW1EkmAPUF1QEaIWAor7q930pJR/5tT4Ijtzi5e3P+7JLZZ
mZ6crvVy8i

e785DAdbhlcQTZVDNjK8pdGHLxSmlfazsqxPf+XnZxWeycnOSSs8w4okOynJ/ryvSC9m1
NdQFejxNjgD

btpC3Ir1ZQ4K3Hdi4mmiItJxcXFSPt09cSVqNf+rlvMVshfvXolubZA0x60kkDe2SCJbJ4PC
FiV0IF7U

mWLvPLNS0eNaPRBoZWVZHx1iA7zxNHLlimzwyYbSwqK1BfQ2jVr1owdO1amWOaaJ
dfGjeuXL11M+XHj

x+zZ84s4dPVt0o5Y6st/a9et2br156CgIESCF4a0LrEvCdQg90yhGEJdZOcX+8rrjRyr2Uogp1
cefVpx

yLdekUWx7+Lfi9etW4dT0j9SwDlw0w0fPnz79q2sfuLio8eO+2H6DNew40Ffff3Z5i3rE07H
qFIMqlFf

Q5Wix9tQ7XDHWrcuQU/C6Xj5Dv3U1JRp06Zu374dH41rjo6O9vLyobj4o2rC19fXUrAilX
3kkUfkT7zw

5JNPyo8rPPbYY6RJOFN9EVd44IEHiHpt7ZdukEngW+LDx1yIvqRqY+/e3S4uE9LSU7Oy
MmJjo8eNG0um

h4cHYTcJFk/16QS6gsoCZbp16zZ9+nSUmNhX1ljHjh1DULt0+S+R7qDBA7b8vAkx/urrL06
eivr0Px/7

+/t++dXnk6dMWr7CPSMjbcCAfnTD29v7k08+2bVrV8+ePb/66quIyMju/+vx2ZdfZOfnhrk
FBI1n30L

WIRCDIUA/UT2EGCwBbgy6PMrjT6tOEp5rTeTKcCLf8iQIQsXLpQVhplzNJjHESOHS9o/
wKfPt//r2bMr

sW+P/3VJTknAjPoaqh493obqRUuv2ImTkbjypKTTy5a5Dx06BL+Mj46KiiIwcsIzz6DIS2kw
Onf77bdL

mu3jCjQPVeAQ09RX4l31vZlvoF2qM68+/PBD9913X6d7O9GTu61PHKG+OvAFfTHVAtJg
CQDBqK+vz7Zt

P5Nmyvbt27tkyeK+ffsyTQRMycnJeG2Kq1OKQaaob0hIyKRJkw4cOLB27Vp2RX1DQoJXr
fpx5coVvXr9

78DBffPnu82aPQPFzc7ORIZJu7svIQ5btnwp6uvq6rJz1w4vL88vvvhi06ZNkZGRLi4uy1csT
8/MXLdh

fWh4WJZ6IpdO0GPb7FdbxUJhXadSX1t69QEnQrtsP/zwQ67ro48+Ik1/1JEyoIc2OqsqGDlyJ
POr/tae

SbDr4jKuW3fru1x69e6ekZGcm5dl1NdQ9ejxNIQv3K72rVsQFBSYaP1OXMwXX3yWmpr6
3nvvpajHnomN

7M/7ymnVhOPXXSG0Dz74oKTRPPlx+2efffbpp592pvoqrMXAgw8+IPHUE088wTqgY8e7
MQ516tRJ/hIs

RTt37qvwplrQ409sOmbMKPk4UGZmOukDBw8sXryYydq6dSu3DzpR5mSRSRIUt9i2KdP
H9m11XfPnl+o

tl+/vqjsps0bxo4d3adPL8R+5szpu3fvmj59KhocGRk+cGD/adOmeHt7BQQETJ48mZdHRET
E8uXLCRD7

Dxwwd/48eUkJ9OOPqL5Au7SelZW1bds2tnSmvJ6Qb6OzfjXr168fOHAgS+HRo0cmJcWP
GTty4sSxUSeO

9+zZzdNrf2paIkGwUV9D1aPH21BdcKOKWfctvg6TT5LEJ8Ru2PBTj//1wJkmJp5OSrK+B
9gJv+8LQ4YM

kbd2EbkHHnigY8eOL7zwAmrXvn17dlFfpI6g08nqS2eeeuop411El/D3xRdffPa5Z//0p9uffPK
Je++9

F0lWndHqyyXoi6kukIO8t95+s3//7wIC/IYPH7rl502Dvx+0efOmESNG9Oz5v+TkpEyHT9+
WgEyRW5CH

rdiGhoYSuR47dkw+76vCa8uVk7AfvyIhf+sl6iWtnr06S25esaAwr7AgPRuxohMFGbnZpDPz
crLzc7Gc

vNzs3BzM+ohr+SB1OIG0tDSuhbTOrQQE/YLePxe6tEN50qL6pGUYgYtiPUpnHEtWBIW3
hd4/F2r87HtT

0mcCjvgxBYS/mZmpObkZ2Tnp6G5cfHR2jrX8KjIpFzFj1NdJ6PE2VAvcqPZNa5l8jEQ+zR
mrfrcu4XR8

XFxsVFTkiRNRJ4o+cYQzEsckyG55VKZMCX788UekzhG0TVRN7x dxweqrzz8XunQpUF
x1tFgBx85wCXLh

54TxLD2kZYHU5SJ4mCySimatMCoqwtPzINOJLILGtpiYaGuOK1Rf5DYwMDAoKIh1FY
mQkJAjR454enrI

3/7j4mKSkLh4nbY+1JsQy5ZMXiHqhWH9pBKZ8rv6sfFx6ZmZyakpaRkZqelppK2crEwsIzs
L3U1KT03J

SEvhUFYmBTIyM7H0jAyx8kDqWBkkJiaSAGRYHygHeQhcPhQEOrd8Ki5Pjsg/CbpBMQ
onJCQkJSWRybmO

qDN0pqSB9IX1R330N5GRlw/9kpDRzspKT0pKOHkqEgFOTrZ+2CoxMV599ItZFrV4MerrJ
PR4G6qektKL

cRunpiWju5GR4aGhwcHBR4mrDvl44dz3H9jn7++3efNmeQIrtugH6YD0OdFFz0W8Ak14ue
h3A60nnYp+

bfAl/aN+ZyEAIWKgs6of2iLsJuHYOsiPLgCHjh8/rq+8yrC+msI2dNE2XDCzJg/KiViKcYo1
zeWrb7SC

RFhYGAnCX5phjRUcHIR466JWYUvp1TdbWe91Y5Iv33WFPOWTeaYgMyeb1xCHUX4p
APJgs+OTVIBQ9Jyz

2qsIyrDukHUJa5TKnFLd8PokhCWhrUAscRskOHtvMtpFH+cVrCUyW/W3Xihk0mVqLh2
M+joJPd6Gqufs

HW4bNzYBCY6bVbZ4c5y7+rXByliI4+iuE37jSBg3bpyOIosoHYyyi9TpHedSoicCoiuJMW
PGVLU7trGm

SWJfEUJ577dIEYvNJjqhMsuA7onWyqIHAZZnnplitseOHZMmVFmqsrbi4IEctSuSDPpoYf
Goi9zs3Bx0

V+87wCE7V8lWRci7AqidowA7B91FhWMO0TBllxz7I0erCGqzZrBotFiC5DD+mFJi9Fik9
4y8D0E+QyU5

lwJGfZ2EHm9D1eN4h2uTd565n4muEF221nfencY/x0RFRfJfeX/31R6ofHS5SpOWl vbWW
2+JmInUIbSl

Ne83UV/phvxl2hFRX7qdkpJSDYGaPV+Woy4yGytIKhEDlaU6VQr6Zr3vrL5qY9q0aa+//n
pISAi7osQh

IcHrfl07bNgQP7/DLMW2/LyJF0ZGRtquX3YEBx/du3c38e68eXN5bZDDQu2Qj3dg0NHgY
6Gbf96CSCYk

Jm7cvImOSuArsW9uYUEeppYMDA2mu3IuKP4bBr7SqGz79OmzZMkSdJf+kFMCVbyqo
DZ9S2I+h70nThy/

YOE8WfEsXDT/X/96e+vWTYO/H+jq6pKcnFh0yqWCUV8nocfbUPUu8PFJJZCdE+eim
K77qc1kyZN7Na9

C+p74iQ5MaU/caR9z7nQpc+HX3755Y033rj//vvtXxgkfe+99yJ7nTp1IkGmk9X3GfVLD88//
zxpmib9

2GOPSduxY0d56grp9fX1FcG44Asvh2LzJeukuXNn9f2uT2RkOEeR3i+++Mzf3xdP/W3f3kxf
meorvaKH

BLtCUILS6NGjIyIiotVvHLENVb/vO/KH4b6+Pn369AoNDZ4xY9rAgf0jIo73+F83ZICVGS
+MFT8u239g

b//+3/28dYuHp8fb77xz0MtzxcqVI0eNOhUbM3zkCEtoi3psSW+h9aY0YizqC7pP5UMZzvg
N33mWFtlu

3759vUL6I0erjWJzzT5LHJY7pLkTV636ccCAfr169WBsevbswaqo6JRLBaO+TkKPt6HqKX
GHW8b9nJmZ

jm/F5J3n+fPdSODfVWa0/YkjcYWVR5qsPJmZmZx17Nixjz76CK1FaOG+++576KGHHn74
4eeee04+9uNM

9aU5to8++ujTTz9NgkXAs88+S3+QXnbvvvvuzz77TL1naw1ONahFifkqlDeBd+zc5uXlkZq
WTJxKtEow

Om3alJCQoB07ticnW7+DJNg9IQH0TaSXYHfz5s0bN26UPwDbse/adWuYdJrgQjdt3pBwO
m7UqJG8PNav

X4foRkVFjJ8wduXKFWnpKSNGDtvty8+YDHge/G9A/Iztr/sIFPfv0RmL79uvHVhoF6bSEw
oyLzj0XdPW3

VV+B1pcuXTpixIgvv/xSnlvWB6qLYnOdmJiA3DIXYcdDt27dMmv2jOeee+bLrz6Xj4TJo3
bqlEsFo75O

Qo+3oeopdofbRkSVkpKEcauz4h78/UBcLaGwk9UX5CyEYfny5dOmTevZsyeRJbEm6ksM
+vjyvjv1Lfb1

UtUHWivq+9JLLzl+/pj0U089Rffee++9xETrPUC67RT1pfq8Y2EhgwYP4BiB75NPPo4v7tL
lvzNnTidm

9fA4kJaWZnfAMQGcLOpLfLxp0yaCXUSXXfupq/DwMKJnWiHeJcwlPb29kQASCDtvD
xoglcId27d4Ud

PxYWfnz3vr2p6Wm+Af6BQUGjx40l8JUWBen0H0t97eYk2E1OTvb395fJreb+UK2eZWzbt
p9dXCYsWDiP

LVPAvo/Z80t+fs7Qod8jvayajPoaqgU93oZqodhNLsaCmrW2PHVF9LN5y6b4eOvPwOzGx
8c74dsTBbw

W1rs1asXYEbm9uBgweWLXMnFkchvv22N5Lwyy87Z86csWjRwkWLFixatGjIkCF9SuH
4xR3nhML6NAeo

IsoXLlxIH4h+vv76axYE//3vfyMiIkaPHr1mzZqYmJh+/foFBwc7qm/1+GXH+bJ+54C4c+7c
WQEBfr17

98QpkzgWFkp4Om78WHqRon5rQbD7QwLoocitfIUZ2zCFvPPs7+8nf0KmCWmOyh2atqpS
h6gtX0qiuzl5

1pdKigmZOdnym76OCfl930r92q1CPqtDuIkE0g9ddVWjGysLvJC8EyOty/NW5KvP5Z79IK
/AIRt7lxdz

6ZJIUITe+vs4Y4sxnGzlfQ5uTNVZYAqsFwODn3A6Ts3OJYRRXyehx9tQXchtbN3JGC4II
yON+1neeU44

HS+/uBARER4ZGRkdfcr+nmdBXHnF6GYc0heA/OZdWnoqy3w6iTfDyEINSxYnJd6KLZG
JsqaEh4hnlAUB

kBMSEiIJyRFSU1N1Sq0eBFXBuZHxORcUs0y3YY2q5Z2Zr+TkxCK9zLONWE2KOaIrKo
p9UdxDhw4dOXKE

1UNQUNCePXt27NhBTlyc9UFejHUY9YsRB6u3QE4hA7w8oqliTlfgGA4/FX0yMzvzdHJS
dFwsFhMfl5aR

EX86AUIOSUtlayfYJqWmJCUnp6UTmVswaBUgH/ZITRcnIFFN6PaKIy6IDrCOkefDExIS
5KO4uvelkFME

ybGrQsJIWwHqPPnUL+1Y468GIAFj2JlidpmItHT6Y/30Mumw46HqZXAjYdTXSejxNIQv
Wn25vbnhuc/x

s8HBRzH8so+Pj7e3N1t/f3/5vC/eim1S8Z86x1tUleKzbGgR6cXpKMN3JylLjow8jgPCPyepry
OQz7my

YqBvVQj+D0jQMTwj186KhByumr6RjwSRJhpWxS2kfLWBHiSz8sC4aqZMEvjo5GStrkwL
lrKktvuhxVF8U

hYGI24TvhLxsqYLMkJDgTOsLrazVWNFZln+XCiwlDkclrbYIF1WsglhrXWQw2d8Lxi6
KouYEhfiNGhd

EtIBol4VnpYN/bQpnaPGvjJQ0lpTijHgOblZMgWyxmLYicmtPlk//3BS/RzkJYRRXyehx9tQ
vWj1xdvK

O88IMJENYU1kpPUoLB6ZLWt/+xNHv8KznAOpvxw4ig/Kw/8oy0tMjMvJzczNyyJAdXB
YVdYZR7hG3G52

djZpuiK7pCWT0UCVZVgENSrV0hMRPK4UL4w7VhpZxrghwGxV+bNIDXRMyl+WEfv2
7VuzZo0S3RDEmIn2

OezDImbFj8uOBPorGbbO44ppDiUQSfb1tV4JvFroxslTJ6JOnkBxkVtV2Hox6T/xqvSvga7K
YJa4ECfD

QM+fP5/FFndBjvrtijLRpRWlc84TzrWMQXZzm2NPxLqf1oSGBpNasmQRqhwXJ78w+Fs
OjpMx6usk9Hgb

qh3uXtwcQpIZFxeLPz1xMirs+LHg4CACO1wzERLO2v6mSdsh2uhqfjW6OgVNAI5PnEtm
ZlpWNpYqhodP

SorPz0d60QNU2XJVUkmVgwTQDfQMibVbUXKLGuXjdkdSOFbTmJQJQshkFX3gxBJC
AmLpj4BUkF+iG9Ix

+inqS7w7bNiwadOmyfyK+jLpm7dsCgkJQm6Dg4/OnDmdBRlyiwbs3LV95coV1OPnd5hDS
5cu9vQ8GB5x

HPX9aeOGyVOnZGRnHTrsM9F10snoaHTX+gNmkykeS1qGyZrdCuEqkDpGHuTPonJpV
Y5urzgyYkCauyAt

Lc3NzS0zM5OXgep+GegTFDrLAX3g3FDWXkrm+/v7pqQkDRo8gOlAg1PTkrt0+e+kSRP
R4P79v2Pe1Wug

el9svyuM+joJPd4GZ4A3xBHIRUefliSKiYmeO3fO4sWLpkyZgszgfci0f10ff6T8iYU+u0qhfh
wua1b

t37++ecqJ2/Hzm0DB/WLT4juP6BP3+96ZWamJibGKt1VsVZ1OiAkgO2RwIDVq1euWvUj
u4d8vKdPnzZ6

9ChCosWLF0+ePBn1pQw9t8dHnVotME2Mz65fds2YMX3nzp3k+Poenj179tq1a/v16zdixIjQ
0FDUgvwS

3VAzdlZ9o6KiBg4c2Ldv32PHjiG9iDE5ERHhGzb8NGzYEPz7Sy+/gPqOGjVy9OgfiLRmzJ
i2ceP6mJhT

w4cPpQyqzKGdu3bs3rO7a/dup2JjXCa79h80EA3+fthQkVvbJA7G6FCxRUr5yGAiuqK75Oj
LqGqkuRLo

Y2pO2bq6uq5bty4rK4uB1SVKleWF0jmVRs5iq42lz759e1TmGVY/Y8aMk19f7tz5A/Xmh3n
m2VAN6PE2

OAWcBY4GLwxoyapVq4YPH75x48Zk9Rs4ZNRPPItNessnaqEbor7bt2/v0qWLyskLOOL39
X+/2PLzBI9f

7w0b15alvtXkg6zvsuA/Io9+/fpu376VKCQ2Nnr48GH9+n334Ycf8kK1P3EkgsEWqmlwgCb
YogQLFixA

OCXdq1evvXv3Llu2jKWS/MJg6QmSHPom6hsUFETsO2nSJB8fH9ZYBPGssSIjI9auWxN4
NIDyI0YOI/Yi

3iKB+i5cNN/dfUnY8dBx48ewCjlxMhJV3rFzu7fPoe8G9E9JS509d+7QEcPZdZ0yGcWVRn
8NdFXGs8SF

OAcCbrZ0IDAw0MvLa9SoUeQQ/lqzWxYyvILOuhC4Xvstgzyk959vvREcfHTr1i3R0SdZg3
7zzdfTp0/d

u3f3tGItkpJOq7/7/gaD81th1NdJ6PE2OBGiHxwx4M2Zgi+++EKeynH8u692MArJqVqkWt
wc3SD2pUsh

IUFRJyLcly1au27lxk3rxk8Y7Uz1VeHFmYiI42wHDuwfFBSo8gv69OkzaNAglI+I05nqK/W
7u7vv2WOF

RKxR0FGGiykbOXKkPCVHr9T8FOuD5HCuqC8z6+fnh2aztCJNTkREBHKOQy8kXrXK5
x8J9M/KykhLT0EJ

CHbJQYaTcxPZcjlJsk7JzM7NyctI4P4IKg3PTPzyNFACXZ/PdJbtrnfuUi7dIAXPyPDipBZ
Fkm2xrEU

6iSNzroQuN6z7zyz9OH1hu66uEygWtLxCbHMhc9hbxoxT10Zqgs93gangGzgX5KSktA84ic
Ud+zYsSgf

rkc8tXzbhnhDGzm3akHCMjKsx0zc3ZceOnRo6dIlxJo7d23fvuNnRJfA90igr1LfOMoqq3bvj
L87eSqK

4A8pGjNmFNHw3Lmzs7Ot77ufPXs2HU5MPK2KWepbfSNjQ007du0i9t25c+fcuXOZI7bk
h4eHr1+/nsQ5

1ZcppjCyzTY0NDREQZDn63tYnmomruIyMbQWDCAyMtJIYxQgzVCQaf3pMTMjKSUP
TOD8Dc5NQU7nZSI

pWWkY+RjmdmZ2rI4Wf+uICuGCqBAivqED9vUos/tVlbKlxTKLE8mMIxg94G7g0OSKU
hhQWcpdNaFwFon

1TYGAH113cOWXRIJp62PADAFter8fBzzqWf30sCor5PQ421wCrm5uWlp1odXYmNj8c
hWcKRAgMPCwnDW

+/fvpwwKLY/ACIhN1SLeBF+DoXIKXAEVsYSksFC+W4B4NIId1AlviA/VmnWXqsaOzTx
79CqiES7OMPrDF

2cnjvkDYp96LtjrGITqQk5PNksU6pD4ZLOgBqmoYfzw0CXSU5oCE6okVrepfJSIDOMuu
HLJhF0R91Z94

IySqI94FSfj7W79kRVCLAGNcndJ6a9gZBLbWs1BW4GvtUICPUbI+FWP9pUBgVyzwK8
M2QmTKYEwr2iQS

RT/PCcW4KLwBCbne8pDxYRz0mef6jouKy9Mc4yyZsqgC1LeCbsuJgmMOp1RwliNSXpD
uMZ6yVesBa/zV

BfpzDawJ7PQlglFfJ6HH2+AUuOHIMxX4ZXHHEhL5+/vLT68TZuEH7XBE4JQqh6gLj5+
WnkJcxQKfLcaq

