

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA AGRÍCOLA

**DESENVOLVIMENTO DE UM PROGRAMA
COMPUTACIONAL PARA O DIMENSIONAMENTO DE
COLHEDORAS, CONSIDERANDO A PONTUALIDADE
NA COLHEITA**

IACKSON DE OLIVEIRA BORGES

CAMPINAS
FEVEREIRO DE 2004

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA AGRÍCOLA

**DESENVOLVIMENTO DE UM PROGRAMA
COMPUTACIONAL PARA O DIMENSIONAMENTO DE
COLHEDORAS, CONSIDERANDO A PONTUALIDADE
NA COLHEITA**

Tese submetida à banca examinadora para
obtenção do título de Doutor em Engenharia
Agrícola na área de concentração em
Máquinas Agrícolas

IACKSON DE OLIVEIRA BORGES
Orientador: Prof. Dr. Antonio José da Silva Maciel
Co-Orientador: Prof. Dr. Marcos Milan

CAMPINAS
FEVEREIRO DE 2004

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

B644d Borges, Iackson de Oliveira
Desenvolvimento de um programa
computacional para o dimensionamento de
colhedoras, considerando a pontualidade na
colheita / Iackson de Oliveira Borges.—Campinas,
SP: [s.n.], 2004.

Orientadores: Antonio José da Silva Maciel e
Marcos Milan.

Tese (Doutorado) – Universidade Estadual de
Campinas, Faculdade de Engenharia Agrícola.

1. Soja. 2. Perdas durante a colheita. 3.
Máquinas agrícolas. 4. Simulação -
Computadores. I. Maciel, Antonio José da Silva.
II. Milan, Marcos. III. Universidade Estadual de
Campinas. Faculdade de Engenharia Agrícola. IV.
Título.

AGRADECIMENTOS

Aos professores Dr. Antonio José da Silva Maciel, orientador, e Dr. Marcos Milan, co-orientador.

À Protecta – Comércio de Produtos Agropecuários Ltda, de Ponta Grossa - PR, na pessoa dos ilustres Engenheiros Agrônomos João Conrado Schmidt e Fábio Schmidt, pelo apoio técnico e logístico que viabilizou a fase experimental de campo deste projeto.

Ao acadêmico de Engenharia Elétrica Júlio César Nardelli Borges, pela colaboração inestimável e essencial na implementação do programa de simulação.

Ao Professor Dr. Jorge Moretti de Souza, pelas sugestões oportunas na finalização deste trabalho.

À Universidade Federal do Paraná, Setor de Ciências Agrárias, Departamento de Solos e Engenharia Agrícola, pela concessão da oportunidade para o doutoramento.

À Coodenação de Aperfeiçoamento de Pessoal de Ensino Superior – CAPES – pela bolsa de estudo concedida.

Aos professores, funcionários e colegas da área de máquinas da FEAGRI, pela convivência profícua.

À Ana Paula Montagner, secretária da pós-graduação da FEAGRI, pelo tratamento que recebi, sempre atencioso e gentil.

Ao Professor Agenor Maccari, pela amizade e companheirismo nesta jornada.

SUMÁRIO

AGRADECIMENTOS	III
SUMÁRIO.....	IV
LISTA DE FIGURAS.....	VI
LISTA DE TABELAS.....	VIII
LISTA DE ANEXOS	IX
RESUMO.....	XI
ABSTRACT	XIII
1 INTRODUÇÃO	1
2 REVISÃO DE LITERATURA.....	4
2.1 Modelos de seleção associados à pontualidade das operações.....	4
2.2 Coeficientes de oportunidade	13
2.3 Dias úteis de campo	16
2.4 Pontualidade na colheita da soja.....	20
2.5 A soja no Estado do Paraná	23
2.5.1 Produção e produtividade no Estado e na região de Ponta Grossa.....	24
2.5.2 Aspectos gerais do manejo	25
2.5.2.1 Semeadura	25
2.5.2.2 Colheita.....	26
3 MATERIAL E MÉTODOS	28
3.1 Determinação das perdas por atraso na colheita da soja	28
3.1.1 Localização e solo	28
3.1.2 Clima	29
3.1.3 Delineamento experimental.....	29
3.1.4 Implantação e condução do ensaio	33
3.1.5 Colheita e processamento do material	33
3.2 Desenvolvimento do programa computacional	36
3.2.1 Metodologia de desenvolvimento.....	36
3.2.2 Funcionalidade do sistema.....	37
3.2.2.1 Entrada de dados.....	38
3.2.2.2 Processamento	41
3.2.2.3 Saída de dados	45
3.2.2.4 Apresentação das telas.....	46
3.2.2.4.1 Hierarquia	46
3.2.2.4.2 Descrição das telas.....	47

3.2.2.4.2.1 Tela SIMULAÇÃO	47
3.2.2.4.2.2 Tela CENÁRIO-Novo	48
3.2.2.4.2.3 Tela CENÁRIO-Alterar.....	50
3.2.2.4.2.4 Tela PROPRIEDADE – Novo.....	51
3.2.2.4.2.5 Tela PROPRIEDADE-Alterar	51
3.2.2.4.2.6 Tela INDICADORES	53
3.2.2.4.2.7 Tela EXPORTAÇÃO-Excel.....	53
3.2.2.4.2.8 Tela REGIÃO	54
3.2.2.4.2.9 Tela COLHEDORA	55
3.2.2.4.2.10 Tela CULTIVAR – Função de Perdas.....	56
3.3 Validação do modelo computacional	57
3.3.1 Condições contorno dos cenários	58
3.3.1.1 Composições de cultivares	58
3.3.1.2 Colhedoras	58
3.3.1.3 Área de cultivo	59
3.3.1.4 Prazos de conclusão da operação	59
3.3.1.5 Parâmetros gerais.....	60
3.3.1.5.1 Indicadores econômicos	60
3.3.1.5.2 Turno de trabalho.....	61
3.3.1.5.3 Velocidade de operação.....	61
3.3.1.5.4 Precipitação crítica	61
3.3.1.5.5 Período de simulação.....	62
3.3.2 Dimensionamento	62
4 RESULTADOS E DISCUSSÃO	63
4.1 Perdas por atraso na colheita	63
4.1.1 Desenvolvimento da lavoura	63
4.1.2 Datas de colheita e atrasos.....	66
4.1.3 Produtividade comercial e perdas.....	68
4.2 Seleção da colhedora	76
4.2.1 Simulações PROPRIEDADE	76
4.2.2 Simulações CENÁRIO	83
5 CONCLUSÕES.....	92
6 RECOMENDAÇÕES PARA TRABALHOS FUTUROS	93
7 REFERÊNCIAS BIBLIOGRÁFICAS	95
ANEXOS	103

LISTA DE FIGURAS

Figura 1	Temperatura absoluta máxima e mínima médias mensais, e precipitação média mensal no período de 1997 - 2002, para a região de Ponta Grossa (Fonte: Instituto Tecnológico – SIMEPAR, 2003).	30
Figura 2	Representação do croqui da área de ensaio.	31
Figura 3	Representação do bloco experimental.	31
Figura 4	Representação da parcela experimental .	32
Figura 5	Representação da subparcela experimental.	32
Figura 6	Fluxograma da entrada de dados da simulação PROPRIEDADE.	39
Figura 7	Fluxograma da entrada de dados de simulação CENÁRIO.	40
Figura 8	Fluxograma do processamento da simulação PROPRIEDADE.	43
Figura 9	Fluxograma do processamento da simulação CENÁRIO.	44
Figura 10	Fluxograma de armazenamento da simulação.	45
Figura 11	Fluxograma de criação de planilha do Microsoft® Excel.	46
Figura 12	Hierarquia de acesso às telas do programa.	47
Figura 13	Tela SIMULAÇÃO.	48
Figura 14	Tela CENÁRIO-Novo.	49
Figura 15	Tela CENÁRIO-Alterar.	50
Figura 16	Tela PROPRIEDADE-Novo.	52
Figura 17	Tela PROPRIEDADE-Alterar.	52
Figura 18	Tela INDICADORES.	53
Figura 19	Tela EXPORTAÇÃO-Excel.	54
Figura 20	Tela REGIÃO.	55
Figura 21	Tela COLHEDORA.	56
Figura 22	Tela CULTIVAR-Função de Perdas.	57
Figura 23	Atributos climáticos durante o período de desenvolvimento vegetativo dos cultivares no período de ensaio (Fonte: Instituto Agrônômico do Paraná – IAPAR; Instituto Tecnológico – SIMEPAR, 2003).	65
Figura 24	Atributos climáticos durante o período de colheita no ensaio (Fonte: Instituto Agrônômico do Paraná – IAPAR; Instituto Tecnológico – SIMEPAR, 2003).	67
Figura 25	Tendência de produtividade de soja comercial do cultivar FT Cometa.	68
Figura 26	Tendência de produtividade de soja comercial do cultivar M - Soy 5942.	69
Figura 27	Tendência de produtividade de soja comercial do cultivar FT Abyara.	69
Figura 28	Tendência de produtividade de soja comercial do cultivar BRS 133.	70
Figura 29	Produtividade média de soja comercial do cultivar FT Cometa.	73
Figura 30	Produtividade média de soja comercial do cultivar M – Soy 5942.	73
Figura 31	Produtividade média de soja comercial do cultivar FT Abyara.	74
Figura 32	Produtividade média de soja comercial do cultivar BRS 133.	74
Figura 33	Comparativo de produtividade de soja comercial dos cultivares e perdas relativas respectivas.	75
Figura 34	Renda líquida média anual da frota e dos demais modelos de colhedoras para a composição I (FT Cometa + M – Soy 5942).	78

Figura 35	Renda líquida média anual da frota e dos demais modelos de colhedoras para a composição II (FT Abyara + BRS 133).....	78
Figura 36	Renda líquida média anual da frota e dos demais modelos de colhedoras para a composição III (Cometa + FT Abyara)	79
Figura 37	Renda líquida média anual da frota e dos demais modelos de colhedoras para a composição IV (FT Cometa).	80
Figura 38	Renda líquida média anual da frota e dos demais modelos modelos de colhedoras para a Composição V (FT Abyara).....	81

LISTA DE TABELAS

Tabela 1	Evolução da cultura da soja no Estado do Paraná, na região de Ponta Grossa e Brasil - Safra 94/95 a 2001/02 (Fonte: IBGE 2002; SEAB, 2002).....	24
Tabela 2	Características texturais e químicas do solo da área experimental (0 – 20 cm).	29
Tabela 3	Eventos da implantação e condução da soja na área experimental.	34
Tabela 4	Regulagens principais da colhedora utilizada (Fonte: New Holland Latino Americana Ltda, 1998).....	35
Tabela 5	Composição de cultivares utilizadas nas simulações.....	59
Tabela 6	Indicadores econômicos utilizados	60
Tabela 7	Data de ocorrência dos principais estádios fenológicos dos cultivares do ensaio.	64
Tabela 8	Datas e atrasos na colheita das subparcelas.	66
Tabela 9	Comparativo das médias de produtividade de soja comercial dos tratamentos.....	71
Tabela 10	Comparativo das médias de produtividade de soja comercial nos subtratamentos.	71
Tabela 11	Comparativo das médias de produtividade de soja comercial dos subtratamentos para cada cultivar.....	72
Tabela 12	Coeficientes das funções de perdas por atraso dos cultivares testados.	76
Tabela 13	Limites de área para uso da frota e demais modelos de colhedoras segundo as composições de cultivares simuladas, em hectare.....	81
Tabela 14	Média de dias úteis de campo para colheita observados na série histórica de 1954 a 2003, região de Ponta Grossa (Fonte: Autor).....	82
Tabela 15	Demanda de colhedoras de 125,1 kW e renda líquida média anual para diferentes combinações de cultivares, áreas e prazos de colheita.	85
Tabela 16	Demanda de colhedoras de 132,5 kW e renda líquida média anual para diferentes combinações de cultivares, áreas e prazos de colheita.	87
Tabela 17	Demanda de colhedoras de 165,6 kW e renda líquida média anual para diferentes combinações de cultivares, áreas e prazos de colheita.	89

LISTA DE ANEXOS

A - Tabela 1 Produtividade de soja comercial (kg.ha^{-1}) obtido no delineamento experimental.	103
A - Tabela 2 Análise de variância do ensaio experimental considerando 4 cultivares e 10 épocas ou atrasos ou atrasos de colheita.	105
A - Tabela 3 Análise de variância dos tratamentos, considerando 10 épocas ou atrasos de colheita.	105
B - Tabela 1 Simulações PROPRIEDADE – Frota de colhedoras.	107
B - Tabela 2 Simulações PROPRIEDADE – Colhedora de 121,5 kW.	109
B - Tabela 3 Simulações PROPRIEDADE – Colhedora de 132,5 kW.	111
B - Tabela 4 Simulações PROPRIEDADE – Colhedora de 165,6 kW.	113
B - Tabela 5 Simulações PROPRIEDADE – Frota de colhedoras (RESUMO).	115
B - Tabela 6 Simulações PROPRIEDADE – Colhedora de 125,1 kW (RESUMO).	115
B - Tabela 7 Simulações PROPRIEDADE – Colhedora de 132,5 kW (RESUMO).	116
B - Tabela 8 Simulações PROPRIEDADE – Colhedora de 165,6 kW (RESUMO).	117
C - Tabela 1 Simulações CENÁRIO – Colhedora de 125,1 kW.	118
C - Tabela 2 Simulação CENÁRIO – Colhedora de 132,5 kW.	121
C - Tabela 3 Simulações CENÁRIO – Colhedora de 165,6 kW.	124
D - Requisitos para instalar e executar o programa.	127
E - Rotina para visualização dos códigos fonte do programa.	127
F - Módulo 1: Módulo Principal (Arquivo PPrograma.dpr).	128
F - Módulo 2: Módulo de Apresentação (Arquivo USplash.pas).	129
F - Módulo 3: Módulo de Simulação (Arquivo USimulação.pas).	130
F - Módulo 4: Módulo Cadastro de Máquinas (Arquivos UMaquina.pas, UMaquinaNovo.pas, UMaquinaAlterar.pas, UMaquinaExcluir.pas).	139
F - Módulo 5: Módulo Cadastro de Regiões (Arquivos URegiao.pas, URegiaoNovo.pas, URegiaoAlterar.pas, URegiaoExcluir.pas).	150
F - Módulo 6: Módulo Cadastro de Cultivares (Arquivo UCultivar.pas).	162
F - Módulo 7: Módulo Visualização de Imagens (Arquivo UVisualizarImg.pas).	167
F - Módulo 8: Módulo Geração de Cenários (Arquivos UCenarioNovo.pas, UCenarioAlterar.pas).	168
F - Módulo 9: Módulo Geração de Propriedades (Arquivos UPropriedadeNovo.pas, UPropriedadeAlterar.pas).	234
F - Módulo 10: Módulo Cofiguração dos Indicadores (Arquivo UIndicador.pas).	294
F - Módulo 11: Módulo de Entrada de Velocidade e Número de Máquinas (Arquivo UCaixaDeEntrada.pas).	296
F - Módulo 12: Módulo Exportação Para Planilha do MS Excel® (Arquivo UExpExcel.pas).	297