Pz4hlmBLfeQXSyEYVbtERdbXN3Ci7lPVwDVa33ggQR7N0QExMtlVTafQVQogtegZ3pP
T5BJ0FdWANEGC

LS2islw7qsAutwxHWTyRrlh9o9UHfJlr628M6ocF1aN21pvPwcFBCKQqbomuWnywpJD3o
qmNNYdlSo+t

r5tWCVkh2aBVLJ6y5USphNMdrNjbJ+UhmiciRIcXb1ToHVplwS3hmQmJvJqzCKnTKTb
gs761UhPpHWp

WdJ0jC0vA7tv1whGfZ2EHm+DU+B+ZrHP/Sx/BWQrgREJHDo+2jm/74vjUoEvsZT28qK+
eG2VeQaJUQWt

P4mpMtUHng53x8BYEsJWYj5QkmPF35Kme/KrBrajda7SHCLBVtyxvct8sS3RH3HfZFrva
ajP+65atWrW

rFnML0suW32Dg496e3tSfNcvO9SnSwsK6rVRl29tsZUrV+zevcvT86B6P0CPCQnVGtKlnx4
q0t0/jPqC

tEjrrGYYJZahIr1y1DmEhYW5ubkFB1uf8aVpd3f3TZs2scaaPHmyLJik2KWUDUV8nocfb4B
TwMjgX7meQ

tyWHDh2KAAM5bO2nrvACNspPVi3UamkbNnfurGNhIZImceDgPlqfO3fmzl3bsrLSCYW
Rw6LC1YHVrl05

Yh8SEqT6Y33r0M5d26OjTzIYo0f/EGv9InI0XWcAGUa5DJATqxxRI19f3ylTptAiOQzZkC
FDWDyhEDNm

zMA7E6XRAfLIFE6Raaob3h4+Ny5c7dt2+bp6Uk0jAbj0I8dO7Z69co+fXohk716/W/Lz5s2
bd7Qv/93

vApG/jAcYe77XZ/xE8Zu2PBTSkrSzJnT9x/Y63PYe8iQwQjw317U5L8YcOG9OzZg6mRV
ZQdAYuJ+uo+

IYXdTwI74vsSV+Fk6IN8cdjAgQMZeXaleyAFqhXcIHM9ePBg0vv27du8efOyZcvefvtt5vqb
b76RZ/0u

KYz6Ogk93gangI/jPscjo7X44r59++J0evbsKepLBGx/4oiSNtoPVSX4XCvGxU2jr2+88XfSK
IGvr8+g

wQPCw49t3Lh+2rTJuPXk5EQVadGHKvfOOFbq1EZPCP6wN9/8x46d2zhMuyNGDgsKCh
z8/UBUuVu3rgge

fpkBlGGRWqoJqZ+oyMvLa+vWrTTavXt37hfc8bhx4zw8PHbv3p2cnEwxRlNOEfT4FqkvjB
w58tNPP/X2

9maWUV9ywsOPL1+xbOLE8dOnT92+fStKjL377juJiQnjxo8hzF24aD41rVm7ikkZO3Y0qx
AvL48vv/p8

27afw8PDRo0auWz50qysjM1bNiLVKiC2pk8EWAW+1Yp9pfOyIuRk6b8zoXWdUixdujQkJ
ET+Nct9K40u

qtBZCp1VafRpCnYXLlwYGhpKgmK9dOgQM/7yyy+z26VLI0vQSRr1dRJ6vA3OgPvcMnw
64osvnjFj+rBh

w7jDJSRCiQMCrM+ACto3nL9nqQRSJ3XnE3i5urqQEMdNIEmAha9fu261o/oWnVKFUK
GWXkwkBLndu3f3

qlU/0hMK0Icjpgf6ff/4f0sSIjupbPcNSBkRj+fnWm8Dvf/A+2w8//HDdunUo6IYNG+gPnSnRE
2vCHNSX

OSVQHj16NAkWWOTIg1dr162JiTl1080dTpyMREq7de/CsgNlnTLVdeXKff369WUomIU
e/+v2xRefbd26

Zd++PZTZ9cuOsOOhHDp5Kuqrr79gXcLQMxFFs0PL9hvR1ijRvfKQEJMEapeWlsZW8p2
M6rbF8uXL//KX

v7i7uzt/KTBt2rTXX38dAXZxcWFn3K1bt/79+6PBLIu3bduWmprq5P785hj1dRJ6vA3Vzlm
xOXnqRLT1

e4Jx+/btHTx48N69e3HNCDDeucz95Xzq5RCeeqH+GnSpImfffbpsbAQQi4isE6dOoaGBuP
05ffGnaa+

ov3e3p4zZ06fNm0KwR96/Ol/Pv62b+8NG37q3/+7NWtWO1N9kaecnGwUkXE4Ehgwb97c
9evXjRgxfPXq

1UOHdu3Tpw9OuTz1la2oL96cZVa8+tkMtuyqjyEdl+WFEk6N/ZAzF27/8ZspkJFhiGRbNG
7y118rwSSq

r0IkfX5IVxnM3/adZxF+SdMlmWKgS6WhgI3OUuisC4UaaFE+2y3InxuAe5NDkr5EMOrrJ
PR4G6oLHJwY

3s0y/GI8QiwRjPrtoAT5hUEICjpKTMz9z20PhCOZRUhOmcjbdMQxFRcrgfJ3udIrnDi24sdl
eHz8ODnK

9Vu+2FF95RMs0p8KkMorrZFnh0VaVG1Z+e7uS8gREVK9wi9bv6dr18xL17qSc+E4PoI+U
DYctQzdRfPo

jGzpwIGD+xBFpX9WT6w+qqdhS1+j7LKVv+vLB3yDg4N9FLLr5+fLwDL7bFPTSCXliwF
LUT+BhQaTYIkW

GxtNGXLi4mLIpDC7ci6vIox8duURcXIKnHRKSnJG0ed9K0BeYywjWBmwlfHRx8qBU0A
Ks/sry8shppVp

og8Cuwge10mCoyWQCgV2KQMUltpkWwFSiZxCK8BQAgsjtrwC6AmHWDBRWDPHT
vGLMvkXiIY9XUSerwN

1UIJdbEMRcFphoQEHfLx8vX18fLy2LlrB0EwaQR4+/btuAZ8kDggQXmYsuGoeBDbE1W
S8PAw1OHkqSix

EycjMdw9jp7YVw4hCRSzHH18nD7tXNj9wXPJnzkJ6MvHepBKLD7BaotGiQxpHRWhMz
RNHdHRp6iNMqGh

oVRI5ZxJK7rJCnEcn0oMEUMtZv3sLkZ/kpJOi8IxGvSN3aioSHqVcBrXbH37R3mwokKA
gQgYD86WHPSY

2DfwaKB6MdirDSuNurPcUbvyw4JgfdGVxL52lCxpWRIIDosDzsXkD+dW6JyTzRpCVh4V
QxmcAGeCzqoQ

ljGy+qIM5VBxeTIBzaOAFONy6AldQvzYqiVRMeQUQXaph1ZKHCoPKUxb0py9VGJLv
qTtBOXZ8mJj95LC

qK+T0ONtqHpKSy+RU352TgZ+PDb2VJz1maOTaF5ExHGc7MITJ5Ar+5lncRACbqJqUX0
7G2iKK8fEidtu

HXCDOIXNFH3+1VIX6ZJ00sooCi6RR0lbWUWZVQ3VitGKNQ5ZWdYHgUQOGZyiQTV
DKLFIknSZWKuLlpBe

Css7z6wGWEZIGG3LrVw1uzL4clR1g8kCPWUUUKecTavy8uqyzTpaeRhJXhLVNp7nBkV
07ADqy2vemucy

kDKCzllorF+BXYkkRIOBBRw9lPQlgJfFj6HH21D1cA87+kT8KY4yLzcvKyX1dFx8dGzc
SbYSY9nvQjs+

dVX9WN+2obyN9TFfB1cuPbfEgF5xNN/6RirirapfBxRBB6w3wOkTzSFy0j/aJf6QyAYPyO
qEBN2QIEaf

WmVwdfrijOgUFuSySivpgCWSh9WvH1iTKm+GyrYz6Dh06VB7l7dmz5+TJk7kEYt+QkO
Dt27eOGjVyz55f

lPqGmi1pp1pal8q9vDzIl3RoahB4eBiHbO0nX82LfEEHp4sxcWIg23Ngzb16Aksmt0pkrPL
QHNAuisuK

5J133mHESEtPykSfqdBZCp11QWzZssXFxWXVqlW8qHCJY8aMGTlyJHfiN998s3z58IT1
O5KXFEZ9nYQe

b0PVg0ew3SKm1Rc/Hh4eSji05eeNPf7XFekdPfqH7j26EgHjmg8dOmSd6SQnaLWCBkycO
F6lC0kMGjwg

Ken0wIH9ycfdE5QXXUj1dkm0h4amTHWV33ojp1+/vuPGjTty5Ei3bt02bdqUWPT9+0iFc4
ZowUK3AQP7

njgZQfrAwb1DhgyMiAibPn0qoxQXF1vxO8+EuRAcHLx+/frp06czufh0eSs+LCxs564dLi4T
goOPDh8+

dPacmRs3rh87dvTixQs+/c/HDD6LHnYXLpo/ecokBmHv3t379u958aXnx44bvX3H1nHjxyxf
4b7ix2VF

82IPhaRl19oqYaoIdA7pZVQdBdg50Jb00I4yV6xYwfikpaXJ/JaJlBR0lkJnXSh4QldXVxL0at
u2bbzq

evTowS4CjPpKVy8djPo6CT3ehqoHj8BN62gIsGWhx4JPnIzMykof/P3AyMjwGTOMrVy5gl
gHB+3l5YUr

cYK6EGtK8BQQ4Ne16zfkZGamz5/vdiTQn155eh6cN28usW9cXExRgFWt/ZFnsAvCjoeiPZ
u3bKTRLT9v

Cjx6BNF97bXXcH9du3Y11lSe1jnjI+/x5iN7Hh77CwvzUtOS0cJly5fOnTuLpRLBa1KS9W0
b+oQiJlet

epbupL+//4YNG2bNmhUVFTVs2DByEJjw8OPrflrLUNPE3//+Vze3OfsP7GXpw+6SJYtQY
l4VI38Yvn79

uuTkROR51y879h3YO2BQ/5i46FlzZo6bMJYm+vTtXWC9nGiuhMmLjcTvWn3BGi/6WqR
ty5YtY4GF+tIZ

XaIUUILQWb+a7Ozs0aNHMwiE3VJz7969u3Tpgu6ypTNGfQ3Vgh5vQ9Vj+8GSdvIUrth6s
Kh3756hocHo

HHEM8Q3q4vgbR1JL9SHvahJmEV2RxmbNnnEsLGTo0O937tpOlxCDhNNxzIFf/tHc+x/8e
/ToH4j/2KUP

fn6HV6788c0338Q/fvnll47qW21DRJ1iVn9YfAwY0I+0PA0eeDSApQkRKgnUNzk5qbRflh
6Sb33e6MSJ

0NBQVITHjh1j9UBQFRERwVWEhATLz9hRnHkn9mWct27dQriPDLMeYIL27PmFVIDfb
dt+JhERFb5125ak

1KSDXgd/2fvL2PFjok5GZqsZLEK6TX+szp8X1TOS5wEKx9Jz1KhRy5cvR32ZbgawTGR4
BZ31q1mzZg1N

E3mjwbhEFxeXLVu2pKSkDB06lOILTk6mP7qjlwZGfZ2EHm9D1SOusAwT9cWzo7v42RE
jhxHoREVFOLL9

5W+cND1z5vTVq1ciA+jK+AljyWRLGt3F+7Orul295ORke3yTYECQn1GjRiJ+Y8aMIh4i
XhwxYoSvr69z

1deqvH//76ZNmxISEkTC29tz7NjRpOfPdyNBb3HQdENOs5FesUV66XIMjPWIZiCPXpMT
Hx9P7EsZBpZr

RHFhNXWegeCrSyGJJ8CFMstyEnPSi9gNzcrMyczR/01mtg3v+hv1UwWRkmVIFyrVCBL
md8w9qVF2uVC

rGtR33lJpsSaJCTfadCZsLAW+SsvabZ0A1gW2J28RDDq6yT0eBuqBTzIWdEVw70ivWHH
QyMjw6OjT544

GZWSkhwXFxsZGYEm79271/LHRT+7VnlwEILerxAqV41YH1BRca31XA99w3Erj18oYT
FliMIq0eirKeu

dBuV4LwKU78y9XyT6k9w8FH6Q2cQCHbREraxsbEO/a+O8aHOItOffraLiTnFgomOSQ+B
NF1CT+mi5NiI

bLBINoHgKSgoyMfHh9UD2+Dg4MDAQH9/K7pNSUliwJOSTstjd+xirMkkh9ZZ+mSq3/d
NSUtGdE8nn45P

jGOLxSXEkpmcmpSekeZoMqdYJT/vC6whWBawlV197FxUvqRQZnlegmxpmviSpRXdYG
kC5MjR0jArNjpL

UV758nAsz8IliZU3vaUnrJmQYRIUoz8ys5cORn2dhB5vQ3WBLxbT6osjEvWNI dgeFBR47
Fior+9hn8M+

u3f/gqeWXxjkzscplPAvZcIMSvnS/qhMpLzlatNTcetp6SIIS2xsNH4fPcDvk6Z7IgaUZVe58j
TqliYq

xm5FBBL0gTKQr7awfluJzqD0CB6NSq8QHukPlnAad5wsHwumcvoPuo5zXbL1EleXbPdE
HygbPLJISBe9

oifIrfQEY6IEJtXgqakQp4zXRn3Li9Iicw158eCUJMFZQHQFERHhqanEzVbMqspaLw+14r
GqkkyC2rz8

nBwV9Rbyssm2btU8/dPLZ9Iy04iD88/kU0CMwpgd+8rgV4CMBisOLpvL4ZrJlDVIBcgKhq
WVVAL6QDmc

s7xk0ge5Lgoz3HRG+lYaqVCwd39lf2QGyWc2ySFNJmVIsKU/zg/Ef1uM+joJPd6G6sVyr2I
4WRQFb044

dfLUCRSF5bZ4Z2I7Ly8vfUb1Y7t+Fclpt66gn5a7wY8jh9J5iUGrFbpBi5JUW6FAyXMKiw
BEgn2nuUI1

JnZbau5yrI8/SbxLX1n0VKC+xE/MqUgvEAdHRkaSIJ9J56IoI5eMifSyRT7tCBuQWNT3W
HgolpSahPpi

lujmW6rMUc6SrT5BTZYyK4e+lQc9twUGqeO61OlORbohaYk1SSiVtKBj9tHqhilhEGiRcc
ArosHsypqA

u/I3GZzfEKO+TkKPt6F6sX2i/q6ryMhwwkrCqejoU/wXERHB/Y/3sb9twzkU6a7140gX6R9
Oz8pk17hR

FbTUV3x11WE1jYn8SBpNQnukVxLGkaB5ukGAhD9UndEoEakW8vNp3VoHkFSDYx19U
7Gv9f1HbFkKpKhv

/wDpTwmUysaMHDllyZIIISO/nn3/O5KLHJ9RXXy1evLBPn15U27nzB6tXrwwI8Nvy8yZ2
Fy9e4OvrExFx

fO261SdjTqakJaO76zas2++xb9rMaQIJCfsO7MUQ4OUrI23cvAEZxpBk4mP1Mqus+gJzQBl
RX7aq104F

VUNIJREYGLhy5crFixeza4fCzsHf33/t2rWzZs1iQFgE0I2ePXvOnz+fzowePZpZ1uUuGYz6
Ogk93oZq

R7tFHDpREO41OPjoG2/8g9iI23727Nl4atLy1FUJtLOsSiy3K8yeM/Ott9+Uvq1cuWLgwP4x
MacGfz/w

m2++Rv8I06XPor76nKrB1gnLRlB9/A4vWDjPzW00OsfqZMCAfsOHD3V3XzJp0sRx48cy
PpyGTkhP1IVU

E9b4EAht3rJp6NDvfQ4fYgR2/bJz7tzZM2fOHDBgwIgRI0JCQlgtSWnrakphBbzR0YjKmjV
rkpKS1q1b

t3//fi7hpPp9323bfv62b++Tp6IYf4SW3Z49ewQFBXLtnEs6Mytjifti1Hfs+DHbd27z8fP59PN
P9+7f

s+an1d8N+G7z1k0Bgf6Lli5Kz0rPyM7IycvJzc/NV99JWTRZ1nxVDIPOlvFkMfGfqK/0QRJ
sGdVt27bh

lBBj6Vtp1HkanaXQWZVGn6bg8lkSLV++nHyaZrdfv359+vRhTL7+++mv6I0uESwejvk5Cj7f
BGeAjLDeB

/5UHYvv1+w53HBoa6u7uHhsbs2BU+p1nvAOeqALEgzimK4Hl8my6de9CRE58iSsePfoH
NMDb23PVqh/J

IE8cIY2cwmn6hKqB2rT0qqvU736j/cuWLyUhP167deuW115+gTJffvkF40M5cc2qhupFflgQ
7VRfy2wR

EBBAILts2TKkd8WKFcnJydZYlpIE2TK/EBQUtHr1anR6y5Ytnp6erLGYaGZ86dLFHh4H
OnXqyPUuXDR/

00ABn332KSPAsmzc+DH9+vWNiY2et8At6mTkR5903rJtCwL8yWcf79i1PSg06Pth3/+8/W
fPQ57uy90z

sjMzc7Jz8/PyCvLzrbcTKiW90kNBAIDnDGkJaF3aZRsWFkYC2SPwRfbIKRPps6CzFJxyX
ujTFJGRkYMG

DWIcfH19GYpevXrRk/79+zNrAwcOpD/osdXdSwajvk5Cj7fBiaAi3PDc23379iUdHBy8YM
GCiAjrmed9

+/Zp16KQ8nqnQqRkpbHKE1/i7nv1+h+6m5iYcCTQH6EdMmTwsbCQTZs3TJ4yCfUI9iUT
aZRTqhQqFOm1

TJqIT4gl7CZ9yMcr8GgA3Zg/363vd3041Pe7bxkxTsN1nv/1njficImHiMaysrKYFwJZV1dXn
PXgwYN3

7tx56NchtDTWK2d7ImlrMpQwyAKLiT5y5AiKy9bf359dMkNCgsOOH6Pwr192MMgnTkb
u2LmNi42KiiD2

TUIJ8vf3Xb7Cnbh25uwZxLjRcdEEwQe9DkaciCDT28crrzCPwHfXnl15rFpU6xeA9BO94W
JJ6FwnQgd0

Sv2Q38yZMxMTE5lfJ8eaLKoWL168fv36SZMmobXz589fuHAhYzJjxgw8JOtjhFkXvTQw
6usk9HgbnIg4

ZQYfn04whFtn0Y2zJuHh4aELOaD8+TnQRSuLlLcewJ49Z+as2TNcXCacPBW1cuUK1C46
+uSKH5f5HPZG

GFACnKESyPNt4pxQoZZeZRaengfpD8Izduzo7OxMd/clBOLkj58w1vr4kxPVV1i5cuXs2b
N37NgxceJE

IIfSmzdVn+1XUwfK1BfkD/x2hocHh7O6oopJic2NoaRZ/XDCKtlh7X4wFT0LwsR/svPLW
DkC9My0/L1

d1pZ1dtb9UBWLI FvvvXglRpN1S46ClZ8V4kPfaFz8ngRW/SGXX3AWYjqS5/VpZ1hVOm
M3RO5EBspKeis

X4c0QbuyBVp3dIyk7UOXCEZ9nYQeb4NTwGXg6XDBRLq4Y/HOCDALfxw06f379+MO
JBxhK5AuD46K+9D7

lfjEBYVxXPL5WpSVLYYMKJW1QOcwlaAvlJycSBoFpc8/F9Rf1IR+k1DqLAdLMtSoWG
1Jo5JDZ9AnKSSZ