G - Código Fonte 1: PPrograma.dpr <<código>>	128
G - Código Fonte 2: USplash.pas <<código>>	129
G - Código Fonte 3: USimulacao.pas <<código>>	130
G - Código Fonte 4: UMaquina.pas <<código>>	139
G - Código Fonte 5: UMaquinaNovo.pas <<código>>	143
G - Código Fonte 6: UMaquinaAlterar.pas <<código>>	145
G - Código Fonte 7 : UMaquinaExcluir.pas <<código>>	147
G - Código Fonte 8: URegiao.pas <<código>>	150
G - Código Fonte 9: URegiaoNovo.pas <<código>>	155
G - Código Fonte 10: URegiaoAlterar.pas <<código>>	157
G - Código Fonte 11: URegiaoExcluir.pas <<código>>	160
G - Código Fonte 12: UCultivar.pas <<código>>	162
G - Código Fonte 13: UVisualizarImg.pas <<código>>	168
G - Código Fonte 14: UCenarioNovo.pas <<código>>	168
G - Código Fonte 15: UCenarioAlterar.pas <<código>>	202
G - Código Fonte 16: UPropriedadeNovo.pas <<código>>	234
G - Código Fonte 17: UPropriedadeAlterar.pas <<código>>	262
G - Código Fonte 18: UIndicador.pas <<código>>	294
G - Código Fonte 19: UCaixaDeEntrada.pas <<código>>	296
G - Código Fonte 20: UExpExcel.pas <<código>>	297

RESUMO

A colheita de grãos é uma operação crítica devido ao risco climático que pode causar atrasos com perdas consideráveis na quantidade e qualidade do material colhido. Embora tais perdas sejam desconhecidas para a maioria das culturas, os produtores procuram evitá-las empregando frotas de colhedoras geralmente com reserva de capacidade. Quando em excesso, resulta no aumento dos custos fixos da operação devido ao capital extra imobilizado em máquinas. A falta de capacidade, por outro lado, aumenta o custo das perdas devidas ao atraso na conclusão da operação no prazo ótimo. Em ambos casos, há redução da renda líquida da operação, o que se denomina de custo de pontualidade. O problema da seleção de colhedoras consiste então, em estabelecer um equilíbrio entre o custo extra do capital investido na capacidade da frota como um seguro para garantir a pontualidade da operação e o custo das reduções de rendimento da safra por falta desta, ou ainda da redução de tempo para conclusão da operação de colheita para dar lugar à operação seguinte. Devido ao número e a complexidade das variáveis envolvidas no problema, o objetivo principal deste trabalho foi desenvolver um programa computacional para o dimensionamento de colhedoras considerando custo da pontualidade da operação. O programa foi validado na colheita de soja (*Glycine max* (L.) Merrill) na região de Ponta Grossa, Estado do Paraná. Para atingir o objetivo principal, o trabalho foi desenvolvido em três etapas. A primeira consistiu da determinação experimental das funções de perdas por atraso na colheita de 4 cultivares de soja nas condições edafoclimáticas específicas. A segunda consistiu do desenvolvimento de um programa computacional utilizando Borland® Delphi 5.0 para simulação de cenários de colheita, capaz de interagir as funções de perdas com as características da máquina e o risco climático, para estimar o custo das perdas de oportunidade, o custo do serviço mecanizado e a renda líquida da operação como indicador da pontualidade. Finalmente, a terceira etapa consistiu da validação do programa simulando cenários de colheita numa propriedade referência na região cultivada com 500 ha de soja, para determinar a capacidade ótima da respectiva frota colhedoras, composta por 2 unidades de 125,1 kW e 1 unidade de 165,6 kW. Foram consideradas 5 combinações de cultivares (FT Cometa + M – Soy 5942, FT Abyara + BRS 133, FT Cometa + FT Abyara, FT Cometa e FT Abyara). Na determinação das funções

de perdas por retardamento na colheita de soja, os cultivares FT Cometa e M – Soy 5942 revelaram menores tendências de perdas, enquanto que os cultivares BRS 133 e FT Abyara maiores tendências, quando comparadas entre si. Na validação do modelo para a colheita de soja, as simulações demonstraram que, dependendo da composição de cultivares e do prazo para conclusão da operação, a capacidade ótima das colhedoras é alterada. A simulação de cenários na propriedade referência revelou também que, para qualquer composição de cultivares, a frota de colhedoras respectiva operaria na colheita de soja com capacidade ociosa. Admitida a composição de cultivares FT Cometa e M – Soy 5942, apenas 1 unidade de colhedora de 121,5 kW com 4,8 m de plataforma atenderia a demanda da área, proporcionando um ganho médio de renda líquida estimado em US\$ 47.255,00. ano⁻¹.

Palavras-chave: 1. Soja. 2. Perdas durante a colheita. 3. Máquinas agrícolas. 4. Simulação - Computadores.

ABSTRACT

Grain harvest is a critical operation due to climatical risk that may cause delays with significant losses in quantity and quality of the harvested material. Although such losses are unknown for most of crops, producers try to avoid them employing fleet of harvesters with more capacity than the necessary increasing the fixed costs of the operation due to the extra capital invested on the machines. On the other hand, the lack of capacity increases the cost of losses due to the delay on the operation conclusion in an ideal appointed day. There is a reduction of net profit of the operation, in both cases, which is called of indirect costs. The problem of selecting harvesters consist in establishing a balance between the extra cost of the invested capital in the fleet capacity and the reductions of the costs of the crop productivity or a time reduction for the conclusion of the harvest operation, so the following operation can take place. Due to the number and the complexity of the involved variables in this matter, the main objective of this study was to develop a computer program for harvesters measuring, considering the indirect cost of the operation. The program was validated for soybean harvest in Ponta Grossa region, state of Parana. In order to achieve the main goal, the study was carried out in three steps. The first one consisted of the experimental determination of the functions of losses by the delay of four soybean cultivars harvest, in specific conditions. The second one consisted in developing a computer program using Borland®-Delphi 5.0 to simulate harvest sceneries capable of interacting the functions of losses with the machine features and the climatical risks, in order to evaluate the indirect costs, the cost of mechanized service and the net profit of the operation, as an indicator of pontuality. The third one consisted of the program validation simulating harvest sceneries in a property, known as reference in the region, with 500 ha of soybean to determine the harvesters fleet ideal capacity with 2 units of 125 kW and 1 unit of 166 kW. Five combinations of cultivars (FT Cometa + M-Soy 5942, FT Abyara + BR 133, FT Cometa + Ft Abyara, FT Cometa and FT Abyara) were considered. For the determination of the functions of losses by the delay in soybean harvest, the cultivars FT Cometa and M-Soy 5942 showed less tendencies for losses, while cultivars BRS 133 and FT Abyara showed more tendencies, when compared to each other. In

the model validation for soybean harvest the simulations showed that depending on the cultivar composition and the appointed day for the conclusion of the operation, the harvesters ideal capacity is altered. The sceneries simulation, at the referred property showed that, whatever the cultivar composition is, the particular harvesters fleet works with idle capacity. Once established the composition of cultivars FT Cometa and M-Soy 5942, only 1 unit of the harvester 121 kW with 4.8 m platform would attend the demanding of the area, giving an average net profit evaluated in US\$ 47,255.00 per annum.

Keywords: 1. Soybean. 2. Harvest losses. 3. Agricultural machinery. 4. Computer simulation.

1 INTRODUÇÃO

Uma operação agrícola, para ser efetiva no sistema de produção, deve ser executada no prazo agronomicamente ótimo, e a isso se denomina pontualidade da operação. No planejamento tradicional, o dimensionamento de máquinas é baseado no objetivo de completar a operação no menor prazo possível para evitar perdas de oportunidade sobretudo devidas ao risco climático e quebra de equipamentos. Todavia, isso pode exigir uma quantidade de máquinas que resulta na elevação dos custos fixos da operação. Por outro lado, uma quantidade pequena pode ser incapaz de executar a operação no prazo ótimo e resultar no aumento das perdas por atraso. Em ambos casos há redução da renda líquida da operação e a isso se denomina custo de pontualidade.

Assim, o problema do dimensionamento de máquinas consiste em estabelecer um equilíbrio entre o custo extra de capital investido na capacidade da maquinaria como um seguro para garantir a pontualidade e o custo das reduções de rendimento da safra por falta desta, ou ainda da redução de tempo para conclusão das operações críticas como semear, cultivar e colher para dar lugar às operações da safra seguinte.

O conceito de oportunidade de uma operação agrícola não é novo. Os modelos de seleção de máquinas mais recentes são processos de simulação, geralmente interativos, que incorporam o custo da pontualidade das operações agrícolas. A simulação da operação associada ao respectivo custo de oportunidade pode ser um instrumento eficiente para auxiliar na tomada de decisão permitindo que as variáveis envolvidas sejam estudadas e avaliadas quanto à sua importância na composição dos custos .

Uma das operações agrícolas mais críticas é a colheita de grãos principalmente devido à incerteza climática. A partir do ponto de maturação recomendado, atrasos significam perdas em quantidade e qualidade do material colhido. Embora os produtores constatem empiricamente a existência dessas perdas, não possuem estimativas precisas da sua evolução. Em decorrência, utilizam frotas de colhedoras com grande capacidade para concluir a operação no menor prazo possível. Todavia, é provável que esta pontualidade tenha um alto custo devido ao excesso de máquinas.

Este trabalho tratou do dimensionamento de colhedoras considerando a pontualidade na colheita de soja (*Glycine max* (L.) Merrill) em face da sua importância econômica para o Brasil

e o Estado do Paraná, em particular a região de Ponta Grossa. Na safra 2001/02, o Brasil produziu 41,9 milhões de toneladas desta leguminosa e o Estado do Paraná 9,4 milhões de toneladas, ou seja, 22,4% da produção nacional. Na agricultura paranaense, a soja ocupa o maior destaque entre as culturas de grãos não só pela extensão de área cultivada, mas também pelo faturamento, que foi de US\$ 1,65 bilhões naquela safra (IBGE, 2002). A região de Ponta Grossa cultivou 309.000 hectares produzindo 930.000 toneladas, equivalente a 9,9% da produção estadual. Esta região se caracteriza por uma comunidade de produtores que usam tecnologias inovadoras associadas ao sistema de produção em plantio direto com intensa mecanização, auferindo produtividade média de soja de 3.000 kg.ha⁻¹ (SEAB, 2002).

Para a safra normal no Paraná, a época de semeadura indicada para a maioria das cultivares se estende da segunda quinzena de outubro à segunda quinzena de dezembro (EMBRAPA, 2002). Na região de Ponta Grossa, a semeadura é preferencialmente executada entre meados dos meses de outubro e novembro, que apresenta os melhores resultados de produtividade e menor porte das plantas.

A colheita pode ser realizada entre os meses de março e maio do ano subsequente, dependendo das características do cultivar e da influência das fatores climáticos. Recomenda-se que ela deva começar tão logo a soja atinja o estágio fenológico R8, observando-se umidade do grão entre 13% e 15% (EMBRAPA, 2002). A partir deste ponto de maturação, a tendência é a deterioração dos grãos e a debulha das vagens em intensidade proporcional ao tempo que a soja permanecer no campo, chegando finalmente à perda total transcorrido algum tempo. Todavia, os produtores locais também observam empiricamente que entre as variedades de soja cultivadas na região há diferenças quanto à tolerância ao atraso. Esta característica pode alterar a disponibilidade de tempo para a operação e a demanda de capacidade e, em consequência, influenciar os custos da pontualidade alterando a renda líquida da colheita mecanizada de soja.

O objetivo principal deste trabalho foi desenvolver um programa computacional para o dimensionamento de colhedoras considerando o custo da pontualidade na operação. Para atingir este objetivo, o trabalho foi desenvolvido em três etapas. A primeira etapa consistiu da determinação experimental das funções de perdas por atraso na colheita de quatro cultivares de soja com ciclos de maturação precoce, semiprecoce e médio, nas condições edafoclimáticas da região de Ponta Grossa. A segunda etapa foi o desenvolvimento do programa computacional

utilizando Borland® Delphi 5.0, capaz de simular cenários de colheita interagindo as funções de perdas dos cultivares de soja, as características dimensionais das colhedoras e o risco climático para calcular o custo das perdas de oportunidade, o custo do serviço mecanizado e a receita líquida da operação, como indicador da pontualidade. Finalmente, a terceira etapa consistiu da validação do programa simulando cenários de colheita numa propriedade referência situada na região de Ponta Grossa, para dimensionar a capacidade requerida e identificar possíveis ociosidades da frota de colhedoras daquela propriedade.