DENychKV4xzZ6vbOhepO5fujobAIgKTBzrfTDDW3j70LUrm0AswpIS/SS9SL7jLdTDp4e
3sHHg2UD3ql

piVz4fK7vMyCDDhp62NOSacTk0+npqckpSRip0lnpGRkZ2TmZKamp6akpZDDNjM7Mys
n2zI1+5ZypGek

paXzMgOUrEzk3gfZlc/XAqdQQwVItVJYzk2tHPr8spBGGW2pkzVWBeXVVWokp3SXXKo
OcC9TDicTc0ihV

sWwiTSZbRoZ55CUkU3yJYNTXSejxNjgFZIA7H/+CAOOO5XsYiHeJfb28vA4dOrR7927c
kPV53FIusjzw

EeIsxA2dE1Wl9XlWJauWyMn3MyAu8tFeMkUP2MX5WO2n45i0X9a1lIOqvJhf5loqQTJN
cB6qk5R0mkZp

nd3UNOtzyXZt8g0SsiudAd1wOVxYf3C4qjnriyBIy66cizDKLmlad1RfR1BfTqcAW3YdW5
fjjLYYoltm

2tEKIRHsWk845+UUWMGuZdSSl5+Xp74JBcvNZa8AK8gvZOFhrT4qRJYmwMvScbcCK
AlqEXIeqxnQ55eF

NG2PJLPGrjRUGS6gM6BPVugsBZ3RqSKQZ7tv1whGfZ2EHm+DsxA3YbsMyZFDIGk8gu
MNb51QPrrQeVPk

2Yve85SEmJKHC675AqAtOsPVWD1x7IxzU6Fh8JkCGVs1UcXmQnatoS81cTbslvbjDsjlno
eJ+haZ9VYZ

9hsMTTVQYugMvzlGfZ2EHm/DbwF+B1fOSp807l4SpaFYBehCv5bS9WiFUBLoNFcvjVb
UVtVdckVUshU1

A2fRuedGLvM8zKivwWkY9XUSerwNvx2OoVWJ98EE8ezloQudH+f2/kWia+9akXrlm5PC
lcaqXxotHvue

bRd01QqdVT39YRZA0lbkWzRBdo6k5ZBkClKsEpx7/EuYvPlcZfZLYhcB5zNuBmdg1NdJ
6PE2/M44T29+

ASifrt/ddfDyRW/5lsi3Xb107JxI4cojQlugPrFa3M4qnC5aHDl0TnTpSmO3aycEe9dOCPq0yk
J5x7E9

XzvvzyEYKo9RXyehx9twyVHCoZ/TqtfjaxGrEF3UKegmK40+rbIY9TX8fjHq6yT0eBsuOU
o49HOaUd+K

0KdVFqO+ht8vRn2dhB5vwyXHeQlAtbt7LWIVoos6Bd1kpdGnnQfnNf6O5tRxMFyCGPV1
Enq8DZci+PFK

mqE6KDHIITSDoXox6usk9HgbDAaDwWDU12no8TYyDAaDwaiv09DjbTAYDAaDUV+
nocfbYDAYDAajvk5D

j7fBYDAYDEZ9nYYeb4PBYDAYjPo6DT3eBoPBYDAY9XUaerwNBoPBYDDq6zT0eBs
MBoPBYNTXaejxNhgm

BoPBqK/T0ONtMBgMBoNRX6ehx9tgMBgMBqO+TkOPt8FgMBgMRn2dhh5vg8FgMBiM
+joNPd4Gg8FgMBj1

dRp6vA0Gg8FgMOrrNPR4GwwGg8Fg1Ndp6PE2GAwGg8Gor9PQ420wGAwGg1Ffp6HH
22AwGAwGo75OQ4+3

wWAwGAxGfZ2GHm+DwWAwGlz6Og093gaDwWAwGPV1Gnq8DQaDwWAw6us09Hg
bDAaDwWDU12no8TYyDAaD

waiv09DjbTAYDAaDUV+nocfbYDAYDAajvk5Dj7fBYDAYDEZ9nYYeb4PBYDAYjPo6
DT3eBoPBYDAY9XUa

erwNBoPBYDDq6zT0eBsMBoPBYNTXaejxNhgMBoPBqK/T0ONtMBgMBoNRX6ehx9tg
MBgMBqO+TkOPt8Fg

MBgMRn2dhh5vg8FgMBiM+joNPd4Gg8FgMBj1dRp6vA0Gg8FgMOrrNPR4GwwGg8Fg1
Ndp6PE2GAwGg8Go

r9PQ420wGAwGg1Ffp6HH22AwGAwGo75OQ4+3wWAwGaxGfZ2GHm+DwWAwGlz6
Og093gaDwWAwGPV1Gnq8

DQaDwWAw6us09HgbDAaDwWDU12no8TYDAaDwaiv09DjbTAYDAaDUV+nocfbYD
AYDAajvk5Dj7fBYDAY

DEZ9nYYeb4PBYDAYjPo6DT3eBoPBYDAY9XUaerwNBoPBYDDq6zT0eBsMBoPBYN
TXaejxNhgMBoPBqK/T

0ONtMBgMBoNRX6ehx9tgMBgMBqO+TkOPt8FgMBgMRn2dhh5vg8FgMBiM+joNPd4G
g8FgMBj1dRp6vA0G

g8FgMOrrNPR4GwwGg8Fg1Ndp6PE2GAwGg8Gor9PQ420wGAwGg1Ffp6HH22AwGAw
Go75OQ4+3wWAwGaxG

fZ2GHm+DwWAwGlz6Og093gaDwWAwGPV1Gnq8DQaDwWAw6us09HgbDAaDwWD
U12no8TYDIbfMX15ecnJ

yYmJiaccOH78+LFKo89RUE9qampBQYGu/fdNTk4OV6p3qh+jvk5Cj7fBYDD8LkEjY2Nj
IyIi4uPjUc0s

Byovn5TU52RlpaenU4/USUKXKF5GsOvX+0WwFJD8SkJ5fWY5oK+6aClEevUKQpGZm
amPVQ9GfZ2EHm+D

wWD4XYJGpqSk6J0qRXQ9JiZGdtE5tE1HxyqwFp0T/dO5p05dgP6h8VyFPr8sOFqmANvS
q/fPnDl58qRR

34sEPd4Gg8Hw+wPdIuTVO9WDo54hdXZcS9NAIjk5GZGWTASbMpKuPHZV5VFXAaO
+Fyd6vA0Gg+H3B1Jn

v/1bTaSnP9vhLyqbmpoqafKRuhKZJOzCladicYWKcxj1vTjR420wGAy/MwhDo6Ki9E51Y
oezjuJqh7mO

bws7KnHIQVIRTZFY0LmqOcmRozq3FEZ9L070eBsMBsPvDGRGos/qBn2VCLvEG8vk0w
fHHNLn+8gVUImo

LJSQUuRc8ivQVKO+Fyd6vA0Gg+F3hsiS3jkXhYWFOnX+OEoa0badRhoJhe0VABEweoz
6kk8CUQTK00kR

b2JZ/RiVliEhQc7S+6dOUaCElCYnJ+tjRZBPbTQhu1RS4hS7aRLp6en6QFHrrA/kKB1zPFp
5jPo6CT3e

BoPB8DvjvNS3hBSdF47qS4v2c14pKSnImN0HdimJvJFjR8CIKwqN1CGZ5KOa8iEiKczpIJ
EcCXMIR6AA

OZwrBUCaA6pl17ESgbbst74pQGfkXXFEmpKSFjhKzXbhymPU10no8TYDYDIbfGaJDeudc
IFGITQkFqiTI

ra3cnE49kkaSqdYWZhHCMvVMIFWQHM4qIZxy1DGHAuSw1fvqKqSY7JaupMT73sgzT
VPM7rMj8ka6xOWV

x6ivk9DjbTAYDL8zEKHKh7MiUUgjeoOanpfkOAoeUIMtclQrVbElTYQt+ewie7YSsysxsV
1PNakviwOu

zu6etEtP7IGiS5xiF+BQcnKypCuJUV8nocfbYDAYfmfExMQgP3rnXNgShSAIJCSgScSFkn
NOHAUPHN/d

Rb0kgbzRhP2mNLuIHK3ILrALFQintOKYI+XZ6v1KqC/yT5hrFwCOAlctu5SngH3tXlh9C
ZXEqK+T0ONt

MBgMvzMQkgtQX4HgT3SoMjU4Ch44KpYdU6K7NGHvyimOjUoBu55qUl/Skqn3VSWOi
4Aya5B0JTHq6yT0

eBsMBsPvDGSjMtopOEoUcCLSSw1ZWVvk6q3wc5QpQbk9Ewg6ybT7I6c4Niq7bGVXZK
9EASmj96tOfR13

KRCr/ggtuyUKVwajvk5Cj7fBYDD8zriwd57RTk5ELO13j8+Jo+AJKLfj01vUSYXImN0fOc
VRF2WXreyW

Fk51RrWrr3xOyb72EoUrg1FfJ6HH22AwGH5nIEKV1BkqUB95ocIla39d9DKQHn7LWU
hISHBlkCQR4uR

sd+ / +pbAqO/vFz3eBoPB8DvDUYfOCbKE7qKR9uO+IQd9smVVKCFa8hyWYzHpm6Muyi
5b2ZUaShSQMnr/

QtXX8TNRQAHHGjjKWTbyiWF9rHIY9XUSerwNhouFQoXeMfyRcdShc4LGVOZPvGX
CuQiV3lGw6yhaUsCx

mPStHliKdsqulC9RQMro/QtV39J9c9ylANrMVmBFil+5VXmM+joJPd4Gwx8f0V2QtGQa/r
g46lC1gnrZ

siqUVjjJsYtJ35A32QXZZSu7Ur5EASmj96tOfe1PIYMUQHTID8AcsmurJEZ9nYQeb4PhYk
TE2Ano9hQF

BQU5OTm5FSII9U4lkMJyVgnIP6+/cf6BKCEz1QdapVNFyGNWeqdlfWOL/9Qgu466yK6j
1FFeYIDZBaWq

51BfTomJiSFTdmUEHE8hLR811vsqx7EANXA6nZFdq0mjvr9P9HgbDBcFWgwVOsu50C5
yyJ2F36wAKax3

KoEUlrNKQH55wvxHBwl0/ANnNcHapbT6gqOkifomJyfb37ZB3zgLpZRdQHodhRMoYJc
HEcIS1ZLjKKWU

55Lt5RSNIIBf2bWfEZNVAqfYT2iju7Rrf9sGFVb+yTXBqK+T0ONtMPzxQfnwQfidpKQkn
Jo8tuoInhq

kWrT0tLQv/z8fCX6hdnZ2XSABuCsy4MyUHEZR6SwLcDcuaTlFpajkn/xgZBcwFNU5wV
K5iiiNqXVVxS3

zHcaOIoEllmVPWvyaildbek67VNIIFDFeiV5VfA6l0Bccijg+PdvtnkW5kpi1NdJ6PE2GP7gI
Hts0aHI

yMjw8PCwsDC2uJ6QkBDSoaGhwchBJCS/qsDhskX/5O1f+sCW2wpXiE/UEu0A+dHR0fh
TErjpMss4IsWk

PI6bREBAgL+/P9dImtZJUOwivpG5ai5Q71QPDKOtdo6UlkkSTBnlS0gg8oaCSgESjmonT0
pLWolvYulq

mVzHa5TapIkS6sshCtuts8vLjy059AollnwbCtNhvVNpjpO6CT3eBsNFAZ4IV4XiIrcKUFzZ
BdKIMbtV

KMCIHw2hvrZDJIHDxRWKApbgxIkTIpmcSzdEic8J5SkeFxfHWUFBQYGBgXIVsqSgwP
m+tfjHglIDtPRO

lcJkMU1IIN4vDoMs4w+2uAJ6xiFOFDWVo/YUiHbKUeYa7HrkQkpXK9oJaDAnkgnsUkC
29insylHZJWG/

BU015NAEI0MrUg9dtV+Zlceor5PQ420wXBQQg+KqRG4RJ5FbiRSR0nQLEDBAt349NH
HkyBF8XH5+vnSA

BGKMNOIoab0ESrIt3QXpmd5QIaLWQHOCyJY0p9MBLpMCtCitX5SgH4wwlym6gs6hW
Da6UIVQgy6tIEYU

fWIYSehCpdCli3CUMamQc8HWPxs5Sj/IPXN1toYcnSrCrpY084h4k2YrR8ExDdRgVyInOi
JNUw9biunc

88Sor5PQ420wXBQgfhIRskVx0dpBgwb16NFj8ODBQ4YMQSnRXRSLfAoI5OhU+YiWO
yI5IoG4YNUH0gF2

yRSZLH1iaahEpxxwPJECdBip2Llz59ixYydNmrRr1y6uTtQXkpKSpPWLGLQEPWNsUU2
uXUCSGZzKoE9Q

iiqjmvASGRwx6usk9HgbDBcFhYWFqBGCisNFrrp06YJ64aMJfP38/L788ktbzMhcuHAhaSS
ZXbacyC5b

VA3IYRcRXbNmjSVxSg7J50S2dg5IED/H2BfPToXkSzGa5hTqsQNuOUvkma1URQLI0w
K+/v7Sw7buLi4

zp07v/vuuytXrly6dOn777//8ccfk0lh2qID8mdvg+HXY9TXSejxNhguFpA3ET+CXQQM0LZ
+/fpFRUUh

qAMGD6Qc98fX2/+ea/X15eQUFHw8OPnzhxAjFzdXX18vKUN4SIGGIORMhFCAUO
UafoKPml1RftJJ+j

SG+LFi0Qy1WrVIEURf5KCiKTtoJqqYfCc+fOpTAFyAGKUTmH6JUsFMaMGcPI0NCSJ
UtoPT4+fuTIkRMn

TqQzXFRycrJRX0NVYdTXSejxNhguFkR90bZvv/1W/tbbrVs3tE2Ec+DAgaKmm2fORJinT
J0cE3PqjTf+

Pnr0Dyt+XN616zeTJ7tu27Z1+PDhU6ZMmT17Ngmgzl69eo0YMQLxmz59OglqoDaqJYH4I
VBfyov6cvSL

L77IyspCgHft2oVkorV9+vSh5o0bN3p6eqK7iOuyZcs2bdo0f/584toFCxbQNLq7ePFiFg10Er
m97rrr

uBZq661AnqFVq1bkCIFGfQ1ViFFfJ6HH22C4WLDVt2/fvihWQEDAV199xRahIkzs37+/kr
Hjb//f2wsX

LXjnnbejoiKGDh0cFx8zdOj3CxfNDw4+OmrUSDQYRUQFORGI9PHxIU1wPG7cONQXR
aT+CtSXsFWOsr3n

nnuQ3m3btqHie/fu7dKli5+f37Bhw0jTMXd3d6oigTCvW7eue/fu+/fvp8yBAwc6d+5M5e+99
x4dfvDB

B6mTQJw+tGnTZv369TT66KOPcq5RX0PVYtTXSejxNhguFmz1Rc8kgXYiuoS87CJ7hJXr
flq37qe1J0+d

WL165Zq1qwYN7h8RETZo8AB39yUuLhM2b9nY43/dFy9eOHPmDG9vL3QXuSVoRno3b
Ngwa9YsKhT1BQRS

3nlg/yA3N5dd0V3R5nfeeSc1NRUFpQZOJIYm5B01ahQajL5SGyWJjwcNGkQOUbWvry8
hL9r87rvvEvX2

6NGDbYsWLeLj45Fe5JlroVpi37Zt29INgmnU1zxAZKggjPo6CT3eBsPFgq2+JHr27InoEjlif
gS+Xbt2

JZaVv6eeOB11LCwk9FhwePixEyfDw44HHwsLDgo6QiYVhIYGh4QERUaGqz/LHkVHqY
Q6IUI9Z0sOIKe+

dgGOLl26IHnhzZo11EWvvv3229WrV2/dutXDw4NImv4QZ2/evJm+EdRShiCYS2CL0Hlin
R87duyYMWPo

tZubG1uUGJ2eNm0aDRn1NVQtRn2dhB5vg+FiAXIDkJBjtsSaqBRqh4wRXBLCII3IsFJGd
PVI6LEgdDf0

WGDAEZ8jgb5hx5HeI2yR5KioCDQYAUa/pTYUkXMRVHW6BjGWd55FfUmwS3kOiQa
TLmrR+nQTWksNIGXo

LUgmR4FdTiSHowS7pOVprM8//yjjz5Cp5csWfKPF/zj008/5Vo4SnmjvoYqxKivk9DjbTBc
LKCRCBLS

JQImwkYEiQSKqkk+Rwh8Q0KPIqbBIQHKjoSEci7bo4S8KGPY8dBjlkRaga+cJTWI5gFpp
DEIJaWE+tIi

ZSgvZegSW5CqJHomE60lh3yU2C5APylg73JIdBqtJVyeNGnS5MmTCZFjY2MpID0h2jbqa
6gqjPo6CT3e

BsPFAqqGVkkQKdrGFhAqEDELdz9O8HksLAR9jToRfiwsSJnlbYjljnRosCRQX1X+7Bd
UCVRL/bTCIcfQ

U9QX+aQwjUofbJUV6KEgmVJPQEAAu0BaMmUrzUnrtGX3BKQb1GPU11CFGPV1Enq
8DYaLgqysLAQJZWKL

kolElcbSruP8Z+lrZCSSG6RMKy4ajMZhERGWSEsl1n9K/1BWiV9FKcmUv/tKB+TvvhTjF
PoAJEREKS+9

4iyRWDuTAvLutJxl5QyIK0LksNROQvZLv31+gbDBWPUt9qxvic0M1PeLtOjbjD8wRHxS
1Q/jJOQkECa

rQ2ZgrVzmv9iseTk03Hx0cpi2I2Li0k4HYclJiaIOULNgtQco76jn7vJMfblKOqIcIqCosSyFeF
kK4dI

25kUECEX2bZLytZGFJeSIrpslY4Hob7mLjZUFUZ9qx2jvoaLFV7SyKG8th0hUyCp9vIwih
cU5BZZXn5+

Tn6+JHLFCgr033Q5AXEVVCXWzwlnZ2eT1g0XQabcXyQoLFtWBpRnyy6QtjPZpTBpcq
SM49YRGtVtFOH4

Y64Gw6/HqG+1w93OfStORI+6wWA4Y24HwyWNUd9qx6iv4ZKHVz5ha5lmbgrDJYpR32
oH6c3+/3bs2AQA

AARg2P9XK7ips5PJE4XWENvfdH5QX5jU95z68p76QheRLWdVR2P5+kMAAAAASUV
ORK5CYII=

</value>

</data>

<data name="\$this.Icon" type="System.Drawing.Icon, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">

<value>

AAABAAYAEBAAAAEAIABoBAAAZgAAAAAAAAAABACAAHyOAAAM4EAAAwMAA
AAQAgaKglAADtLgAAICAAAEEA

IACoEAAAIVQAAICAAAABACAAGAgBAD11AABAQAAAAQAChCAABlbQEAKA
AAABAAAAAgAAAAQAAGAAAA

AAAABAAAEwsAABMLAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADu7
vQv7/H0le/v9Nfu7/L47u/y+O/v

9Nbv8fST7fPzLQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAP///wju8fOY7u/y/v
Dx9P/w8fT/8PH0//Dx

9P/w8fT/8PH0/+7v8v7v8fSU/v7+BwAAAAAAAAAAAAAAAAAAAAAP///wjw8PTA8PH0//Dx9P/w
8fT/8PH0//Dx

9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9L7+/v4HAAAAAAAAAAAAADu8fOY8PH0//Dx9P/w8f
T/8PH0//Dx

9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7/H0IAAAAADu7vQv7u/y/vDx9P/h4ub/qa2
y/6it

sf+orbH/qK2x/6itsf+orbH/qK2x/6mtsv/i5Of/8PH0/+7v8v7t8/Mt7/Hzl/Dx9P/w8fT/rK+1/+3u

8v/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/r6+//r7O4//Dx9P/w8fT/7/H0k+7v9Nnw8fT/8PH0/6
uu

tP/w8fT/8PH0/9vn9f+r0vj/uNj3/7vZ9//w8fT/7e7y/66xtv/w8fT/8PH0/+v9Nbu7/L58PH0//Dx

9P+rrrT/7O/0/+ft9P+v1Pf/jcX5/63T+P9/v/r/stX3/+bq8v+usbb/8PH0//Dx9P/u7/L47u/y+fDx

9P/w8fT/q660//Dx9P/O4vb/jMX5/zad/v9Pqfz/18n5/+3w9P/t7vL/rrG2//Dx9P/w8fT/7u/y+O7v

9Nrw8fT/8PH0/6uutP/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7e7y/66xtv/w8fT/8PH0/+
/v

9Nfu8fOY8PH0//Dx9P+sr7X/cHV8/3B1fP9wdXz/cHV8/3B1fP92fIP/pqqv/4WKkf+usrf/8PH
0//Dx

9P/v8fSV7+/0MfDx9P/w8fT/4OHI/6Klq/+fo6n/n6Op/5+jqf+fo6n/n6Op/5+jqf+ipav/4eLm//D
x

9P/u7/L+7u70LwAAAADw8POc8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0//Dx

9P/w8fT/7/HzlwAAAAAAAAAAAA4v7+CfDw9MLw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8f
T/8PH0//Dx

9P/w8fT/8PD0wP///wgAAAAAAAAAAAAAAAAAADI/v4J8PDznPDx9P/w8fT/8PH0//Dx9P/w8
fT/8PH0//Dx

9P/u7/L+7vHzmP///wgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAdv7/Qx7vHz
mO7v9Nru7/L57u/y+e7v

9Nnv8fOX7u70LwAAAAAAAAAAAAAAAAAAAAAAAAADwDwAAwAMAAIABAACAAQA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAACAAQAAgAEAAMADAADwDwAAiVBORw0KGgoAAA
ANSUHEUgAAAQAAAAEACAYAAABc

cqhmAAAABmJLR0QA/wD/AP+gvaeTAAAACXBIWXMAAAsTAAALEwEAmpwYAA
ApXUIEQVR42u3dd5xcZb0/

8M/3OWdmN9mpu2mGkoRUCAkEEuAaJAKKgoI/bCCCigrovfwu1wZeK/izi90X1wsoggg
TaUKSIdQQiAJ

JJCENAhJyJapW2bmfL+/PzZts21mpzws9/3PzDtOZ9z4Pnuqc9DUDVNm4dmwo1TiN2p4
B4Mjy0CFEL

gHEAWiAyjoCIGGkUMQ1G2A9j/MzcAMC/q5mMMaYHzBkmkyHiHmLqBtAhRG0A2gC
0kkgbHLRCzDYm2hRM

JjfRAQd02t4GauTIIdgA1PBHxpVLtc+A580EylyDTBZhKjGkwZrzVcMxvicEmAjYKaAOE
XoLjrQwEml8h

opztbaeGpgWgyjBvGZNBRYZoWPAOAKG5jPzHGOMv/jWK7kenIExawzLKhisYJJngs
Hoc0TUbTub2ksL

gGXJZHKi5HKLDdFiEV4swFHGGJ/tXOXAzBkCLSeiJ8nIk0zuk8Fg8C3buUYzLQAVxsw
N6UTiHQScCpFT

YehQ25nsbhBZDaJ7ReTeQCTyOBFIbEcaTbQAVEC6tfVAz3XPoN4OfyIBTbYzVamUCB
4S4D7HI/tHU9O4

rbYD1TstAGWSSqXeJp73ETCfJZB/M8boti4AM4sh86QAfyXXvSUQCGy3nake6f+UJRSP
x1uMyNKEfJJSF

jzfGGNuZ6gEzszHmMRb8lYGbI5FIu+1M9UILQJFEhFKx2Elk5AJmnFlrZ+trDTP3GEO3C
+OaQCTyCBGJ

7Uy1TAvACKVSqUnI5c4H8WcAM912ntGJ14PpWnHdPwSDwR2209QiLQAFSnV0HAW
DLzHLR40xru08CmDm

rDH0VzCuDESjL9rOU0u0AORBRCidSJwGyJcAnGg7jxrSQwBdGQiH77UdpBZoARiCiLj
pROw8ML4y6q/X

1xrByyD+SVMoeiMRbbjVCstAAMQEZNODJwL4Ft6fF/jmNcJmSsC4fBfiIhtx6k2WgD2IS
KUisc/SiKX

w9Ac23IUCbGsZuDbwUjkNr1ysJcWgF1S8fipEPkRCPNsZ1Hlw8CLJLg0GIk8YDtLNRj1B
SCRSMYBeD83

oPfazqIqR1juYmO+GA6H19nOYtOoLQCxWCzqilzOkH/Xy3mjEzNnDZlFz5i/09zcHLedx4
ZRVwBEhDrj

8QtF+HswpsV2HIUFmHfCof9uCkZ+P9rOD4yqAhCPx2c7IteCcLztLKOqPZwTXBiJRNbbD
lipo+JhFRHx

pWKxbxB7K7TzqyGcaIRXpWKxr4rIqDgsrPs9gHQstkgI1wKYbzuLqh0MvGhYPhOIRpfbz
lJOdbshICJO

Khb7lif8FLTzqwIZ4EiGPJOKxb4uInXbT+pyD6Cro2NaDriRDL3ddhZVBwSPGebzXjY3b7
EdpdTqrrKl

4vHzPEMvaudXJUM4gR2zMhmLnW07SulXrU7wtm1NnWMbrwboHNtZVP1iwQ3BdPrz9
TlhSl0UgHg8PtMR

uQOEubazqFFhZU7wwUgk8prtIMWq+UOAdLz9DAfynHZ+VUHzXcKyVDz+PttBilWzB
UBETDLe8V2P8TcA

Ydt51KgTYfbuTMY7LheRmt2TrsngO3bsClz1+28iQzVfgVXtE8E/Aun0x2rxvEDNFYB0eu
dkL+e7ywAL

bGdRajcGLzOO//Ram7+gpgpAMtk2T9i5xwAH2s6i1P4Y2Awyp4VCodW2s+SrZs4BpGKx
U4idJ7Tzq2pl

gClG+Klkr8fJtrMUKLn6JePxc1n4bgAh21mUGkZYIPfWyk1DVV8A0rHYRcLeDTpoh6oV
xhifCP8pFe/4

tO0sw2a1HWAoyXjHF4TwW51YU9UaY4xhlmuT8fh/2s4yZE7bAQaTisW+SaCf2c6h1EgZ
Y4ggv0zFYv9t

O8ugGW0HGEgy3vE9EL5jO4dSJUH4fjLeUZX/P1fdrnUqFvumdn5VlwRfC0QiP7AdY19V
VQCS8Y4v6G6/

qmci+M9gJPJr2zl2q5oCkI7FLhLCb23nUKqcmFmMoc8GwtHf284CVEkBSMbj5wp71xtjqv
KchFKlxMxM

ZD4ejERusp3FegFIxWKnsPDdep1fjSbMnCXQe4PR6EM2c1gtAMlk2zxi5wnoHX5qdIoxmc
U2nx2wtsud

Tu+cLOzo7b1qNItA+J5kMjnRVgArBWDHjh2BXY/0HmRrxZWqBgaYIpy9S7ZuHWtp+Z
Ullmas33+TPs+v

VC8DszDV1PQnGyMLVbwApBKxK3QkH6X6IsL/Scfj36j4ciu5sHS8/QyP8Td9uEep/piZjX
HeHwiH763U

MivWEePx+EwH8hx0AE+lhsAdjpiFYyKRDZVYWkUOAXjbtizH5A5o51dqGCaaJdzGvG
VMRZZWiYV0jm28

WsftVyo/BjgynQhdVYllf0QIBWPnwfIDZVYGaXqiQjODkYiN5dzGWUtAF0dHdM8Qy9
Cb/ZRaiRiJufN

H9vS8nq5FlC2QwARcXLAjdDOr9RIRdhx/igiZeunZWs4HY9/XafoVqpIhCXpePyy8jVfBul
YbJEn/JQ+

4adU8Zg5a0DHBqLRF0rddsn3AETEJ4RrtfMrVRrGGB8b+p2ILxPlbwApOPxSwHMr8iW
UWqUMMCCdDL2

xVK3W9JDgHg8PpvYW2GMaajcplFqdGBBFwPzI5HI+IK1WbI9ABEhR+Qa7fxKIYchjHE
JV5e0zVII1BmP

XwjCOyq/WZQaVU5MxTvOL1VjJTkEiMViUVd4HYxpsbddlBolmN/q9njmuHHjEsU2VZI
9ABf4tnZ+pSrE

mAl+1/1mKZoqeg8gkUjMgZdbpZf9lKocZs4wmbnFnhAsfg/A836mnV+pyjLG+B3gp8W2U
9QeQCoePxWQ

e2xvDKVgKxG8OxiJPDjS3494D0BECCI/sr0BIBrNhPDjYgYTHXEBSMXjHwVhnu0NoN
RoZoAFyVjszCJ+

XzgRcUjKtsrr5QCDNEVI31keEQ/SicS58DQHNSrrpQCQDg8FY9/dGQ/LZCIuOIExyuAm
W57vUvN8zy0

te5EMpFELpeFQGxHUiVAiLiuD6FwEM0t4+E4ju1Ipcf8alMkOpeIvEJ+VvDlu3Qidl49df5
MTw+eX/Ys

Xly+DJs2bkAul7UdSZWRz+fDtENm4IgFC3HUokXw+/y2I5WGMbPTicQ5AP5YyM8K2g
MQEUrHYi/D0KG2

17dY7HI4/NGH8eD996KzM207jrKgKRDAKe99H95+/BIYY22e3NIRrApEIgU9il9QAUjF4
+8D5C7b61ms

WKwDN/z+GmzeVJG5F1SVmzZ9Oj7xqQsRCtfBtBWC9wYikX/m+/UCC0DsIQAn2l7HYu
x8azt++5tfIBaL

2Y6iqkg02ozPX/xfabk/wXaUIvEDgXDzKfl+O+/9nlRHx1Go8c6fTMTx29/8Uju/6qejox3/e9
WvkEol

bUcPKnl3sr0978OA/A98DL5ke9WKISK48frfIxbrsB1FVam2tlb8+Y/X2Y5RNCH6cr7fzasA
pFKpScwy

ouuM1WLZs09j/bpXbcdQVe7VNavxwvPP2Y5RJdk7mUzmdSyT3x5ALnd+LT/xx56H+++t
+XOXqkLuu+dO

iNTuPSDGGB9x7pP5fHfYTi0ilEp0fJYqM49oWaxZsxt7W22Y6h9GMfBgqMWYuKkt434
mVRhxobX1uPV

Nav7fTZ+wiQsOfFkhMnhbHhtPR5/9OG87/Fo3fkW1r66BrPnHGZ7M40cy2cB/GS4rw1bA
FKx2ElkzCG2

16cYq1Y8bzuc2gcR4YLPXYxZs0tzO8nDD96Pu/5x+57X4ydMxBe+/FU0NDYCAA47fD6
mz5yFa3/7m7zb

XLXihdouAMbMSiTal4RCzY8O+bXh2iEjF9hel2JteO012xHUPg6dO69knR8A3nnyu/tcwz/
hxJP3dP49

yzzscEydlv/fdsdeK9nI29YYoWH77pAFIJFIjGPGiB81rAa5XBbtba22Y6h9RKLkrZHRIiE
97YZCUcG

/F44nP9y23a+BWa2sn1KhVk+FivFmof6zpaFgJjPMsbU9M3SnenOmj6hU482rF9b0v8mqV
QS27Zt3fN6

44b+e3ye5xV056fneeju6rK7oYpkjGn0AR8Z8jtDfUhATV/6A4CcI7MdQe1n+7ZtuPWmP6E
nkym6rUQ8

jhuuuxrZ7N4TfI8/+hBefWXviUHP83DHrTcVfA/Ivm3WKqGh+/CgJwHT6Z2TvQwfb6h2z/
6r6vX00iew

fNmziLaMA9HI9gY8T9DWthPs9X0CNpvN4uqrfoWp06YjHI5g86YNI7oBrB72G5I5STKZ
nBgMBncM9Pmg

BYBz/g8bM7JRRpTKRyabwY7tb5at/U0b9eSvMcZhz/sQgKsG/HzQXzKfZTu8Uqp4BjhriM
/6S7e2HiiQ

f7MdXCIVPBY+Pp3eOXmgzwYsAJ7rnmGMKenU4UopO4wxBlf+wf8bKA3SeRU26GV
UqXDwIB9ul8BYOYG

GKRp5/6VUn0R4WQR8e3/fr+rAOIE4h1EaLIduFLmHDoXjWPG2I6hLOjq6hzwQaI6FUzF
YosBPLLvm/0K

AA2yq1CvTv/AhzBp8uTiG1I1Z+vWN0ZTAQCITsV+BaD/OQA9/leqLg30x71PAUgmkxPr
YchvpdQACPMS

icT4fd/qUwAk1tsO6NSqnwc5N6+7+s+BcAQaQFQqo4xmz59vO8egLAWAKXqmcjABYB
5yxgBjrKdTylV

PgI5mpkhdr/eUwBSqcAiY4xvZM0qpWqBMaYhlUot3PN6z78IHWm7nFKq/Ei8Y3f/+95zA
IwjbAdTSpWf

CO2ZOmXvATBU0LTCSqkaRehbAETEx8xzbOdSSIUA86Ei4gC7CkAq1T6n1kf/VUrlxxjT
mEwmZwG7DwE8

R3f/IRpFiHk+sLsAkMy1HUgpVUEkhW07CgBBptvOo5SqHAI0AXafBASm2g6klKocYU
wDdu8B7HqhlBod

dv/Rd3nbtqZOY8YX2Z6qRuKBMkkgt2uOOyIABDiNkIZQ77+rUUomiUijmw74p1JtT4I6e
nEOJrEBpv1V

mOQboNQboNRWmPR2oCcGynZh0AmuyIH4Q5DGKCR4EDg8FRyaDm6eAY7MBhx9L
KSeGWMokUhMcYndKfUx

C1r9o544nO3Pwmx/Dk7byzAdawGvZ2SNiQfq6QD1dADxDXDeeHTvZ44PHD0U3oQj4U
08GjxxIcQftL36

qsQckWkuiCdDdFewWpm21+FufhDOm0/BdLwKSAV217wsTOtKmNaV8K2+ASAH3HIY
vAOR+6gd4KbddS4

ukDyNhceWqBTgFYV07EW7oZ74G6+H5R6w3YcQDyY11UwragW/E/kMBk5Ka+B7lpp4
OjM2ynUyPX4gpR

i/79rwK5Hrib7oNv7V9hWlfZTjMkSr0J30vXwffSdeDx85GdfTZyU98LGLf4x1XFCGicC2C
c7SCjGXW1

wbmerhrbwdlErbjFMzsXImGnSvhX/5LZA8/H9mZHwYcfaykJghaXAAttnOMRtS5E76Xr4
Nv3S1AboQn

8qoIde6A/9kfwvfy9cgcdQly006zHUkNb5wLkXG914dVJVC2E75Vv4VvzV9Gfga/ilF6Gxoe
/yrctbcg

c+w3wBG9y7xaETDOJSBiO8joIHA33AX/8l+AOnfm/avczA8jN+VdkDEtEF8AcBqxp2BzF
pRNg7rb4Vv1

ezhvPmF7JfdwdjyPMXd/FJl5FyB7+Gf1/EB1irpipJH0jrCyMvGN8C+9HM5bLwz4uTS9Dbk
DToBv/W0A

5/a+P3YCe065DHAaBm2bYmvhf+Z7MLHXbK9mf14W/hevgrv5QfQc/z1wdLbtRGpfxA2ui
GnQI4Dy8b3y

F/iX/6zfcB74mpA75HTkpp8OHjcPzvbn4Ft7c5/vZA89b8jO77z1Ahr+9XlQtjPvPBw8CNkjPg
/T9jKc

N5fCxDeUfRuYjrUYc/fHkJ17PjLzLxxynVTIMEyDa4T9IL0RoNSoqxUNT30Lzta+u+XSN
AnZuZ9CdsaZ

gLt3WvI+d+IBgHGRm3HG4O13voWGhy8pqPMDgEm+DnfjPeg++aredlJvwn3jUTivPwxn
x7I+eyAlxTn4

Vl0DZ9N9yCy6DN6BJ5RnOSpvhtnvQocCKznTugqNj/xXn2N9aYgie8Tndl0m63+fvbNjWZ
/X3oQFkIbo

oMvwP/cjUE9sRPmcrU/A3XgPctNOgwQmIzvnY8jO+Rgok4Dz+qNwN/8TzhuPoxy3iJvk62
h86GJ4BxyP

zMJLweGpJV+GypMxDWbfWUJU8dwNd2PMPz+9t/O7DcjOuwBdZ96F7JyPDFyQjXgwsf
V93vImLhxyOZm3

fwddp9+G7PyL+uxJ5Mv30u/6x/CHKJt+OrpP+k3Z7/Bztj6BMXd+EP7nf7H3aUVVUQz4DQ
DdAygR/wu/

RsMT/73n8l7u4Heh8wN3IrPg/w75MA11vgV4mT7vDXfCTHxN4OhMZI78D3S972bI2AkF
ZTUd62DiGwf9

nKOz+i8vXOJhIzgH38u/x9i/nQF3032lbXtX+9TTUfp26wVzgx78l4j/2R/Ct+oaAL1n77tP+j
V63vkz

SNOkYX9L3W393svnd7txeCp6TvhJwZlN60uDfiZNk/u+4Y5B1wf+jq7T/oTsrLN6L0mWC
HXuQMNjl6Lx

vk/A2bG8uMY4B2fbUjQsvRyNj34R4jSWLGc9cgFkAOhWGjGB/5kfwPfqTQCA3CHvR+b
YrxXUQSjb3f9N

X1NBKbwJC+BNXNjvXMKQy+1qHXyt/PstP9d7spHHzUNm3DxkFn0J7oZ74Xvlz71PKZa
A89aLcP75KXgT

j0JuzjnIHbgkrysGIHoTzvZn4by5FM6bT4AjM5Cd+yl4B70TOujJEIzpcY0xPdACMGL+p78
P39qbIb6x

yBz3rZHdAjvQVRgp/AQcF1gAYJzBP+v32PF+GZ1G5GaeidzMM+HsWAbfy38o2YIDZ8d
yODuWw++OBU88

Gtx8KDgwCXDH7hrHIAHq3AYT3wzTvhrUuRPiGwtv6qnoPuVafVw5TwbIuGDOwOiRw
Ej4XroWvrU3g6Mz

0bPkZ+DQlBG1Iw399xZGcoZfCjwZKIEDBv2MMsn9VnbwPRpv4kJ4ExfCdKyDb9U1cDff
X5JxCyjXCWfr

43C2Pj7IFwy8iQuRW/BfyE1514hOho5qzD0uk8lo9y+cu+Vf8C//NXJTT0XP4it6b9EdIRnT/
wQepd4A

JhxZUDsmuTH/L5OBN2HB4B8n+45DIE3Dn2Tk6Ez0nPBjZGMXwbei9w7A019KpN6Riqa
cgtyUUyBjdTjL

kWJjMi4R90BHBCmIaV8D/xNfR2bBxcjOu6Do9qQhAmmI9Pmrb9rXAIe8P+82KJuCs/lfe
X/fO+B4SGPz

EOv4Sp/XhVwB4Mh09Cz5KbJtq+F//udwtj9T3AYyLrxJx8A7+ETkDjxJO32JGHCPS0zd2v/
zR9IONDz+

VWQWX4HclPeUrF1umQvnzSf3vHa2PV3ArwX+pf+v/277oCthkDni84N/nHoTJvl6v3yFr9
Nh6D7lGrhb

/oWGR75Q0G9lTAu8yYvhHXgCvMmLIQWeFFV5ENPtAtALpQXwrbgKmeO+DW/iUSVt
15t0TJ8C0HudftOw

d8pRejsanv0BnNcfzntZ2XkXDNmh3U3/7J9v4qKRr9wAlzn7cXzwxh8Jb/K/wZu8GNw8B3
oGv+w6XCFq

082cH9O6ErkZZ4Ajs4pvbD/eQScCy3/e5z137V+RWXTpED/qRuPDI/QeLuQpO/tsZI7898G
/IAx33a19

3uLQIJE/1+91w7/y6v7vkwNuPhTe2xb17t5PPLqo8yiqcAK0uQDyKM8KACQ4BdIQLkvbH
J4KHj8fZufK

Pe/51t6K7OGfhowZZNQ2pxFd7/0DGpb9tLfTDnHmXRrCyBz9ZeRmfGDIHO7Gu/vt/uemn
z7i9fKvvKb3