2 REVISÃO DE LITERATURA

2.1 Modelos de seleção associados à pontualidade das operações

Um sistema de produção agrícola envolve um processo contínuo composto de uma série de operações que inicia na preparação do solo, seguindo pela semeadura, cultivo e colheita, cuja execução na época ótima depende do tempo e da capacidade da maquinaria disponível. A época ótima de uma operação é o período de tempo durante o qual ela deve ser realizada para se tornar efetiva no sistema de produção. Embora decisões sobre a antecipação ou atraso das operações agrícolas podem ser tomadas, uma vez iniciada uma operação, o seu desenvolvimento segue num ritmo que depende da capacidade da máquina, fatores climáticos e agronômicos relacionados com a produção (LAL et al., 1991).

A indagação a respeito do tamanho de máquina adequado para executar uma operação agrícola particular é de difícil resposta. Sucede que os custos da maquinaria não dependem de uma só máquina ou operação, mas de um conjunto de máquinas composto por tratores e unidades autopropelidas que executam várias operações, na maioria das vezes em sistemas de produção policulturais. Ademais, o desempenho de uma máquina pode ser afetado pelo desempenho de outra. Alguns custos da maquinaria embora reconhecidos, não podem ser facilmente calculados. É o caso, por exemplo, do custo da pontualidade resultante do dimensionamento inadequado do equipamento para a execução de uma operação num prazo agronomicamente ótimo. Estimativas de rendimento das lavouras relacionadas aos custos da maquinaria são ainda dificultadas pelas condições agronômicas e climáticas, sobretudo, que introduzem a incerteza na administração do sistema mecanizado.

No passado, os modelos de seleção da máquinas eram baseados no objetivo de completar as operações num prazo arbitrariamente escolhido objetivando um custo total mínimo (HUGHES e HOLTMAN, 1976; OSKAN e FRISBY, 1981). Mais recentemente, os pesquisadores introduziram nos modelos de seleção as perdas de oportunidade que ocorrem quando as operações não são concluídas nos prazos ótimos, somando o custo dessas perdas aos custos diretos do serviço mecanizado. Porém, o prazo de conclusão das operações é afetado pelo tempo útil disponível, número de operações e área de cultivo, que são diferentes

para cada produtor. Ademais, a disponibilidade de máquinas de diversos tamanhos e capacidades tornou a seleção um problema relativamente complexo. Máquinas maiores e em maior quantidade aumentam a pontualidade reduzindo os custos de oportunidade na medida em que permitem a realização das operações no prazo ótimo. Em excesso, porém, aumentam os custos fixos pela quantidade de equipamentos alocados na execução da operação.

Para WHITSON et al. (1981), o problema da seleção é, portanto, buscar um equilíbrio entre o custo extra do capital investido na capacidade das máquinas e o custo das reduções de rendimento da lavoura ou da redução do tempo exigido para concluir operações críticas como semear, cultivar e colher para dar lugar às operações da safra seguinte.

CHEN e McCLENDON (1985), TSAI et al. (1987) e MILES e TSAI (1987) desenvolveram modelos de seleção de máquinas que incorporaram os efeitos das condições agronômicas e climáticas para simular sistemas mecanizados envolvendo outras operações além da semeadura e colheita. Desenvolveram também modelos de seleção de máquinas para sistemas de produção policulturais onde empreendimentos diferentes competiam pelos mesmos recursos da propriedade. Estes modelos testavam o comportamento dinâmico de um sistema de produção, porém, a sua capacidade de simular situações reais dependia da formulação e incorporação de fatores que influenciavam o sistema.

BURROWS e SIEMENS (1974) desenvolveram um programa computacional para determinar o custo mínimo, o número e o tamanho das máquinas para fazendas de soja e milho com áreas variando entre 120 a 840 ha no Cinturão do Milho, EUA. O programa foi projetado também para ser utilizado como uma ferramenta de apoio à decisão para os agricultores na aquisição de máquinas. Os dados de entrada principais foram as operações a serem executadas numa ordem de programação com as suas respectivas datas de início e término, área e jornada de trabalho e custo da mão-de-obra. A saída de dados incluía os custos fixos e variáveis da maquinaria, mão-de-obra e custos de oportunidade. Os autores chamaram a atenção para a inexistência de informações relativas às perdas de oportunidade na colheita da soja e do milho, e a complexidade de obtê-los dada interação de fatores como variedade, perdas durante a operação e custos de secagem. Concluíram, finalmente, que a habilidade de um programa para dimensionar o tamanho ótimo das máquinas depende do grau de precisão com que a capacidade do equipamento pode ser estimada nas condições em que é utilizada na propriedade. As simulações mostraram que se a capacidade assumida fosse dobrada, poderiam

ser utilizadas máquinas menores e os custos variáveis diminuiriam. Por outro lado, se capacidade fosse menor que a assumida, além do aumento do custo variável, um custo de oportunidade maior também seria contabilizado.

PHILLIPS e O'CALLAGHAN (1974) desenvolveram um modelo computacional para analisar algumas variáveis que influenciam o custo da colheita de cereais no Reino Unido. Os resultados mostraram que os custos associados à operação foram minimizados quando as máquinas operaram numa área de trabalho denominada “economicamente ótima”. Porém, os fatores que restringiram o tempo disponível para a operação tenderam a elevar os níveis de perdas das máquinas acima do economicamente aceitável. Os autores concluíram que a duração do tempo de colheita poderia ser reduzida se a máquina operasse com níveis de perdas mais elevados, porém isso afetaria o tempo disponível para as etapas de produção subsequentes aumentando os custos totais de oportunidade. Os autores concluíram também que o tempo disponível para operação variou com o limite de umidade do produto. Se o limite fosse reduzido, seriam necessárias colhedoras com maior capacidade para evitar perdas de oportunidade excessivas, o que elevaria os custos totais. Por outro lado, um aumento do limite de umidade implicaria em custos adicionais com secagem. Os autores observaram, entretanto, que a maioria dos produtores prefere aguardar a redução da umidade do produto no campo antes de iniciar a colheita porque a redução da qualidade do grão devido à umidade elevada não compensa os ganhos obtidos com a antecipação da colheita. Os autores concluíram, finalmente, que as perdas por oportunidade na colheita de cereais tiveram o maior peso no custo total da operação. Em decorrência da magnitude das mesmas, grandes colhedoras poderiam ser economicamente justificadas para áreas de trabalho relativamente pequenas quando o tempo útil disponível é reduzido.

EDWARDS e BOEHLJE (1980) desenvolveram um modelo de simulação para calcular o custo da pontualidade de dez conjuntos de máquinas considerando as funções de perdas por oportunidade para lavouras de milho e soja variando de 81 a 364 ha no estado de Iowa (EUA). O modelo foi usado para simular o comportamento dos sistemas em cada ano da série temporal de dias úteis verificados em nove distritos agrícolas daquele estado entre 1958 e 1977. Comparados com amostras de 180 propriedades, os resultados das simulações não apresentaram diferença significativa com valores observados. Todavia, o modelo não

considerou nas simulações as variações de produtividade das lavouras devidas às condições climáticas e agronômicas.