Tsc+uf3oPPMuyNj8b3ZSZdHqAmgtuplRolydf7fsoR9Hwz4FAF43/Mt/jp7F3xv8R+4Y9Bz3
DWQPOw/u

+r/B2bYUINgM8nog/lDvX9mDliB3yBkQ39ghl0/ZNPzLf7lf+w3IzfwzwiNbHtK2G7+Xr+q/nr
I9o568G

hDaXRNp0SLDqkJvyHvjC/9vnGX33tTvhHXQScgefPORvOTQFmaMuAXDjiJfvf+b7/f5aZ
2efNeTVgsFQ

NoWGxy7t93ixNER6H2BS1hGk1cDRPYCqQQaZRV/p97b/yW/AxNaWddG+NTfC3XBnn/
eksRnZeSPorOKh

4bHLYJJb+n2UOfpLkIZIWddF5a3NQmw22ynUXt7kxchN7zsQCGXTaHzgc2Ub9stddyv8z
/V/mChz7NdG

MCWYoOHpKwa8e887aMmw5yBUBQltM0y0yXYO1Vfm2K+Bw4f0eY+6WtF43yfhvPIU
6RYkHvzLf4WGpd/B

/nfs5WZ+CLkppxTcXsPSy+Gu+1v/j8ZOQs/bv1OhLajy4RFtNMFkcpPtIKovccei56Rf9zv2p
kwCjQ9+

Hv7nflzwUGD7M/ENaPznp+F76dp+n3mTjkHPMV8rrMFCfxof+SLcdXf0/8xpRPeJvxhyhC
NVWcswWBw

k6EDDugE81vFN6lKiYMHofvdVw9wAk7gW3MjxvztfC9fD0omyqoXRPfiIalV2DMnR8ec
JRib8IC9Jz0

q4KmBzfxTRhz78cHvhmJHHSf8BNwy2G2N6nqg7YbY3pcABCDTQQUNqSMKjuOzkL3e
69Hw0MXwyQ29/mM

utrgf/6n8K/4DXKT3wHvgMXg8fPAgSmAu/cZeurgOI4DWbncrivPwbTunLQ5XkHn4Tu4
3/U5/dDE/he

vRn+538+8LBe5KBn8XfhHbTE9qZU+yFgI9A7IMjuF8fYDqX649AUdL/vL/A/dXnvY7b7y
/XA3fIg3C0P

7n3PHQMxLsjrBrzs8Asxbu+DTXPPR76335rYWvif/t6gex3AuOg5/vu9k4aqkNmnwIgoA1
6J0D1E18A

PUuuhLflQfif+wkoPcyFm1xX3nfRexOPRuaYr+c9CCil3oR/1W/hrr8TEG/wvO/8Gby3HWD
706lBCLAB

2FUAI PSSPndR/XIHv2vXDEK3w11944DX2fPITVyI7OGfhfA8Xl933SshW/19XA33jvk3
AE8/gj0LP7u

iAdHURUi9BKwuwA43kqwU1R7qkIcP7Kzz0Z29llwtj8PZ8uDcLYtHXKEXwCAccEtc5E7
4Hh4U98DDk0d

dlGUTcPZ9ADc1+4YfFd/f14GlnKaAGoamLMSmBXAQgEml9JxjoyRicJqSEeb9JCeJN6
5w+gbAoUew0m

vQ3IJEccgzgNQEMEHDwQHJoGOMP/56Xu9t6JQ15/CM7WJwuewdi0r0Hjg59DbsYHkFl
0WUIHD1alwczd

wUh4HbD7JCBRLhWPvQJgvu1wamTEF4CMPwI8/oiCfked2+HsXAXz1nI425+D6ViHUgz
j5a7/O5w3lyKz

8Mt6IrDaGLOaiDxg9yEAALCshCEtAHWIcp2g9A5QeitM4nVQYjNMbD1Mx7rhJ85wGiE
NYXBkBjhwIGTs

OMAf6t27gIC8DJBNgrraYdLbQMktvecmvGzv/IWPXQr3IT8ju+CS3mf+IXUk2HMteG8B
MFgB4Fzb4dTI

UHc7TGILKPk6THIzKPnGrtdbQJnEwD9yfJDGZkhDqPefYyZAmiZCggeCQ1N7zxP4mgrf
jeccTHwDTotL

cFpfnm+HI3/PB/epGORnX8hvElFjC6kikfSvwAwytNG9FJArZLGZniNzQOPJMw5gLog
XQOGCBnA+Adj

FraQfBkXHJ0Fjs5CbuYHAfROf+Zsfwbua3+H+++pN8A4+ufdZg3JIUINimD2jtO7Z+sFg9Dk
9EVinjAsY

twxz/eZPpk5A7pDTkTukd3QhSm+H07oKXsuhOhRYBTFzTzAceX736z3jARNRN4GKnJh
NqfxI06TeeQm0

81eUAS3bNRvYrtf7IKInC29SKVUrZL8+3qcAsEgJHzZXSUUbQzx4ASDX1T0ApeqYR26f
P/J9CkAwGNwB

ltW2QyqlymJIKBTqMwZo/0nBiO61nVIpVXoC9Ovb/QqAiGgBUKoOCfHwBSAQiTWOoL
BxppRS1S4RDEb7

nePrVwCIKCOC/GeaVEpVPRH8i4j6DeQw4MTgAx0rKKVqFw1yaD9gAXB8uX8ws807R
5VSJcLMDJ/vroE+

G7AANDWN22rI6D0BStUBY8xjgUBgWIEkzWA/EuCvtoMrpYrHMnhfHrQAkOveysyDTz
ivlKp6zOyR49w2

2OeDFoBAILDNGPOY7RUoFunU52qE6uH/HGPwcDAYHHTmLzPUj4fadagVjhl6tONM
NmM7orIk0z30gKc+

f/7To1UtpiH78NAFALiZmQsbFrbKjG0aO+RewNY3XrcdUVmydevg8yo4joPGxjG2IxaFB
V1Z4JahvjNk

AYhEIu3G0022V6QYrutD87jxg37+7NN6sWM0EhE8+/TgT7+PGz8RxpgCWqxCIrdGo9H
YUF8Zdg2FcY3t

9SjW9OmDT3u1ZfMmLH3qcDsRVYU9+fgjQ+79TZ8x03bE4jly7XBfGbYABCKRRwBeb3
tdijHviAVDfn7H

LTdh2bNP246pKuSZpU/i73fcOuR35h+5IM/WqhTzq6FQ87An8YcdkpWIJNXRcS0Mfmh7
nUZqzqFz0TJu

PNpadw74ued5+MuNf8Dy55/F2xcvwZRp02r++E/11d3ViU2bNuGpXX/B2lfXDPndCRMnY
cbM2bYjF8fQ

sH/9gTyvdCSTyYmSy75ujKnZ06LPP/cM/vzH62zHUDXgE5++AEccWbuTmDBzBo57YCg
U2jncd/M6yxEM

BncYQzV9SfDoRcdi9pzDbMdQVe6ww+fXdOcHAJC5KZ/OD+RZAAAjCtr1exzjnvfDQ
3t9iOoapUy7jx

+NjHP2k7RtHI8fLuq3kXgEA0+iKAh2yvXDECwSA+9x+XIBptth1FVZmWlnG46D8uwdi
mJttRisLg+4PB

lIX5fr+gux1T8fipgNxjeyWLIYjH8cfr8WG9etsR1FVYMbM2Tj3U59BMBiyHaVoIjglGIk8
kO/3C77d

ORWLvQTCXNsrWiwRwdInHsP9992NZDJRfIOq5oTCYbzntNNx7HGL6+KZEQZWWhMK
RIww5TeEFIN7+ScD8

wfbKlko2k8WLLyzDiheex4b1a9GT0WcD6pm/oQEzZszCkUcdjflHHg2fr2YvbPUjgnODkci
fCvINwQVA

RJx0rGMNjKmDW6X6rRva29uQTCSQzWYgooMi1QMigs/nRzAUQnNzS138te+HZU1TJ
HI4ERX0CP+ItkQy

Fvs4EW60vc5KqV4iOCsYiRR8qX5EBUBETDoWWwVDemFdKftWNoXCRxJRwbusI3rci
YhYiC63vdZKKYBI

vj2Szg8UMeiJiFAyEX/BAEfY3gBKjVYMXhYKNy8a6e9H/MAzEYkRXGp7Ayg1mhFTU
X2wqBEPApHI/cJy

VzFtKKVGRgR3BKPRombxKnrIEzbmS8yctb0xlBpNmLnHBb5cbDtFF4BwOLzWkPm17Q
2i1GhChn4+JhLZ

UHQ7pQjT3t4e9hPWwZjxxbemlBoKs2zvymRmTpw4sehZvEsy6mFzc3McDv237Q2j1Ghgj
FxWis4PIHDu

AxGhdCL+EIB32towStU/fiAQbj6lVK2VbNxjIhIPdCEzd9vZMERVNwY6HaaLStlmSQc+D
4fD6wyZKyq7

WZQaHQjyrTHR6MbStlliIuImE/HnDHBkxbaMUnWOwcuCoehxROSVst2ST31CRDnD8hl
mzlVu8yhVv5g5

Yzx8ttSdHyhDAQCAQDS63JC5vNwbRqnRwDj0zUBz84pytF22kRFExKTj8UdAeEf5No1S
de+hplD4XSN9

2m84ZZv9kIjYMJ8LIF62TaNUXeN2yuY+Wa7OD5SxAADA2ObmLSL4fDmXoVS9YqaL
msaNe6Ocyj7/MfB

SOQvLLih3MtRqp6I4HehaPTW4lsaWkVGR5StW8emA01LAcyvxPKUqmUMLA+GwouJq
Ow31ZV9DwAA6IAD

OnOCDwKIVWJ5StUs5jbX4w9WovMDFSoAABCJRF4D6Fxm1rG2lRoAM7OQ+djY5ubN
IVpmxQoAAATC4bvJ

0HcquUylaoUh881CpvUqhYrPkCAilIrH/0aEMyq9bKWq2G1NofBHynnJbyAV3QMAep8ab
OrqOoeB5yu9

bKWqkTA/MzaYOK/SnR+wsAewWyqVmsRe7hkDHGwrglK2CbARxjkuGAy+ZWP5Fd8
D2C0QCGwn450GvVNQ

jVrcIWROs9X5AYsFAACCwZaXheVDOqqwGm2YOcOEM0Oh0Cs2c1gtAAAQjEb/RWQ
+ycwFzWqqVK1iZo/I

nBsKNT9qO4v1AgD03i5sHLpQ7xFQ9Y6ZxRh8JhiJ3GI7C1AIBQAAAqHo78jgC7ZzKFV
WZC4OhJuvtx1j

t6opAAAQDDf/EoKv286hVFmQXBaKRK6yHWNfVVUAACAQiXxfIN+1nUOpUhlIFYF
Q9Me2c+zP2n0Aw0nF

Y18D4Xu2cyhVNJLLqrHzA1VcAAAAGY//p7D3C2NMVedUaiDMLCBzcbXt9u+r6jtWKtH
xGfbkamNM1R2u

KDUYZvaMwWeq6YTfQKq+AABAMhY7W4RvMMb4bGdRajjMnCEy51bLpb6h1EQBAI
BkR8dJZO2AGHbWZQa

HHf03uFn/yaffNRMAQCARCJxGITvMcAU21mU2p8AG4XMabZv7y1ETR1Xh0Kh1cZxj2
PwMttZINqXMD8D

4xxXS50fqLECAPQ+RRhMdS0Rwd9tZ1Fql9uawqkTbT7VN1I1dQiwLxGhdDz+DRa+XK8
QKBuYmQ2ZbzaF

wz+wMZhHKdRsAdgtFY+fCnh/AkzUdhY1ijC3gcw5gUjkfttRilHzBQAAumKxQ7KE2w1w
h00sqv4xsNz1

+IOVHL23XOqiAAAA85YxqUTofwzhk7azqPolgt8FwuGLKzVuf7nVTQHLYLRmLnU2E/w
EQsZ1F1RNUZ6YL

Q9HobbaTIFLdFQAA6GxvP5iN+SMIJ9jOourCQ+TmPtHUNG6r7SCIVpcFAABExKTj8ct
Y+Aq9hViNBDNn

jEPfbApGriSiuhyym4LwG6pjo4FbOh3BlhgO4uqHQxeZjx8NtDcvMJ2lnKq+wIAACLippO
xLzLT5YYw

xnYeVb0Y6CTItwKhyC+IyLOdp9xGRQHLYLRaLzXAJVwM40XYWVY34AYfpjHR6Ebb
SSplVBWA3VLxjvPB

8kMYM8F2FmUfs2w3Ri4LhJtvsJ210kblLbSBcPS6bo9nsuBKZs7YzqPsYOYegfywJ5ebNRo
7PzBK9wD2

FYvFZjjAT3W24IFGcLsDfGVMJLLBdhSbRn0B2C0Zi71LCD/WqwX1jcHLiOnSYDT6sO
0s1UALwD5EhJKx

2JmG6AoQDredR5XUSiL+VIOoWR8j34cWgAGliEnF42eR8OUwZpbtPKoILGuE6PJAoH
xLrT6yW05aAIYg

Ik46kTgHil8BYZ7tPCp/DKwGwU8C4fBf6vUuvlLQApCnVCz2HhB/GTDvsp1FDY7B95O
YK4ORyAO2s9QC

LQAFSrW3H8FEXwLkbH3GoDowcwZkbiLHuzIYbFllO08t0QIwQslkegJJ7IPw5LMwZqbt
PKMS86swdC2T

c30oFNppO04t0gJQAoIE+xIjdAGzfMgY02g7Tz1jQRdEboUj14RCzY/bzlPrtACUUCwWa/
YBHxHCR515

iTHGsZ2pHuyaZuthMP01C9wSjUZjtjPVCy0AZZJMJieK533IAGex8PE6cnFhmJmNMY+R
4GbPmNt0F788

tABUQDq9czKyvvczcCoRTgYQtJ2pSiUgeBAi98HnuysQCGyzHajeaQGoMBHxpWKxxSA
6lYBT9f4CrBTg

XiG+NxiMPkVEWduBRhMtAJYIEonxDnJvZzaLlbJYIEcbYxps5yoHZu4xoGVC9KQhftIj9
6lQKNRqO9do

pgWgyjBzQyqVWkjiHStC80GYD+bDaq0oMHM3jFlNgpUgWckwzSDweeNMT22s6m9t
ADUABFxsksnkLGKe

D5LDCThEGNMEArIJGOMlf+OzCwAbSdgIxlsFGADhF4SY1YGg8F1o2FIrVqnBaDGi
UhjIpGY4ohMA8nb

ALQIaBwELQDGETAOQBTEdQzTYJj9MKaBAT+Ye/cqjOkxQAbMPWxMxoB7IKYbQI
cAbQBaqWgjSCuANght

84g2hkKhzfUyQcZo9f8BOPkCRj8yCmoAAAAldEVYdGRhdGU6Y3JlYXRlADIwMTU0MTU0
DUtMDZUMTM6NDU6

NTctMDU6MDBGHlrqAAAAJXRFWHRkYXRlOm1vZGlmeQAyMDE1LTA1LTA2VDEz
OjQ1OjU3LTA1OjAwN0Pi

VgAAAABJRU5ErkKgAAAAwAAAAAYAAAAEAIAAAAAAAAACQAABMLAAATC
wAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

APPz8xbw8PNX8PD0j+/v8rbu8fTb7u/06+7v8vnu7/L57u/06+7x9Nvv7/K28PD0j/Dw81by8v
IVAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

APHx8RPw8PNs7vHzye7v8v7w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8f
T/8PH0

//Dx9P/u7/L+7u/zyPDw82vw8PASAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAD09PQZ8PD0jvDx8vXw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/
w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fP08PD0jPT09BgAAAAAAAAAAAAAAAA
AAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAA////Be/x9HXw8fP08PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/
w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9PPv8fNz//
//

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAADx8fEl8PHyz/Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//
Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PHyzfDw8CMAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAO/y9VDw8fTy8PH0//Dx9P/w8fT/
8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9PHu8fVOAAA
AAAAAAAAAAAAAAAAAAAA

AA8PD1Z+7v8v3
w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u7/L87/L0ZAAAAAAAAAAAAAAAA
AAAAAAA

AA
AADw8PVn8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/+/y9GQAA
AAAAAAAAAAAAAO/y9VHu7/L98PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+7v8vzu8fVOAA8fHx

JvDx9PPw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fTx8PDwIwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAD///8F8PHy0PDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PHyzf///wQAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAADv7/R38PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//
Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

/+/x83MAAAAAAAAAAAAAAAAAAAAAAAAAAAAOv19Rrw8fP08PH0//Dx9P/w8fT/8PH
0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9PP09PQYAAAAAAAAAAAAAAAAAAAAAAAAAAAPDw9JDw8fT/8PH0
//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+jq7f+usrf/naKn/5yhp/+coaf/nKGn/5yhp/+coaf/nKGn/5yhp/+coaf/nKGn

/5yhp/+coaf/nKGn/5yhp/+coaf/nKGn/5yhp/+coaf/nKGn/5yhp/+coaf/naKn/6+zuP/p6u7/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8PSMAAAAAAAAAAAAAAAAAA8fHxE/Dx
8vXw8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/7O3w/3R5gf9eZG3/cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh
/

/3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh/

/11kbP92fIP/7e7y//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fP08PDwEgAAAAAAA
AAA8fHz

bvDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/xcjM/1hfZ//U19r/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0/9DT1v9YX2f/yczQ//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/
8PDz

awAAAAAAAAAAA7vHyy/Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/u7/E/1hfZ//r7
O//8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+nq7v9YX2f/vsHG//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/7u/zyAAAAAD09PQY8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/1hfZ//r7O//8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+nq7v9YX2f/vsHG//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/7u/y/vLy8hXw8PNZ8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/u7/E/1hfZ//r7O//8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+ru9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+nq7v9YX2f/vsHG//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw81bv7/SR8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/u7/E/1hfZ//r7O//8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/P4vb/qdL4/63T+P+UyPn/cLj7/3a6+v9wuPv/mMr5/9rn9f/w8fT/8PH0//Dx9P/w8fT/8PH0

/+nq7v9YX2f/vsHG//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw9I/u8fK58PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/u7/E/1hfZ//r7O//8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/zeH2/06p/P9brvz/aLT7/0Gj/f+ay/j/7O/0/6PO+P82nf7/qdL4//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+nq7v9YX2f/vsHG//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/+/v8rbu8fTc8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/u7/E/1hfZ//r7O//8PH0

//Dx9P/w8fT/8PH0//Dx9P/f6vX/RKP9/8zg9v/v8fT/7O/0/+ju9P9fsPz/o874//Dx9P/Y5vX/O6D9

/8Pd9v/w8fT/8PH0//Dx9P/w8fT/8PH0/+nq7v9YX2f/vsHG//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+7x9Nvu8fTt8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/u7/E

//1hfZ//r7O//8PH0/9jn9f/w8fT/8PH0//Dx9P+KxPn/nsz4//Dx9P9fsPz/JJb//5/M+P/i6/X/RKP9

//Dx9P/w8fT/qtL4/zug/f+LxPn/xt72/+x9P/w8fT/8PH0/+nq7v9YX2f/vsHG//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+7v9Ovu7/L68PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/u7/E/1hfZ//r7O//8PH0/93o9f+by/j/6+70//Dx9P9lsvv/z+L2//Dx9P+ezPj/HpP/

/zug/f/w8fT/R6X9/9Pk9v+Cv/r/KZj+/x6T//8ol/7/brj7/5fJ+f+62ff/3un1/+nq7v9YX2f/vsHG

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+7v8vnu7/L68PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/u7/E/1hfZ//r7O//8PH0//Dx9P/b5/X/b7j7/6rS+P9xuPv/ttf3

/9jn9f8ynP7/HpP//1as/P+dzPj/LJn+/yCU//8ek//HpP//2q1+//b5/X/8PH0//Dx9P/w8fT/7/H0

/+nq7v9YX2f/vsHG//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+7v8vnu8fTt8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/u7/E/1hfZ//r7O//8PH0//Dx9P/w8fT/7fD
0

/5vL+P84n/7/OZ/+12w/P8xnP7/HpP//x6T//8ek//HpP//x6T//9Bo/3/ttf3//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+nq7v9YX2f/vsHG//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/+7v9Ovu8fTc8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/u7/E/1hfZ//r7O//8PH0