PARSONS et al. (1981) desenvolveram um modelo de programação linear para analisar alguns aspectos econômicos da relação entre pontualidade e confiabilidade considerando investimento numa máquina maior com boa assistência técnica para diminuir o tempo consumido em reparos ou duas máquinas menores para diluir os riscos de pontualidade. Para testar esta relação, foram realizadas simulações na época de semeadura e colheita em áreas de 202 ha e 404 ha de soja e milho com sistema de produção convencional. Os autores demonstraram que em ambas áreas estudadas, a máquina maior funcionou como um seguro para garantir a pontualidade reduzindo as perdas de tempo consumidas em reparos. Por outro lado, o uso de duas semeadoras menores ao invés de uma única unidade equivalente, mostrou-se mais confiável e rentável na área maior do que na área menor, quando a disponibilidade de tempo foi crítica.

WHITSON et al. (1981) desenvolveram um modelo de programação linear para selecionar combinações ótimas de culturas e máquinas considerando o risco climático em grandes áreas do Texas Coastal Prairie, EUA. Os resultados mostraram que sob condições de maior risco climático, foi interessante os agricultores trocarem a cultura do algodão pelo sorgo granífero, cujas operações do sistema de produção demandavam níveis de confiança menores para serem concluídas nas épocas ótimas. Concluíram também que os investimentos em máquinas por unidade de área aumentaram com o aumento do nível de confiança requerido. Os resultados também indicaram que o risco climático foi uma variável importante que deveria ser incluída nos modelos de decisão, nas estratégias de produção e na seleção de máquinas para a maximização de rentabilidade.

HUAN-WEN e HUNT (1985) modificaram a equação para seleção de máquinas baseada na largura efetiva de operação e na capacidade campo proposta por HUNT (1964) e propuseram um novo modelo de seleção baseado na demanda de potência e no custo da pontualidade da operação. A validação do modelo mostrou que a área de trabalho foi a variável que mais influenciou o nível de capacidade de campo, seguido pela energia requerida na operação. Ainda, segundo os autores, foi igualmente importante a manutenção de um fator de carga, jornada de trabalho, disponibilidade de tempo útil, a operacionalidade do equipamento e eficiência de campo elevadas para minimizar o custo da capacidade

operacional da máquina. A diferencial de primeira ordem da equação geral do modelo forneceu a potência ótima para uma capacidade de campo demandada. Como a sua transformação em unidades autopropelidas conduziu à frações, os autores recomendaram o arredondamento para o inteiro imediatamente acima, porém, buscando sempre o mesmo modelo visto que a uniformidade da frota melhora o gerenciamento dos serviços de manutenção. A comparação dos resultados das simulações com as observações de campo de dois anos numa área de lavoura de milho no estado de Illinois (EUA), mostrou que o modelo foi válido disso resultado a recomendação de uma colhedora a mais para reduzir as perdas de oportunidade face às incertezas climáticas.

OZKAN e EDWARDS (1986) desenvolveram um modelo computacional para investigar a influência de mudanças nos conjuntos de máquinas e parâmetros de produção sobre os custos da pontualidade e na renda líquida nas lavouras de milho e soja. Os custos de cada operação especificada no sistema de produção foram estimadas usando técnicas de fluxo de caixa, e os respectivos prazos de conclusão foram estimados com base na jornada de trabalho, expectativa de dias úteis de campo, capacidade das máquinas e área de trabalho. Os custos de pontualidade foram estimados em função das datas de semeadura e colheita. O modelo foi testado numa propriedade de 243 ha de milho e soja em Iowa (EUA), com cinco cenários de pontualidade variando a jornada de trabalho, o número de operações de preparo do solo e tamanhos de máquinas. Os resultados das simulações mostraram que a combinação do uso de implementos mais largos para incorporação de herbicida numa única passada com o aumento de uma hora na jornada de trabalho permitiu uma redução do custo de oportunidade em 64,9% e um aumento de 23,6% na receita líquida. O modelo foi desenvolvido para administradores rurais, permitindo variar parâmetros econômicos e de produção. Entre os parâmetros econômicos, destacam-se a taxa de juros, impostos, preço da mão-de-obra e preço dos produtos. Entre os parâmetros de produção, destacam-se a localização geográfica da propriedade, a área de cultivo e a expectativa de redução da produtividade em função das datas de semeadura e colheita. Interagindo esses parâmetros, o usuário tem flexibilidade para simular cenários máquina/lavoura e buscar o melhor resultado econômico do balanço entre os custos de máquinas e o custo de pontualidade.

McCLENDON et al. (1987) desenvolveram um modelo de simulação para administração de sistemas mecanizados para produção de soja e trigo. Os autores assumiram

que a data de maturidade e o rendimento da lavoura dependiam apenas do cultivar e da data de semeadura, não considerando a influência dos elementos climáticos. Assim, não fizeram ajustes no rendimento das lavouras devido a antecipação ou atraso na colheita. Para calcular atrasos na programação das operações, usaram uma regra simples de decisão baseado nos eventos de chuva.

Para OSKAN et al. (1990), a seleção correta do equipamento requer a determinação cuidadosa dos custos da maquinaria que incluem os custos de propriedade, os custos operacionais e os custos de pontualidade associados. Para atingir este objetivo, desenvolveram um modelo de simulação denominado TIMECOST para calcular a capacidade e os custos da maquinaria. O TIMECOST foi associado a outro modelo denominado DRAINMOD, desenvolvido por SKAGGS (1983) segundo OSKAN et al. (1990), capaz de estimar os dias úteis para operações de campo a partir de dados climatológicos, propriedades do solo, parâmetros do sistema de drenagem e trafegabilidade e parâmetros de produção para estimar a produtividade da lavoura. O TIMECOST foi testado numa área de 202 ha de solo drenado no Cinturão do Milho, EUA. Apresentou restrições, todavia, porque o custo de oportunidade associado à colheita não pôde ser determinado devido à falta de dados de redução de rendimento com o atraso na operação. Ademais, como o DRAINMOD foi desenvolvido para avaliar a produtividade da cultura do milho em solos drenados, o TIMECOST não pôde ser utilizado para outras culturas e em outros tipos de solos. Apesar dessas restrições, os autores consideraram que estimativa do custo de pontualidade a partir da drenagem do solo através do TIMECOST foi uma metodologia inovadora, e permitiu que análises de sensibilidade testassem a interação da intensidade de drenagem com os sistemas mecanizados sobre o custo de pontualidade das operações para determinar a combinação economicamente mais atraente.

SIEMENS et al. (1990) desenvolveram um programa para seleção e administração de máquinas agrícolas onde os dados de entrada do programa consistiam de uma lista de operações a serem executadas com as respectivas datas de início, área e jornada de trabalho. Outras informações de entrada incluíam também parâmetros econômicos, produtividade da cultura e coeficientes de oportunidade para semeadura e colheita, associados a um banco de dados contendo uma lista de preços e produtividade das máquinas, probabilidade de dias úteis e constantes de equações para estimativa do custo mecanizado. O programa foi capaz de

simular as operações agrícolas com diferentes conjuntos ou um conjunto específico de máquinas, e produzir um relatório com os custos de máquinas e os custos de pontualidade respectivos.

LAL et. al (1991) desenvolveram um simulador para operações mecanizadas em sistemas de produção policulturais capaz de calcular o trabalho diário com base na chuva diária e nos recursos de maquinaria e mão-de-obra disponível. O modelo podia ser ajustado a uma variedade de cenários agrícolas, de propriedade altamente mecanizada à propriedade baseada na utilização intensiva da mão-de-obra com pouca ou nenhuma mecanização. O modelo foi capaz de simular operações em sistemas de produção indicando aquelas que não puderam ser completadas no prazo ótimo devido à escassez de mão-de-obra ou falta capacidade do equipamento. Contudo, o simulador não foi capaz de estimar o custo de oportunidade das operações, além de ter-se revelado limitado para propriedades cujos recursos são compartilhados na produção animal.

LAL et al. (1992) desenvolveram também programa integrado de suporte à decisão para administração de máquinas em sistemas de produção policulturais, denominado FARMSYS. Utilizando dados de clima, máquinas, mão-de-obra e informações gerenciais da propriedade, o sistema foi capaz de avaliar o comportamento operacional de um sistema de produção, estimando a produção, a renda bruta e a rentabilidade tanto para talhões individuais como para a propriedade inteira. O modelo consistiu de quatro componentes operadores: gerenciador de informações, simulador de operações, calculador de rendimento e um analista de sistemas. O gerenciador de informações era uma interface que permitia ao usuário atualizar as informações da propriedade desenvolvido anteriormente por LAL et al. (1990). O simulador de operações implementava as operações de campo utilizando informações de máquinas, mão-de-obra e chuva diária também desenvolvido por LAL et al. 1991. O calculador de rendimento simulava o desenvolvimento da lavoura e estimava a produção. Finalmente, o analista de sistema fazia recomendações de ajustes baseadas nos relatórios de simulação garantindo que as operações fossem completadas no prazo ótimo. O programa assumia que todos os recursos da propriedade estavam disponíveis para a produção agrícola, o que restringiu o seu uso em propriedades que desenvolvem atividades pecuárias, sobretudo em escala de subsistência.

PARMAR et al. (1994) desenvolveram um modelo de simulação para administrar sistemas mecanizados na cultura do amendoim, denominado PNUTMM, capaz de simular o

desenvolvimento da lavoura, os custos mecanizados e a renda líquida para diferentes conjuntos de máquinas e programas de operações, condições agronômicas e climáticas. Este modelo incorporou um outro simulador denominado PNUTGRO desenvolvido por BOOTE et al. (1989), segundo os autores. O modelo foi testado por HOOGENBOOM et al. (1992) numa variada gama de solos, clima e condições de manejo da lavoura. A partir de parâmetros agronômicos e climáticos, o PNUTGRO foi capaz de estimar a produtividade na data de maturidade fisiológica permitindo assim que o PNUTMM calculasse os efeitos da pontualidade na operação de colheita do amendoim sobre receita líquida. O programa foi utilizado para simular um sistema de produção composto por seis operações com dois conjuntos de máquinas e duas estratégias de irrigação, utilizando uma série histórica de dados climáticas entre 1979 e 1985. O resultado das simulações mostrou que a data de colheita prevista pelo PNUTMM nem sempre coincidiu com a maturidade fisiológica prevista pelo PNUTGRO, resultando diferenças significativas de produção estimadas por ambos programas. A restrição foi solucionada incorporando ao PNUTMM dados experimentais das perdas por atraso na colheita do amendoim para corrigir a produtividade estimada pelo PNUTGRO.

PARMAR et al. (1996) utilizaram simulação computacional com algoritmos genéticos para otimizar a seleção de máquinas para cultura do amendoim e maximizar a renda líquida das operações mecanizadas. O conjunto ótimo de máquinas foi determinado com base em duas estratégias de busca, uma exaustiva e outra de inteligência artificial. A estratégia de busca exaustiva consistiu em rodar o modelo de simulação PNUTMM com todos os conjuntos de máquinas possíveis para então selecionar aquele que produzisse o maior retorno líquido. Por outro lado, algoritmos genéticos foram usados na estratégia de busca inteligente para gerar conjuntos de máquinas para serem avaliados no PNUTMM. Nas simulações, um algoritmo genético foi capaz de determinar o conjunto próximo do ótimo em 10% do tempo requerido na busca exaustiva. Ajustes no algoritmo não só permitiram reduzir à metade o tempo de busca como também melhoraram a qualidade dos resultados encontrados. Os autores salientaram que a otimização das máquinas através de modelos tradicionais de seleção é um processo tedioso que consome muito tempo porque os cenários são simulados variando o valor de um parâmetro de cada vez para avaliar os efeitos do mesmo sobre resultados. Os autores advertiram ainda que este método não produz resultados satisfatórios porque no gerenciamento de máquinas interagem simultaneamente um grande número de variáveis.

No Brasil, VEIGA (2000) desenvolveu um modelo empírico para seleção de máquinas agrícolas na cultura da soja baseado na pontualidade da operação de semeadura. Para determinar a pontualidade da operação, foram obtidas experimentalmente as funções de produtividade de dois cultivares de soja para sete diferentes épocas de semeadura. Em seguida, o modelo foi utilizado para simular cenários num sistema de produção convencional com nove operações, considerando a semeadura realizada num prazo variando de uma a doze semanas, para três diferentes extensões de áreas e de três faixas de potência para tratores agrícolas. Os resultados das simulações mostraram que o número de máquinas diminuiu com o alongamento da estação de semeadura e com o tamanho dos tratores. Também mostraram que o retorno médio econômico, pela metodologia adotada, foi diferente entre épocas de semeadura e não variou com o tamanho da área e a faixa de potência. Finalmente, que o custo de mecanização diminuiu com o alongamento do prazo de semeadura devido a redução do número de conjuntos e o aumento do número de horas de uso anual do maquinário agrícola.

O conceito de pontualidade está presente em todos os modelos citados. A abordagem e a metodologia adotadas variaram de autor para autor, porém o objetivo das ferramentas desenvolvidas sempre foi obter a seleção otimizada com base numa relação aceitável entre o custo da capacidade das máquinas e das perdas de oportunidade nas operações agrícolas para maximizar a renda líquida do empreendimento. Alguns modelos foram capazes de simular o comportamento econômico de sistemas mecanizados projetados para desenvolver várias operações em empreendimentos policulturais considerando parâmetros agronômicos e climáticos. Outros ainda, também incorporaram simuladores de desenvolvimento fisiológico para estimar a produtividade das culturas na época de colheita e permitir a determinação dos custos de oportunidade associados. A maioria dos modelos, porém, destinou-se a simular o comportamento de sistemas mecanizados para empreendimentos monoculturais, geralmente milho ou soja, ou ainda uma única operação, geralmente semeadura e colheita. Naturalmente essa tendência está associada ao valor econômico das culturas e a importância crítica das operações na rentabilidade do empreendimento. Alguns autores, todavia, ressaltaram que seus modelos sofreram restrições devido à falta de dados relativos aos coeficientes de oportunidade das operações para as culturas tratadas.