//Dx9P/w8fT/8PH0//Dx9P/t8PT/t9j3/3y++v9Vq/z/QaP9/0Gj/f9XrPz/g8D6/77a9//v8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+nq7v9YX2f/vsHG//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+7x9Nvu8fK58PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/u7/E

/1hfZ//r7O//8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+nq7v9YX2f/vsHG//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+v8rbv8fSS8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/u7/E/1hfZ//r7O//8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0
0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+nq7v9YX2f/v
sHG

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw9I/w8PNa8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/u7/E/1hfZ//r7O//8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0
0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//+nq7v9YX2f/vsHG//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw81f09PQ
Z8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/u7/E/1hfZ//+an6X/nKGn/5yhp//+coaf/nK
Gn

//5yhp//+coaf/nKGn/5yhp//+coaf/nKGn/5yhp//+coaf/nKGn/5yhp//+coaf/nKGn/5yhp//+coaf/nKG
n

//5yhp//+coaf/nKGn/5yhp//+coaf/nKGn/5yhp//+coaf/nKGn/5yhp//+coaf/nKGn/5yhp//+coaf/nKG
u/y

/vPz8xYAAAAA8PHyzPDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/u7/E/1hfZ/9YX
2f/WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/11ja/9wdXz/cHV8/3B1fP9bYmn/aG92/1hfZ/9YX2f/vsHG//Dx9P/w8fT/8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/7vHzyQAAAAAAAAAAAA7vHzb/Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/
w8fT/xsjN

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/42SmP/k5en/5OXp/+Tl6f+FipH/2tzg/2Bnbv9YX2f/x8rO//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PDzbAAAAAAAAAAAAA8vLyFPDw8vbw8fT/8PH0//Dx9P
/w8fT/8PH0

//Dx9P/w8fT/7O3w/3J5gP9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
/WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/90eoL/
7e7x

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fL18fHxEwAAAAAAAAAAAAAAAAAO/x9
JLw8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+fp7P+sr7X/mp+1/5qfpP+an6T/mp+k/5qfpP+an6T/m
p+k

/5qfpP+an6T/mp+k/5qfpP+an6T/mp+k/5qfpP+an6T/mp+k/5qfpP+an6T/mp+k/5qfpP+an6T/
mp+l

/6yvtf/o6e3/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8PSOAAAAAAAAAAAAA
AAAAAAAAA

AOzs9Rzw8fL18PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/
/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx8/T09PQZ
AAAA

AAAAAAAAAAAAAAAAAAAAAAAAADw8PR68PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/
/Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/+/x9HUAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD///8G8PH0fDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PHyz///wUAA
8fHxJ/Dx9PPw8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fTy8fHxJQAA
AAAAAAAAAAAAAAAAAAAAAA

AO/v8lPu7/L98PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+7v8v3v8vVQAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAdw8PJp8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//
Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw9WcAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA

AAA8PDyau7v8v3
w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u7/L98PD1ZwAAAAAAA
AAAAAAA

AA
AAAAAAAAAAAAA/v8IPw8fTz8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9PPv8vVR
AAAA

AA
AA

AAAAAADx8fEn8PH0fDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PHy

0PHx8SYAA
AA

AA////BvDw9Hrw8fL18PH0//Dx9P/w8fT/8PH
0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx8/Tv7/R3///BQAA
AA

AA
AADs7PUc7/H0kvDw8vbw8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fL18PD0kOv19RoAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAA

APLy8hTu8fNv8PHyzPDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8f
T/8PH0

//Dx9P/w8fT/7vHyy/Hx827x8fETAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAA

AAPT09Bnw8PNa7/H0ku7x8rnu8fT
c7vH07e7v8vru7/L67vH0

7e7x9Nzu8fK57+/0kfDw81n09PQYAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA

AAAD//4AB//8AAP/8AAA//
wAA//AAAA//AAD/wAAAA/8A

AP+AAAAB/wAA/wAAAAD/AAD+AAAAAH8AAPwAAAAAPwAA+AAAAAAfAADw
AAAAAA8AAOAAAAAABwAA4AAA

AAHAADAAAAAAAAAMAAMAAAAAAAAwAAgAAAAAABAACAAAAAAEAIAAA
AAAAQAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAIAA
AAAAQAAGAAAAAABAACAAAAAAEA

MAAAAAAAwAAwAAAAADAADgAAAAAAcAAOAAAAAABwAA8AAAAAPA
AD4AAAAAB8AAPwAAAAAPwAA/gAA

AAB/AAD/AAAAAP8AAP+AAAAB/wAA/8AAAAP/AAD/8AAAD/8AAP/8AAA//wAA//
+AAf//AAAoAAAAIAAA

AEAAAAABACAAAAAAAQAAATCwAAEwsAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAOnp6Qzw8PNW7vHzmO/v88fu7/Tp7u/y+e7v8vnu7/Tp7+/zx+
7x85jw8PNV5+f+

CwAAA
AA

AAAAAAAAAAAAAAAAAAPDw8CPw8POb8PHz9PDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/
/Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAP///wjw8POJ7u/y+vDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//
P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//
AAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAADw8PAj8PH00fDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//
//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//
AAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA7u70L+7v9Onw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//
H0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//
AAAA

AAAPDw8CPu7/Tp8PH0//Dx9P/w8fT/
8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

/+/v8ufv7/chAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADi/v4J8PD00vDx9P/w8fT/8PH0//
Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx8s/+/v4HAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAPDw84rw8fT/8PH0//
Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+/x84YAAAAAAAAAAAAAAAAAAAAAAAAADw8PAk7u/y+vDx
9P/w8fT/8PH0

//Dx9P/w8fT/7u/y/+vs7//r7O//6+zv/+vs7//r7O//6+zv/+vs7//r7O//6+zv/+vs7//r7O//6+zv

/+vs7//r7O//7u/y//Dx9P/w8fT/8PH0//Dx9P/w8fT/7u/y+fDw9yIAAAAAAAAAAAAAAAAAAPDw85zw8f
T/8PH0

//Dx9P/w8fT/8PH0/7W3vP9la3P/aG92/2hvdv9ob3b/aG92/2hvdv9ob3b/aG92/2hvdv9ob3b/aG92

/2hvdv9ob3b/aG92/2hvdv9la3P/uLq//Dx9P/w8fT/8PH0//Dx9P/w8fT/7vHzmQAAAADr6+sN8PHy

9fDx9P/w8fT/8PH0//Dx9P/w8fT/bHJ6/+Pl6P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+Dh5f9wdXz/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fP05+f+

C/Dw81jw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P9ob3b/7e7y//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/6+zv/2pweP/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8PNV8PDzm/Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/2hvdv/t7vL/8PH0//Dx9P/w8fT/8PH0

//Dx9P/n7fT/y+D2/8rg9v/i6/X/3ej1/9fm9f/w8fT/8PH0//Dx9P/r7O//anB4//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+7x85ju8fLK8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/aG92/+3u8v/w8fT/8PH0

//Dx9P/w8fT/mcv5/3y++v96vPr/rdP4/4TA+v9Hpf3/6+70//Dx9P/w8fT/8PH0/+vs7/9qcHj/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/7+/zx+7v9Ovw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P9ob3b/7
e7y

/+/x9P/w8fT/8PH0/5jK+f+32Pf/tNf3/9Hj9v9xuPv/4er1/7/b9/99vvr/7vD0//Dx9P/w8fT/6+zv

/2pweP/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u7/Tp7u/y+vDx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

/2hvdv/t7vL/3un1/8jf9v/w8fT/e736//Dx9P9otPv/PqL9/7/b9/+izvj/l8n5/ymY/v9Aov3/qtL4

/9Lk9v/r7O//anB4//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+7v8vnu7/L68PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/aG92/+3u8v/w8fT/zeH2/43F+f9Zrfz/x9/2/y+a/v82nv7/Xa/8/x+T//8ilf//gb/6

/+ft9P/w8fT/8PH0/+vs7/9qcHj/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7u/y+e7v9Ozw8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P9ob3b/7e7y//Dx9P/w8fT/7fD0/6vS+P9lsvv/PqL9/zOd/v9Jpv3/fL7
6

/8/i9v/w8fT/8PH0//Dx9P/w8fT/6+zv/2pweP/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u7/Tp7v
Hy

yyDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/2hvdv/t7vL/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/r7O//anB4//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/+/v88fw8POb8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/aG92/+3u8v/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//vs7/9qcHj/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/7vHzmPDw81nw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P9ob3b/hImQ/4WKkf+FipH/hYqR

/4WKkf+FipH/hYqR/4WKkf+FipH/hYqR/4WKkf+FipH/hYqR/4WKkf+FipH/hImQ/2pweP/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8PNW7OzsDvDw8vbw8fT/8PH0//Dx9P/w8fT/8PH0/2xyev9YX2f/WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/3F4f//FyMz/xcjM/4uPlv99g4n/b3V8

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PHz9Onp6QwAAAAA8PDznvDx9P/w8fT/8PH0//Dx9P/w8fT/sbW6

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+7v9Onw8PAj
AAAA

AAAO/v9DHu7/Tq8PH0//Dx9P/w8fT/
8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u7/Tp7
u70

LwAAAPhx8
SXw8PTU8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

0fDw8CMAA
AAAAAAAAAAAAAAAAAOL+/gnw8PSM7u/y

+/Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7
u/y

+vDw84r//8IAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAADx8fEm8PDznvDw8vbw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/
w8fT/8PHy

9fDw85zw8PAkAAA
AA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA6+vrDfDw81nw8POb7vHyyu7v9Ozu7/L6
7u/y+u7v9Ovu8fLK8PDz

m/Dw81jr6+sNAAA
AAAAAAAAAAP/AA///AAD//AAA

P/gAAB/wAAP4AAAB8AAAAPAAAADgAAAAYAAAEAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAACAAAABgAAAAcAAAAPAAAAD4AAAB/AAAA/
4AAAf/AAAP/8AAP//wAP/KAAA

AIAAAAAAAAQAAAQAgAAAAAAAAAAEAEwsAABMLAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAA7u72H/Dw9Ejw8PJp8PDziu7x9Knw8fS+8PD00vDw8uXw8fTv8PHy9
e7v8vzu7/L88PHy

9e7x9O7w8PLI8PD00vDx9L7u8fSp8PDzivDw8mnw8PRI7u72HgAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7vH07vDw857w8PR
I///

BAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA8PDwI+/x84fv7/Ln8PH0//Dx9P/w8fT/8PH
0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/v7/Lm7/HzhfDw9yIAAAAAAAAAAAAAA
AAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAA7+/zQe/v87Tu7/L+8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/7u/y/u/v87Pu8vI+AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAA7vH1TfDx8szw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P
/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u8fLK8
fH0

SwAA
AA

AA
AA

AA
AA

AA
AA

AA8PDzRe7x8sr
w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7+/zx+/v80IAAAAAAAAAAAAAAAAAAA
AAAAAAAAAA

AA
AA

AA
AA

AA
AA

AA
AA8fHx

Ju/v87Tw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dw87Dw8PAkAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AA

AA
AA

AA
AA

AA4v7+CfDx84ju7/L68PH0//Dx9P/w8fT/8PH
0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+7v8vvnv8fOE/
///

CAAA
AA

AA
AA

AA
AA

AA
AAAAAAAAAAAAAAAAAAO/v80Lu8fTc8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u7/TZ7vLyPwAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAA

AA
AA

AA
AA

AA
AA

AAAAAAAAAAAAAAAAAAAAAAAAAP///wjv7/SR7u/y/vDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P
/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/u7/L+8PD0jP///wYAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AA

AA
AA

AA
AADt8/Mr8PD00vDx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PHyz/Hx8ScAAAA
AAAAA

AA
AA

AA
AA

AA
AA

AAAAAAAAAAAAAAAA8PDzV/Dx8/Tw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P
/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH08u/v8IIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA

AA
AA

AA
AA

AAA////AvDw9Izw
8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx84j//8CAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AOfn/gvw8POx8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw9K3i/v4JAAAAAAAAAAAA
AAAAAAAAA

AA
AA

AA
AA

AAADz8/MW7vHyy/Dx9P/w
8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0/+7v88jy8vIUAA
AAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA7Oz1

HO7v9Njw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+v9Nbr9fUaA
AAA

AA
AA

AA
AA

AAO/v9yHw8fPe8PH0//Dx9P/w8fT/
8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+7x893u7vYfAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAADs9fUb8PHz

3vDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+/v9NXy8vIUAAAAAAAAAAAA
AAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAOfn/gvw8fLM8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+7v88ji/v4JAA

AA

AAAAAAAAAAAAAAAAAAAAAAAAAAD///8C8PDzsfDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw9K3///8CA
AAA

AA
AA

AA
AAAAAAAAAAAAAAAAAPDw9I3w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx84gAA

AA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADw8PNY8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH08vLy8igAAAAAAAAAAAAAAAA
AAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAP7+/gfw8PTS8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx
9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PHyz///wYAAA

AAA
AAAAAAAAA7+/0kfDx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0jAAAAAAAAA
AAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAO/v80Lu7/L+8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx
9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/u7/L+7vLyPwAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA

AA
AADi/v4J7vH03PDx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u7/TZ/v
7+

BwAA
AA

AAAAAAAAAAAAAAAAAAAAAPDw84rw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx
9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/v8fOEAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA

AA

AADy8vIo7u/y+/Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/+7v8vnw8PAkAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAO/v8rbw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw87EAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA

AA
AADw8PRH8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//+/v80IAA
AAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAO7x8svw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx
9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7+/zxwAAAAAAA
AAAAAAA

AA
AADv8vVQ8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9nbXT/mp+k/+Hj5//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u8fLK
AAAA

AA
AADv7/ND8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/xsjN/2FocP9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/Y2lx/8jLz//w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u8vI/AAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAO/v8rbw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9
P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/9HT1/9dZGz/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/XWVs/9TX2v/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT
/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

/+/v87MAAA
Dw8PAk7u/y/vDx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/dHqC/1hfZ/9YX2f/WF9n/2JpcP+UmJ//oqWr/6Klq/+ipav/o
qWr

/6Klq/+ipav/oqWr/6Klq/+ipav/oqWr/6Klq/+ipav/oqWr/6Klq/+ipav/oqWr/6Klq/+ipav/oqWr

/6Klq/+ipav/oqWr/6Klq/+ipav/oqWr/6Klq/+ipav/oqWr/6Klq/+ipav/oqWr/6Klq/+ipav/oqWr

/6Klq/+ipav/oqWr/6Klq/+ipav/oqWr/6Klq/+ipav/oqWr/6Klq/+ipav/oqWr/6Klq/+ipav/oqWr

/6Klq/+ipav/oqWr/6Klq/+ipav/oqWr/6Klq/+ipav/oqWr/5OZn/9hZ2//WF9n/1hfZ/9YX2f/eH6
F

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7u/y/vDw9yIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAPDx84jw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/8zP1P9YX2f/WF9n

/1hfZ/9wdXz/4uTn//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+Hi5v9tc3r/WF9n/1hfZ/9YX2f/0NPW//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/7/HzhQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD///8F7u/06PDx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/qK2x/1hfZ/9YX2f/WF9n/8PGyv/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/8DCx/9YX2f/WF9n

/1hfZ/+sr7T/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/v7/Lm///BAAAAAAAAA
AAAAAA

AAAAAAAAAAAAAAAAAPHx9Evw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//D
x9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P+boKX/
WF9n

/1hfZ/9YX2f/4+To//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/4OHI/1hfZ/9YX2f/WF9n/56jqP/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8PRIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA7
+/yovDx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/5qfpP9YX2f/WF9n/1hfZ//k5en/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/h4+f/W
F9n

/1hfZ/9YX2f/nKGn//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw854AAAA
AAAAA

AAAAAAAAAAAAAAAAAAP7+/gfw8fTw8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/m
p+k

/1hfZ/9YX2f/WF9n/+Tl6f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+Hj5/9YX2f/WF9n/1hfZ/+coaf/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/7vH07v///wUAAAAAAAAAAAAAAAAAAAAAAAAA8PD0R/
Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P+an6T/WF9n/1hfZ/9YX2f/5OXp//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/4ePn

/1hfZ/9YX2f/WF9n/5yhp//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7+/z

QwAAAAAAAAAAAAAAAAAAAAAAAAADw8PSQ8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/5qfpP9YX2f/WF9n/1hfZ//k5en/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/h4+f/WF9n/1hfZ/9YX2f/nKGn//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8PSMAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
O7v9Nnw8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/mp+k/1hfZ/9YX2f/WF9n/+Tl6f/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

/+Hj5/9YX2f/WF9n/1hfZ/+coaf/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

/+/v9NUAAAAAAAAAAAAAAAAADs9fUb8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9
P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P+an6T/WF9n/1hfZ/9YX2f/5OXp//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/4ePn/1hfZ/9YX2f/WF9n/5yhp//w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Pz8xcAAAAAAAAAAAPDw81bw8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/5qfpP9YX2f/WF9n/1hfZ//k5en/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/h4+f/WF9n/1hfZ/9YX2f/nKGn//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/7+/yUgAAAAAAAAAAAA8PD0kPDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/mp+k/1hfZ/9YX2f/WF9n/+Tl6f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/7O/0/+zv9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/9jn9f+t0/j/y+D2/+7w9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+Hj5/9YX2f/WF9n/1hfZ/+coaf/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8PSNAAAAAAAAAAADv7/PG8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P+an6T/WF9n/1hfZ/9YX2f/5OXp

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/9/q9f/C3Pb/pM/4/4fC+f9vuPv/Uq
v8

/zCb/v8rmf7/PqH9/1as/P91ufr/jcX5/63T+P/B3Pb/yN/2/8Lc9v+v1Pf/kcb5/2m1+/9FpP3/iMP
5

/+Pr9f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/4ePn/1hfZ/9YX2f/WF9n/5yhp//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dw9MMAAAAA///AfDx9PLw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8
fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0/5qfpP9YX2f/WF9n/1hfZ//k5en/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/4+v1/8vg9v/H3/b/1+b1/+x9P/w8fT/8PH0/9jm9f+m0Pj/brf7/y+a/v8ek///HpP/

/x6T//8ek///HpP//x6T//8ek///arX7/97p9f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/h4+f/WF9n/1hfZ/9YX2f/n
KGn

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH07wAAAADw8
Pci8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/mp+k/1hfZ/9YX2f/
WF9n

/+Tl6f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/b5/X/hMH6/z6i/f8ek///HpP//x6T//8ek///LJn+

/2i0+/+82ff/8PH0//Dx9P/w8fT/7O/0/7LV9/9arfz/H5P//x6T//8ek///HpP//4fC+f/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+Hj5/9YX2f/WF9n/1hfZ/+coaf/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/7u72HvHx9Ew8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P+an6T/WF9n/1hfZ/9YX2f/5OXp//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/t8PT/isP5

/yWW//8ek//HpP//x6T//8fk//IpX//x6T//8ek//HpP//x6T//9arfz/2ef1//Dx9P/w8fT/8PH0

//Dx9P/I3/b/VKv8/x6T//8ek//IZT//5/M+P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/4ePn/1hfZ/9YX2f/WF9n

/5yhp//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

bfDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/5qfpP9YX2f/WF9n

/1hfZ//k5en/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/5+30/1mt/P8ek///HpP//z6h/f+VyPn/0OL2/+3w9P/w8fT/3un1

/6zT+P9dr/z/H5P//x6T//8wm/7/xd72//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/5zM+P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/h4+f/WF9n/1hfZ/9YX2f/nKGn//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw8mnw8PSN8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/mp+k/1hfZ/9YX2f/WF9n/+Tl6f/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+7w9P9arvz/HpP/

/yGU//+Oxvn/7fD0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P+82ff/Np3+/x6T//8smf7/00P2

//Dx9P/w8fT/8PH0//Dx9P/w8fT/yN/2/zGc/v8ek//IZT//7XX9//w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+Hj5/9YX2f/WF9n

/1hfZ/+coaf/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PDz

ivDx9Kvw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P+an6T/
WF9n

/1hfZ/9YX2f/5OXp//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/jsb5/x6T//8hIP//rdP4//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/a5/X/Op+/x6T//9Hpf3/7fD0//Dx9P/w8fT/8PH0//Dx9P/w8fT/zuL
2

/yyZ/v8ek///L5r+/93o9f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH
0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/4ePn/1hfZ/9YX2f/WF9n/5yhp//w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u8fSp8PD0wPDx9P/w8fT/8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/5qfpP9YX2f/WF9n/1hfZ//k5en/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/97p9f8mlv//Hp
P/

/4jD+f/w8fT/8PH0//Dx9P/w8fT/zeH2/5XI+f+Pxfn/ttf3/+7w9P/w8fT/8PH0//Dx9P/K4fb/IpX/

/x6T//+gzPj/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/tNb3/x6T//8ek///a7X7//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/h4+f/W
F9n

/1hfZ/9YX2f/nKGn//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9L7w8PTU8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/m
p+k

/1hfZ/9YX2f/WF9n/+Tl6f/w8fT/8PH0//Dx9P/w8fT/4er1/+ju9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/jMX5/x6T//82nv7/6+70//Dx9P/w8fT/7vD0/3C4+/8ek//HpP
/

/x6T//8ek//RKT9/9Pk9v/w8fT/8PH0//Dx9P93u/r/HpP//0Sj/f/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/bLb7/x6T//8glP//j8X5/8jf9v/v8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+Hj5/9YX2f/WF9n/1hfZ/+coaf/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PD00u/v8ubw8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P+an6T/WF9n/1hfZ/9YX2f/5OXp//Dx9P/w8fT/8PH0

//Dx9P/w8fT/mMr5/9/q9f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P9Kpv3/HpP/

/4jD+f/w8fT/8PH0//Dx9P/w8fT/w932/zCb/v8ek///HpP//x6T//8ek///Op/+ft9P/w8fT/8PH0

/87i9v8ek///HpP//9Tl9f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/J4Pb/HpP//x6T//8ek///HpP/

/y+a/v9ns/v/os74/93o9f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/4ePn

/1hfZ/9YX2f/WF9n/5yhp//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8PLl8PH08PDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/5qfpP9YX2f/WF9n/1hfZ//k5en/8PH0//Dx9P/w8fT/8PH0//Dx9P/m7PX/YLH8/83h9v/w8fT/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/7fD0/yKV//8ek///v9v3//Dx9P/w8fT/8PH0//Dx9P/w8fT/mcv
5

/x6T//8ek///HpP//x6T//8ek///lcj5//Dx9P/w8fT/8PH0/zOd/v8ek///qdL4//Dx9P/w8fT/8PH0

//Dx9P/o7vT/j8X5/yyZ/v8ek///HpP//x6T//8ek///HpP//x6T//8ek///IJT//02o/f+Rxvn/1+b1

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/h4+f/WF9n/1hfZ/9YX2f/nKGn//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+7x9O7u7/L38PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/mp+k/1hfZ/9YX2f/WF9n/+T16f/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/X5vX/PaH9/6XP+P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/a5/X/
HpP/

/x6T///c6PX/8PH0//Dx9P/w8fT/8PH0//Dx9P+82ff/HpP//x6T//8ek///HpP//x6T//9gsfz/8PH0

//Dx9P/w8fT/UKr8/x6T//+Tx/n/8PH0//Dx9P/j6/X/jsb5/zCb/v8ek///HpP//x6T//8ek///HpP/

/x6T//8ek///HpP//yCU//9EpP3/XrD8/1uu/P9Eo/3/Uqv8/5/M+P/o7vT/8PH0//Dx9P/w8fT/8PH0

/+Hj5/9YX2f/WF9n/1hfZ/+coaf/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PHy9e7v8v3w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P+an6T/WF9n/1hfZ/9YX2f/5OXp//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/Q4vb/M53+

/2i0+//h6vX/8PH0//Dx9P/w8fT/8PH0/9jm9f8ek///HpP//9/q9f/w8fT/8PH0//Dx9P/w8fT/8PH0

/4jD+f8ek///HpP//x6T//8ek///HpP//1yv/P/w8fT/8PH0//Dx9P9Sq/z/HpP//5LG+f/F3vb/c7n7

/yiX/v8ek///HpP//x6T//8ek///HpP//x6T//8ek///HpP//ymY/v+Gwfr/3Oj1//Dx9P/w8fT/8PH0

//Dx9P/p7fT/t9j3/4fC+v+czPj/7vD0//Dx9P/w8fT/4ePn/1hfZ/9YX2f/WF9n/5yhp//w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u7/L87u/y/fDx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/5qfpP9YX2f/WF9n/1hfZ//k5en/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/U5fX/QaP9/y+a/v+czPj/7O/0//Dx9P/w8fT/6e30

/x+T//8ek///x9/2//Dx9P/w8fT/8PH0/+Xs9f+by/j/I5b//x6T//8ek///HpP//x6T//8ek///icP5

//Dx9P/w8fT/8PH0/zif/v8ek///KJj+/x6T//8ek///HpP//x6T//8ek///HpP//x6T//8ek///HpP/

/x+T//9ut/v/3un1//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+7w9P/s7/T/8PH0

//Dx9P/h4+f/WF9n/1hfZ/9YX2f/nKGn//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+7v8vzu7/L38PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/mp+k/1hfZ/9YX2f/WF9n/+Tl6f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/17PX/YLH8/x6T//9Bo/3/rNP4/+7w9P/w8fT/PqH9/x6T//+Tx/n/8PH0//Dx9P/w8fT/YLH8

/x6T//8ek//HpP//x6T//8ek//HpP//y2a/v/e6fX/1eX1/5TI+f9Pqfz/HpP//x6T//8ek//HpP/

/x6T//8ek//HpP//x6T//8ek//HpP//x6T//9Kpv3/wt32//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+Hj5/9YX2f/WF9n/1hfZ/+coaf/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PHy9fDx9PDw8fT/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P+an6T/WF9n/1hfZ/9YX2f/5
OXp

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/n8z4/ymY/v8ek///OJ/
+

/4TB+v9brvz/HpP//0Gj/f/v8fT/8PH0//Dx9P/k7PX/Uqr8/x6T//8ek///HpP//x6T//8ek///SKb9

/z+i/f8ek///HpP//x6T//8ek///HpP//x6T//8ek///HpP//x6T//8ek///HpP//x6T//8nl///lsn5

/+7w9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/4ePn/1hfZ/9YX2f/WF9n/5yhp//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fTv7+/y5vDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/5qfpP9YX2f/WF9n/1hfZ/k5en/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/3Oj1/2y2+/8fk//HpP//x6T//8ek//HpP//1qt/P92uvr/fr/6

/32++v9yuPv/LJn+/x6T//8ek//HpP//x6T//8ek//HpP//x6T//8ek//HpP//x6T//8ek//HpP/

/x6T//8ek//HpP//x6T//8ek//WK38/9X19f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/h4+f/WF9n/1hfZ/9YX2f/nKGn

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw8uXw8PTU8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/mp+k/1hfZ/9YX2f/WF9n

/+Tl6f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/83h9v9ns/vIpX//x6T//8ek//HpP//x6T//8ek//HpP//x6T//8ek//HpP//x6T//8ek//HpP/

/x6T//8ek//HpP//x6T//8ek//HpP//x6T//8ek//HpP//x6T//8ek//QqT9/67T9//w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+Hj5/9YX2f/WF9n/1hfZ/+coaf/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH00vDw9MHw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P+an6T/WF9n/1hfZ/9YX2f/5OXp//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/b5/X/iMP5/z2h/f8ek//HpP/

/x6T//8ek//HpP//x6T//8ek//HpP//x6T//8ek//HpP//x6T//8ek//HpP//x6T//8ek//HpP/

/x6T//8llv//YrH7//7TX9//v8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/4ePn/1hfZ/9YX2f/WF9n

/5yhp//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fS+8PH0

q/Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/5qfpP9YX2f/WF9n

/1hfZ//k5en/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/8ng9v+Mxfn/Vqz8/ymY/v8ek///HpP//x6T//8ek///HpP/

/x6T//8ek///HpP//x6T//8ek///H5P//0Gj/f91ufr/rtP3/+bt9f/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/h4+f/WF9n/1hfZ/9YX2f/nKGn//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+7x9Knw8PSO8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/mp+k/1hfZ/9YX2f/WF9n/+Tl6f/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/7/H0/9Lk9v+01vf/mcv5/4jD+f98vfr/ebz6/3y++v+KxPn/oMz4/7/b9//l7PX/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/4ePn/1hfZ/9YX2f/WF9n/5yhp//w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8Pjp8fH0TPDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/5qfpP9YX2f/WF9n/1hfZ//k5en/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/h4+f/WF9n

/1hfZ/9YX2f/nKGn//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dw9Ejw8Pci8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0
p+k

/1hfZ/9YX2f/WF9n/+Tl6f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+Hj5/9YX2f/WF9n/1hfZ/+coaf/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7u72H///wHw8fTz8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P+an6T/WF9n/1hfZ/9YX2f/5OXp//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/4ePn

/1hfZ/9YX2f/WF9n/5yhp//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9PAAAAAAAAAAAAO/v88bw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/5qfpP9YX2f/WF9n/1hfZ//k5en/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/h4+f/WF9n/1hfZ/9YX2f/nKGn//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7+/0xAAAAAAAAAAAAA7+/0kfDx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/mp+k/1hfZ/9YX2f/WF9n/+Tl6f/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/+Hj5/9YX2f/WF9n/1hfZ/+coaf/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8PSOAAAAAAAAAAADw8PNX8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P+an6T/WF9n/1hfZ/9YX2f/5OXp//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/4ePn/1hfZ/9YX2f/WF9n/5yhp//w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+/v8IMAAAAAAAAAAAAOzs9Rz w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//5qfpP9YX2f/WF9n/1hfZ//k5en/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/h4+f/WF9n/1hfZ/9YX2f/nKGn//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/9PT0GAAAAAAAAAAAAAAAAAAO7v9Nrw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/mp+k/1hfZ/9YX2f/WF9n/+Tl6f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+Hj5/9YX2f/WF9n/1hfZ/+coaf/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+/v9NYAAAAAAAAAAAAAAAAAAAAAAAAA7+/0

kfDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P+an6T/WF9n/1hfZ/9YX2f/c
HZ9

/3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh/

/3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh/

/3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh/

/3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh//3F4f/9xeH//cXh/

/3F4f/9xeH//cHV8/1hfZ/9YX2f/WF9n/5yhp//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PD0jQAAAAAAAAAAAAAAAAAAAAAAAAADx8fRJ8PH0//Dx9P/w8fT/8PH0//D
x9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0/5qfpP9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
nKGn

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8PNFAAAAAAAAAAAAA
AAAAAAAAA

AP7+/gfw8fTy8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/mp+k/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/+coaf/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH07////wYAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAO/x9KTW8fT/8PH0
//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P+an6T/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9ZYGj/Y2px/2Nqcf9janH/
Y2px

/2Nqcf9janH/Y2px/2Npcf9YX2f/WF9n/1hfZ/9gZm7/Z251/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/5yhp//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8PKgAAAAAAAAAAAA
AAAAAAAAAA

AAAAAAAAAAAAA7vH1TfDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P
/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/5ugpf9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/XmRt/9fY3P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/8vM0f9ZYGj/
anB4

/+Xn6v/v8PP/kJSa/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/nqKo//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Hx9EkAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD///8F7u/
06fDx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/qKyx/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9zeYD/8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/2JpcP+JjpT/8PH0//Dx9P+4u8D/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/+rrrT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/v7/Ln///BAAAAAAAAA
AAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAADw8POK8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0
//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/MztP/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/+prbL/zc7T/83O0//NztP/zc7T/83O0//NztP/zc7T/83O0/+coaf/WF9
n

/15kbf+/w8f/0NPW/3R5gf9YX2f/WF9n/1hfZ/9YX2f/WF9n/8/R1f/w8fT/8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0/+/x84cAA
AAAAAPHx8SXw8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+/w8/9yeYD/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+v87QAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA8PDz

RPDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/8PGyv9gZm7/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/2Fnb//GyM3/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7+/zQAAAAAAAAA
AAAAAAAA

AA8PHy0PDx9P/
w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0/97g4/+VmaD/Ymlw/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
/WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/2Nqcf+Xm6H/4OHk//Dx9P/w8fT/8PH0//Dx9P/w8f
T/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx8swAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAADv7/JS8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8f
T/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/v8PP/4e
Lm

/9vd4f/a3OH/2tzh/9rc4f/a3OH/2tzh/9rc4f/a3OH/2tzh/9rc4f/a3OH/2tzh/9rc4f/a3OH/2tzh

/9rc4f/a3OH/2tzh/9rc4f/a3OH/2tzh/9rc4f/a3OH/2tzh/9rc4f/a3OH/2tzh/9rc4f/a3OH/2tzh

/9rc4f/a3OH/2tzh/9rc4f/a3OH/2tzh/9rc4f/a3OH/2tzh/9rc4f/a3OH/2tzh/9rc4f/a3OH/2tzh

/9rc4f/a3OH/2tzh/9rc4f/a3OH/2tzh/9rc4f/a3OH/2tzh/9rc4f/a3OH/2tzh/9vd4f/h4ub/7/Dz

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7vH1TQAAAAAAAA
AAAAAAA

AAAP///w
Hw8fLO8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+7x8soAAA
AAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAPHx9Erw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PDzRQAAAAAAAA
AAAAAAAA

AA
AAAAAAAAAAAAAAAAO7x8rnw8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u7/L68fHxJgAAAAAAAAA
AAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA8PD0

jfDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx84gAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAADn5/4L8PHz3vDx9P/w8
fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u8fTb///CAAAAAAAAAA
AAAAAA

AA
AA

AAAAAADw8PNF8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/7u/y/u/v80IAA
AAAAAAAAAAAAAAAAAAAAAAAA

AA
AADv8fSV8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8PSQAAAAAAAAAAAA
AAAAAAAAAA

AA
AA

AAAAAAAAAAAAAAAAAAAAAAAAAOL+/gnv7/TW8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//D
x9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PD00v7+/gcAA

AAO7u8y7w8PL28PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx8/Tt8/MrAAAAAAAAAAAAA
AAAAAAAA

AA
AA

AAAPhx81zw8fT/8PH0//Dx9P/w8fT/
8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PDzVwAA

AA
AA

AO/v9JHw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw9IwAAAAAAAAAAAAAAAA
AAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA///A+/v8rXw8
fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8POx///AgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAA6enpDPDx8s7w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P
/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7vHyy+Xl/woAAAAAAAAAAAAAAAA
AAAAAAAAAAAA

AA
AA

AA
AAAAAAAAA9PT0GO7v9Nrw8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

/+7v9Njz8/MWAA
AA

AA
AA

AA7e32HfDx8uDw8fT/8PH0//Dx9P/w8fT/8P
H0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fPe7PX1GwAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA

AA
AA

AA
AA8PD3

IvDx8uDw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PHz

3u/v9yEAA
AA

AA
AA

AAA7e32He7v9Nr
w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+7v9Njs9fUbAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAA9PT0GPDx8s7w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9
P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fLM
8/Pz

FgAA
AA

AA
AA

AA
AAAAAAA6enpDO/v8rXw8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PDzseXl/woAA

AA

AA

AA///A+/v9JHw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw9Iz///8CA
AAA

AA
AA

AA
AA

AA
AA

APHx81zw8PL28PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx8/Tw8PNXAA

AA

AA

AAO7u8
y7v7/TW8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8PTS7fPzKwAAAAAA
AAAAAAA

AA
AA

AA
AA

AA
AA

AAAAAAAAAAAAAAAAAAAAAOL+/gnv8fSV8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/u7/L+7+/0kf7+/gcAA

AA

AA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAADw8PNF8PHz

3vDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7vH03O/v80IAAAAAAAAAAAAAAAAA
AAAAAAAAAA

AA
AA

AA
AA

AA
AA

AADI5f8K8PD0je7v8vww8f
T/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7
u/y

+vDw84ri/v4JAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAA8vLyKu7x8rnw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P
/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+v8rby8vIoAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA

AA
AA

AA
AA

AA
AA

AA
AAAAAAAAAAAAAAAAAPh9Erw8fLO8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+7x8svw8PRH
AAAA

AA
AA

AA
AA

AA
AA

AA
AA

AAAP///wHv7/JS8PHy0PDx9P/w8fT/8
PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx8s7v8vVQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA

AAA
AAAAAAAAAAAAAAAAAAAAAAAA

AAA
AAAAAAAAAAAAAAAAAAAAAAAA

AAA
AAAAAAAAAAAAAAAAAAAAAAAA

AAA
AAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAA8PDzRO7x8rjw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P
/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7u/y/u/v8rbv7/NDAAAAAAAAAAAAA
AAAAAAA

AA
AA

AA
AA

AA
AA

AA
AA

AA
AAAAAAAAAAAAAAAAAAAAAPHx8SXw8POK7u/0

6fDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7
u/0

6PDx84jw8PAkAA
AA

AA
AA

AA
AA

AA
AA

AA
AA

AAAD///8F7vH1Te/x9KTw8f
Tx8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fTw7+/yovHx9Ev///8FAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA

AA
AA

AA
AA

AA
AA

AA
AA

AA
AA

AA
AA
P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+7v9Nnw8PSQ8PD0R/7+/gcAAA
AAAAAA

AA
AA

AA
AA

AA
AA

AA
AA

AA
AA

AA
AA

AOzs9Rzw8PNX7+/0ke/v88bw8fTz8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8f
T/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9PLv7/PG
8PD0

kPDw81bs9fUbAA
AA

AA
AA

AA
AA

AA
AA

AA
AA

AA
AA

AA
AAAAAAAAAAAAAAAAAAAAAP///wHw8Pci8fH0

TPDw823w8PSO8PH0q/Dw9MHw8PTU7+/y5vDx9PDu7/L37u/y/e7v8v3u7/L38PH08O/v8u
bw8PTU8PD0

wPDx9Kvw8PSN8PDzbfHx9Evw8Pci////AQAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AA

AA
AA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAD/////////4

AAAf//////////AAAAAP//////////8AAAAAP//////////4AAAAAAf//////////

//4AAAAAAB//////////4AAAAAAAH//////////4AAAAAAAf//////////4AAAAAAA
B///

////4AAAAAAAH//////////4AAAAAAAf//////////8AAAAAAD//////////8AAA
AAAAAA

AAP////////+AAAAAAB//////////AAAAAAP//////////AAAAA
A////////gAAAA

AAAAAAH////////wAAAAA//////////4AAAAAAH////////8AAAA
AAAAAA////////+A

AAAAAAH////////AAAAA//////////gAAAAAAH//
//wAAAAA/

//8AAAAAAP///+AAAAAAB//////AAAAA
AAAAP///gAAAAA

AAAAB//4AAAAAAf//8AAAAAAD//+AAAAA
AAAAAAf//gAAAAA

AAAAAAH//wAAAAA//8AAAAAAP/+
AAAAAAB//gAA

AAAAAAAAAAAAAAAAAAf/wAAAAAAAAAAAAAAAAAAD/8AAAAAAAAAAAAAAAA
AAAAA/+AAAAAAAAAAAAAAAAAAAA

H/gAAAAAAAAAAAAAAAAAAAB/wAAAAAAAAAAAAAAAAAAAAAP8AAAAAAAA
AAAAAAAAAAD+AAAAAAAAAAAA

AAAAAAfgAAAAAAAAAAAAAAAAAAAH4AAAAAAAAAAAAAAAAAAAAAB8AAA
AAAAAAAAAAAAAAAAAAAAAPAAAAAAAA

AAAAAAAAAADwAAAAAAAAAAAAAAAAAAAA8AAAAAAAAAAAAAAAAAAAA
AAOAAAAAAAAAAAAAAAAAAAAABgAAA

AAAAAAAAAAAAAAAAAAAYAAAAAAAAAAAAAAAAAAAAAGAAAAAAAAAAAA
AAAAAABAAAAAAAAAAAAAAAAAAAA

AQAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AA

AA
AA

AAAAAAAAAAAAABgAAAAAAAAAAAAAAAAAAAAAAAAAYAAAAAAAAAAAAAAAAAAAA
AAGAAAAAAAAAAAAAAAAAAAAAAAAABgAAA

AAAAAAAAAAAAAAAAAAAAcAAAAAAAAAAAAAAAAAAAAAPAAAAAAAAAAAAAAAAAA
AAAAAADwAAAAAAAAAAAAAAAAAAAA

A8AAAAAAAAAAAAAAAAAAAAAPgAAAAAAAAAAAAAAAAAAAAH4AAAAAAAAAAAA
AAAAAAAAAAAAAB+AAAAAAAAAAAAAAAAAAAA

AAAAAAfwAAAAAAAAAAAAAAAAAAAAAP8AAAAAAAAAAAAAAAAAAAAAD/gAAA
AAAAAAAAAAAAAAAAAAAAAB/4AAAAAAAAAAAA

AAAAAAf/AAAAAAAAAAAAAAAAAAAAAP/wAAAAAAAAAAAAAAAAAAAAAD
/8AAAAAAAAAAAAAAAAAAAAAB//gAA

AAAAAAf/8AAAAAAAAAAAAAAAAAAAAAP//AAAAAAAAAAAAAAAAAAAA
AAD//4AAAAAAAAAAAAAAAAAAAAAB

//+AAAAAAAAAAAAAAAAAAAAf//wAAAAAAAAAAAAAAAAAAAAAP//+AAAAAAAAAAAA
AAAAAH//gAAAAAAAAAAAA

AAAAB//8AAAAAAAAAAAAAAAAA///gAAAAAAAAAAAAAAAAAf//8AAAAAAAA
AAAAAAAAAP///AAAAAA

AAAAAAAAAD///4AAAAAAAAAAAAAAAAAB/////AAAAAAAAAAAAAAAAA////4AAA
AAAAAAAAAAAAAf/////A

AAAAAAAAAAAAAP/////4AAAAAAAAAAAAAAH/////AAAAAAAAAAAAAD/////4
AAAAAAAAAAAAAB//

////AAAAAAAAAAAAA/////8AAAAAAAAAAAAA/////gAAAAAAAAAAAAAf/////8
AAAAAAAAAA

AAP////////wAAAAAAAAAAAP////////+AAAAAAAAAAH////////4AAAAAAAAAAH////////
//gAAA

AAAAAH////////8AAAAAAAH////////4AAAAAAAH////////gAAAAAAH////////
/

//+AAAAAAH////////8AAAAAP////////8AAAA////////8AAD////////

/ygAABAAAAAgAAAAEAIAAAAAAAAEAAABMLAAATCwAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAD09PQZ7+/yU+/x84Pw8fSs8PHyzfDw8uTw8fL17u/
y/O7v8vzw8fP08PHy

4/Dx8szw8fSr7/Hzgu/v8IP09PQYAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAD///8B8fH2OfDw9I3v7/TX8PH0//Dx9P/w8fT/8PH0//
Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+v9NXw8PKL8fH1NwAAAAA
AAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAA07u7g/v8fR07u/02vDx9P/w8fT/8
PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u7/Tr8P
D1

Zv//wEAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAADx8fE17+/z

xvDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8PTD8PDwIwAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAADx8fNc8PHz9PDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w
8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9PPw8PNZAAAAAAAAAAAA
AAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAP///wLw8PSP8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx
9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dw8ov///8CAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAP7+/gfw8POu8PH0//Dx9P/w8fT/
8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0qv///wYAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA

AAAP7+/g
fv8fK38PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/v7/K1///BgAAAAAAAAA
AAAAAA

AA
AAAAAAAAAAAAAAAAAAAP///wLw8POv8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9Kr//8BAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAADv7/SR8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8
fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PDyiwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAADu8fRe8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//
Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8PNZAAAAAAAAAAAA
AAAAAAAA

a/Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw9WYAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAA8PDwEe7x9O3w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u7/Tq7u7u

DwAAAO/x84fw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/6Ort/+fo6//n6Ov/5+jr/+fo6//n6Ov/5+jr/+fo6//n6Ov/5+jr

/+fo6//n6Ov/5+jr/+fo6//n6Ov/5+jr/+fo6//n6Ov/5+jr/+fo6//n6Ov/5+jr/+fo6//n6Ov/5+jr

/+fo6//n6Ov/5+jr/+fo6//p6u7/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+/x84IAAAAAAAAAAAAAAAAAAAAAAAAAAAAO/v7xDw8fTz8PH0

//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P+/wsf/bHJ6/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/21zev/Bw8j/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fTx7OzsDgAAAAAAAAAAAAAAAAAAAAAADv8vR28PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/KzdH/WWBo/1tiaf94foX/fIGH

/3yBh/98gYf/fIGH/3yBh/98gYf/fIGH/3yBh/98gYf/fIGH/3yBh/98gYf/fIGH/3yBh/98gYf/fIGH

H

/3yBh/98gYf/FIGH/3yBh/98gYf/FIGH/3yBh/98gYf/FIGH/3yBh/98gYf/FIGH/3h9hf9aYWn/W
mFp

/83Q1P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+/x83MAAA
AAAAAA

AAAAAAD///8B7vH03PDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8
fT/iY6U

/1hfZ//Bw8j/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/vcHF/1hfZ/+NkZf/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/u7/TZAAAAAAAAAAAAAAAAAA8vLyO/Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9
P/w8fT/8PH0

//Dx9P/w8fT/8PH0/3p/hv9YX2f/6+zv//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+jq7f9YX2f/e4KI//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Hx9TcAAAAAAAAAAAPDw9JDw8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P95f4b/WF9n/+vs7//w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/p6u7/WF9n/3yBh//w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8PKLAAAAAAAAAAADu8fTb8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/eX+G/1hfZ//r7O//8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/6eru

/1hfZ/98gYf/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7+
/0

1QAAAADs7PUc8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8
fT/8PH0

/3l/hv9YX2f/6+zv//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0/+nq7v9YX2f/flGH//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/09PQY8PDzVfDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT
/8PH0

//Dx9P/w8fT/8PH0//Dx9P95f4b/WF9n/+vs7//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0/+zv9P/Q4/b/tNb3/5jK+f+VyPn/rtP3/8ng9v/c6PX/1uX1/7fY9/+lz/j/0+T2

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/p6u7/WF9n/3yBh//w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7+/yU+/x84bw8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/eX+G/1hfZ//r7O//8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/9Di9v+Nxfn/drr6/4bB+v/C3fb/1eX1/5zM+P9Pqfz/HpP/

/x6T//+Avvr/7O/0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/6eru/1hfZ/98gYf/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+x84Lw8POv8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/3l/hv9YX2f/6+zv

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7fD0/3u9+v8nl///Z7P7/4fC+v9wuPv/LZr+

/2Gx+//m7fX/8PH0/77a9/8+ov3/QaP9/9zo9f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/+nq7v9YX2f/fIGH//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fSr8PHyz/Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P95f4b/WF9n/+vs7//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/3u9+v9CpP3/1+b1

//Dx9P/w8fT/8PH0/+Pr9f9Zrfz/Wq38//Dx9P/w8fT/5uz1/1Cq/P9Kpv3/7O/0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/p6u7/WF9n/3yBh//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PHyzO/v8ubw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/eX+G/1hfZ//r7O//8PH0/+zv9P/u8PT/8PH0//Dx9P/w8fT/8PH0

/9Dj9v8ll///1eX1//Dx9P+UyPn/Wa38/4TB+v/q7vT/5u31/zOd/v+y1ff/8PH0//Dx9P/g6vX/MJv+

/4XB+v/n7fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/6eru/1hfZ/98gYf/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx8uPw8PL28PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/3l/hv9YX2f/6+zv//Dx9P/w8fT/sN
T3

/+ju9P/w8fT/8PH0//Dx9P+Rxvn/YbH7//Dx9P/w8fT/ncz4/x6T//8ek///d7v6//Dx9P+DwPr/b7j
7

//Dx9P/w8fT/1uX1/0yn/f8ek///I5b//1Kq/P+Rxvn/0+T2//Dx9P/w8fT/8PH0//+nq7v9YX2f/fIG
H

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fP07
u/y

/fDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P95f4b/W
F9n

/+vs7//w8fT/8PH0//+nu9P95vPr/zOD2//Dx9P/w8fT/ebz6/3+//+v/w8fT/8PH0/8ng9v8ek///HpP/
/z+i/f/w8fT/n8z4/1qt/P/I3/b/brf7/yKV//8ek///HpP//yGU//9otPv/oMz4/5/M+P+jz/j/vdr3

//Dx9P/p6u7/WF9n/3yBh//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/7u/y/O7v8v3w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/eX+G/1hfZ//r7O//8PH0//Dx9P/w8fT/6e30/2u2+/96vPr/3+r1/4vE+f9otPv/8PH0

/8ng9v8/ov3/HpP//x6T//9tt/v/0uT2/2Wy+/8ilf//HpP//x6T//8ek///HpP//1Gq/P/L4Pb/8PH0

//Dx9P/w8fT/8PH0//Dx9P/v8fT/6eru/1hfZ/98gYf/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+7v8vzw8PL28PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/3l/hv9YX2f/6+zv//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/zee/v9Hp3/J5f//63T+P+01vf/QaP9/x6T//8ek///MZz+/x6T//8ek///HpP//x6T//8ek///LJn+

/6DN+P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+nq7v9YX2f/fIGH//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fL17+/y5vDx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P95f4b/WF9n/+vs7//w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/n7fT/k8f5/z6i/f8ek///HpP//x6T//8ek///HpP//x6T//8ek///HpP/

/x6T//8wm/7/gL76/9/q9f/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/p6u7/WF9n

/3yBh//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PDy

5PDx8s/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/e
X+G

/1hfZ//r7O//8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/zOD2/5XI+f9ut/v/Vav
8

/0qm/f9Pqfz/ZLL7/4zF+f+/2/f/7fD0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH
0

//Dx9P/w8fT/6eru/1hfZ/98gYf/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx8s3w8POv8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/8PH0/3l/hv9YX2f/6+zv//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+nq7v9YX2f/fIGH//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fSs7/Hzh/Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P95f4b/WF9n/+vs7//w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/p6u7/WF9n/3yBh//w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7/Hzg/Dw81bw8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/eX+G/1hfZ//r7O//8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/6eru

/1hfZ/98gYf/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/+/v8IPu7vYe8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

/3l/hv9YX2f/6+zv//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0/+nq7v9YX2f/fIGH//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/09PQZAAAAA07x9Nzw8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P95f4b/WF9n/6+zuf+ytrr/sra6/7K2uv+ytrr/sra6/7K2uv+ytrr/sra6

/7K2uv+ytrr/sra6/7K2uv+ytrr/sra6/7K2uv+ytrr/sra6/7K2uv+ytrr/sra6/7K2uv+ytrr/sra6

/7K2uv+ytrr/sra6/7K2uv+ytrr/sra6/7K2uv+usbf/WF9n/3yBh//w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/v7/TXAAAAAAAAAAADv7/SR8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/eX+G/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/98gYf/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0jQAAAAAAA
AAA7vLy

PfDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/3p/hv9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/95f4b/qK2x/6itsf+orbH/qK2x/3N5gP+Ch47/jp
KZ

/1hfZ/9YX2f/e4KI//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8P
H0

//Hx9jkAAAAAAAAAAAP///wHu8fPd8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8
fT/8PH0

//Dx9P+IjpT/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/mp+k/+Dh5P/g4eT/
4OHk

/+Dh5P+Rlpz/p6uw/7u+w/9YX2f/WF9n/4yRl//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/8PH0/+7v9Nr///8BAAAAAAAAAAAAAAAAAAAA7+/0d/Dx9P/w8fT/8PH0//Dx9P/
w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/x8rO/1lgaP9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1lgaP/LzdL/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/v8fR0AAAAAAAAAAAAAAAAAAAAAAAAAAPD
w8BHw8fP08PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P+7vsP/aG93/1hfZ/9YX2f/W
F9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/
WF9n

/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/1hfZ/9YX2f/WF9n/2pweP++wc
b/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fTy7u7uDwAAAAAAA
AAAAAAA

AAAAAAAAAAAAA8PDzifDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/
w8fT/8PH0

//Dx9P/m5+v/5OXp/+Tl6f/k5en/5OXp/+Tl6f/k5en/5OXp/+Tl6f/k5en/5OXp/+Tl6f/k5en/5OX
p

/+Tl6f/k5en/5OXp/+Tl6f/k5en/5OXp/+Tl6f/k5en/5OXp/+Tl6f/k5en/5OXp/+Tl6f/k5en/5OXp

/+bn6//w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7/
Hz

hAAAPhx8RPw8fTv8PH0//Dx9P/w8fT/
8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/7u/07O/v7xAAA
8fHzbvDx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dw8mkAA
AAAAAAAAAAAAAAAAAAAA

AP///wPu8fLK8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//+/v88b///8CAAAAAAAAAAAAAA
AAAAAAAA

AA8vLyKPDw8vbw8fT/8PH0//Dx9P/w8fT/8
PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx8/Tx8fE1A
AAA

AA
AADv8fRi8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/x8fNcAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AO/x9JT w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8PSPAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAD///8D8PDzsvDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//
Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8POu///AgAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAD///8D7/H0IPDx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//
Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/v7/SR///AgAA
AAAAAAAAAAAAAAAAAAAAAAAA

AA
AADv8fRh8PDy9vDx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx8/Tu8fReAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAA

APLy8iju8fLK8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/
8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+v88fx8fEmAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA

AA
AA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA///A/Hx827w8fTv8PH0//Dx9P/w8fT/8PH
0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8
PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/7vH07fDw82v///8CAAAAAAAAAAA
AAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAA8fHxE/Dw84nw8fP08PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fTz7/Hzh/Dw8BEAAA
AAAAAA

AA
AA

AA
AA

AAAAAAAAAAAAA8PDwEe/v9Hfu8fPd8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0/+7x9Nvv8vR2
7+/v

EAA
AA

AA
AA

AAA////Ae7y8j3v7/
SR7vH03PDx9P/w8fT/8PH0

//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/w8fT/8PH0//Dx9P/u8fTb8
PD0

kPLy8jv//8BAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAADu7vYe8PDzVu/x84fw8POv8PHyz+/v8ubw8PL27u/y/e7v8v3w8PL27+/y5vDx8s
/w8POv7/Hz

hvDw81Xs7PUcAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAA////AAD////////AAAB//

////wAAAA/////8AAAAA/////AAAAAA////4AAAAAB/////AAAAAAD////wAAAAAAD//+AA
AAAAAH//wAA

AAAAAP/+AAAAAAAf/4AAAAAAB//AAAAAAAD/4AAAAAAAH/AAAAAA
AAP8AAAAA/gAAAAAA

B+AAAAAAAAAHwAAAAAAAAAPAAAAAAAAA4AAAAAAAAADgAAAAAAAA
AGAAAAAAAAAYAAAAAAAAABAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAACAAAAAAAAAYAAAAAAAAABgAAAAAAAAAGAAAAAAAAAcAAA
AAAAADwAAAAAAAAAPgAAAAAAAA

B+AAAAAAAAAH8AAAAAAAA/wAAAAAAAAAD/gAAAAAAAf/AAAAAAD/+
AAAAAAAf/4AAAAAAB//wAA

AAAAAP//gAAAAAB///AAAAAAP///AAAAAD///+AAAAAf///8AAAAAD///8AA
AAA///8AAAAP/

///8AAD////////AAD///8=

</value>

</data>

</root>

Oraculo\Screens\RemoveColumnsForm.resx

?<?xml version="1.0" encoding="utf-8"?>

<root>

<!--

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"  
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET  
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
```

```
<xsd:element name="root" msdata:IsDataSet="true">
```

```
<xsd:complexType>
```

```
<xsd:choice maxOccurs="unbounded">
```

```
<xsd:element name="metadata">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element name="value" type="xsd:string" minOccurs="0" />
```

```
</xsd:sequence>
```

```
<xsd:attribute name="name" use="required" type="xsd:string" />
```

```
<xsd:attribute name="type" type="xsd:string" />
```

```
<xsd:attribute name="mimetype" type="xsd:string" />
```

```
<xsd:attribute ref="xml:space" />
```

```
</xsd:complexType>
```

```

</xsd:element>

<xsd:element name="assembly">

  <xsd:complexType>

    <xsd:attribute name="alias" type="xsd:string" />

    <xsd:attribute name="name" type="xsd:string" />

  </xsd:complexType>

</xsd:element>

<xsd:element name="data">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

      <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

  </xsd:complexType>

```

```
</xsd:element>

<xsd:element name="resheader">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" />

  </xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

  <value>text/microsoft-resx</value>

</resheader>

<resheader name="version">

  <value>2.0</value>

</resheader>
```

<resheader name="reader">

<value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

<value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<assembly alias="System.Drawing" name="System.Drawing, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />

<data name="\$this.Icon" type="System.Drawing.Icon, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">

<value>

AAABAAEAEBQaAAAAACoAQAAFgAAACgAAAAQAAAAIAAAAAEABAAAAAA
AwAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAACAAACAAAAAgIAAgAAAAIAAgACAgAAAgICAAMDAwAAAAP8A
AP8AAAD//wD/AAAA/wD/AP//

AAD///8AA
AAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAAAAAAAAAAAAAAAAAAAAAAAA

AA
AA

AA
AA

AA
cAAL8DAAC+/wAAvv8AAL8D

AAC/BwAAv88AAL//AAC//wAAv/8AAMB/AAA=

</value>

</data>

</root>

Oraculo\Screens\ReportViewer.resx

?<?xml version="1.0" encoding="utf-8"?>

<root>

<!--

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format

that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```

<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />

<xsd:element name="root" msdata:IsDataSet="true">

  <xsd:complexType>

    <xsd:choice maxOccurs="unbounded">

      <xsd:element name="metadata">

        <xsd:complexType>

          <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" />

          </xsd:sequence>

          <xsd:attribute name="name" use="required" type="xsd:string" />

          <xsd:attribute name="type" type="xsd:string" />

          <xsd:attribute name="mimetype" type="xsd:string" />

          <xsd:attribute ref="xml:space" />

        </xsd:complexType>

      </xsd:element>

    <xsd:element name="assembly">

      <xsd:complexType>

        <xsd:attribute name="alias" type="xsd:string" />

        <xsd:attribute name="name" type="xsd:string" />

      </xsd:complexType>

    </xsd:element>

  </xsd:choice>

</xsd:complexType>

</xsd:element>

```

```

</xsd:element>

<xsd:element name="data">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

      <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

  </xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

```

```
</xsd:sequence>

<xsd:attribute name="name" type="xsd:string" use="required" />

</xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

  <value>text/microsoft-resx</value>

</resheader>

<resheader name="version">

  <value>2.0</value>

</resheader>

<resheader name="reader">

  <value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

  <value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
```

</resheader>

</root>

Oraculo\Screens\VideoPlayer.resx

?<?xml version="1.0" encoding="utf-8"?>

<root>

<!--

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

<resheader name="resmimetype">text/microsoft-resx</resheader>

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,  
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,  
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a  
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"  
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
<value>[base64 mime encoded string representing a byte array form of the .NET  
Framework object]</value>
```

```
<comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a

type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture.

Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.soap.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Soap.SoapFormatter

: and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64

value : The object must be serialized into a byte array

: using a System.ComponentModel.TypeConverter

: and then encoded with base64 encoding.

-->

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

```
<xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
```

```
<xsd:element name="root" msdata:IsDataSet="true">
```

```
<xsd:complexType>
```

```
<xsd:choice maxOccurs="unbounded">
```

```
<xsd:element name="metadata">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element name="value" type="xsd:string" minOccurs="0" />
```

```
</xsd:sequence>
```

```
<xsd:attribute name="name" use="required" type="xsd:string" />
```



```

    <xsd:attribute name="type" type="xsd:string" />

    <xsd:attribute name="mimetype" type="xsd:string" />

    <xsd:attribute ref="xml:space" />

</xsd:complexType>

</xsd:element>

<xsd:element name="assembly">

    <xsd:complexType>

        <xsd:attribute name="alias" type="xsd:string" />

        <xsd:attribute name="name" type="xsd:string" />

    </xsd:complexType>

</xsd:element>

<xsd:element name="data">

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

            <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />

        </xsd:sequence>

        <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>

```

```

    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />

    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />

    <xsd:attribute ref="xml:space" />

  </xsd:complexType>

</xsd:element>

<xsd:element name="resheader">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>

    </xsd:sequence>

    <xsd:attribute name="name" type="xsd:string" use="required" />

  </xsd:complexType>

</xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

</xsd:schema>

<resheader name="resmimetype">

  <value>text/microsoft-resx</value>

```

</resheader>

<resheader name="version">

<value>2.0</value>

</resheader>

<resheader name="reader">

<value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

<resheader name="writer">

<value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

</root>