2.2 Coeficientes de oportunidade

BURROWS E SIEMENS (1974) definem o custo de oportunidade das operações agrícolas como aquele decorrente o serviço não concluído no tempo ótimo, isto é, que resulta em perda de rendimento da lavoura. Em seu modelo computacional para otimização de maquinaria para lavouras de soja e milho, os autores usaram um coeficiente de perdas linear na produtividade do milho equivalente a $62,7 \text{ kg. ha}^{-1} \cdot \text{dia}^{-1}$ de atraso na semeadura na área do Cinturão do Milho, EUA. Este coeficiente foi obtido a partir da média de alguns anos de observações. Os autores não utilizaram coeficientes para a semeadura da soja face a inexistência de dados experimentais ou observações empíricas à época.

CHANCELLOR e CERVINKA (1974) desenvolveram um trabalho para examinar alguns fatores que influenciam a magnitude do coeficiente de oportunidade para operações de semeadura e colheita de arroz na Califórnia, EUA. Foram analisados os efeitos das datas de maturidade, de início da colheita e a interação das datas de semeadura e colheita sobre os coeficientes de oportunidade. Concluíram que o controle das datas de início das operações, o conhecimento das características das culturas e o aumento da distribuição de datas ótimas para a colheita através do escalonamento por talhões, reduziram as perdas de oportunidade. Concluíram, finalmente, que o conhecimento de interação entre os efeitos das datas de semeadura e colheita permitiu a seleção de combinações de cultivares de tal modo a minimizar os coeficientes de oportunidade e otimizar a seleção de equipamentos. Os autores ressaltaram que a determinação de coeficientes de oportunidade é uma tarefa que consome tempo porque devem ser feitos testes experimentais múltiplos para validá-los. Por conseguinte, a quantidade de dados disponíveis é muito limitado. Além disso, cada valor reflete as condições particulares de colheita, solo e clima nos quais foi obtido de modo que, para ser usado em outros locais, serve apenas como referência para estimativas de perdas. Os autores propuseram um modelo de perdas por oportunidade como sendo uma função linear decrescente antes de um período ótimo que se prolonga sem perdas por alguns dias e, após este, uma função linear crescente.

EDWARDS e BOEHLJE (1980), em seu trabalho de seleção de máquinas através da pontualidade, utilizaram funções quadráticas obtidas a partir de resultados experimentais para estimar a redução de produtividade em função da data de semeadura de milho e soja em várias localizações de Iowa, EUA. Salientaram, todavia, que as equações foram baseadas em

observações experimentais desenvolvidas naquele estado apenas e, por conseguinte, não poderiam ser utilizadas para outras regiões. As perdas de rendimento na semeadura foram calculadas multiplicando percentual de perda média ocorrida após época ótima pelo potencial de produtividade. As perdas de rendimento na colheita foram calculadas multiplicando a percentual de perda diária após a época ótima pelo potencial de rendimento já deduzidas as perdas de oportunidade da semeadura. Finalmente, o custo de oportunidade foi calculado pela soma dos produtos das perdas totais de cada lavoura pelo seu respectivo preço.

HUAN-WEN e HUNT (1985) definiram os custos de oportunidade como perdas de rendimento na safra devido às máquinas que não operaram no tempo apropriado, aí incluídas as reduções de quantidade e qualidade do produto. Com base na primeira aproximação de HUNT (1964), os autores também assumiram o custo de oportunidade como uma função linear decrescente antes e crescente depois do tempo ótimo para execução de uma operação agrícola.

OSKAN e EDWARDS (1986) utilizaram coeficientes de oportunidade derivados de observações experimentais em várias localidades do estado de Iowa (EUA), para estimar o percentual de redução da produtividade em função da data de semeadura e colheita para milho e soja. As perdas de rendimento foram calculadas multiplicando os rendimentos potenciais assumindo a pontualidade na semeadura e colheita pelo fator de redução indicado segundo o atraso nas operações.

WITNEY (1988) conceituou o custo de oportunidade como um custo indireto da maquinaria que representa uma perda financeira decorrente das operações mecanizadas de implantação, pulverização e colheita que ultrapassam os prazos ótimos, causando uma redução de rendimento e na qualidade da safra. Para o autor, as perdas de oportunidade na colheita seguem uma função quadrática. Em sua obra, o autor citou coeficientes de oportunidade obtidos experimentalmente em várias localidades do Reino Unido para operações de semeadura, pulverização e colheita de algumas culturas.

Para OSKAN et al. (1990), a estimativa do custo de oportunidade de uma operação é um dos aspectos mais difíceis e mais restritivos dos modelos de seleção de maquinaria. Os autores sugeriram que, talvez por isso, poucos modelos de seleção de máquinas baseados na pontualidade foram desenvolvidos. Em seu modelo de seleção, os autores informam que fizeram uso de coeficientes genéricos publicados pelo ASAE Standards (1988). Alertaram,

todavia, que embora muito utilizados nos EUA em locais onde não há dados experimentais disponíveis, tais coeficientes devem servir apenas para estimativas, visto que foram obtidos sob condições específicas e não asseguram precisão.

SHORT e GITU (1991) compararam o método desenvolvido por BOEHLJE e EIDMAN (1984) para calcular custo de oportunidade com o método proposto pela American Society of Agricultural Engineers - ASAE (ASAE, 1983). Para a ASAE, o custo de oportunidade depende do valor do produto e da produtividade, e considera que as perdas de oportunidade são aquelas que acontecem fora do período de colheita. Para BOEHLJE e EIDMAN (1984), custo de oportunidade só é afetado pela produtividade. Os autores ressaltaram a seleção de máquinas dependem de diferenças nos custos de oportunidade apurados. No exemplo desenvolvido para ilustrar a comparação, o modelo da ASAE resultou em diferenças no custos de oportunidade superiores a duas vezes aqueles estimados pelo método de BOEHLJE e EIDMAN (1984). Os autores, porém, consideraram que o método de BOEHLJE e EIDMAN (1984) deixou diretrizes pouco claras em relação à seleção de coeficientes de oportunidade adequados. Como os coeficientes de oportunidade propostos em ASAE (1983) resultam de observações experimentais da colheita publicados anualmente, os autores recomendaram a utilização do método por ela preconizado .

Para a American Society of Agricultural Engineers - ASAE (ASAE, 2000 b), a pontualidade em máquinas agrícolas é a capacidade de realizar uma operação agrícola num prazo tal que produção é otimizada considerando quantidade e qualidade de produto. O valor econômico da oportunidade é a redução no rendimento da lavoura decorrente do prolongamento extemporâneo das operações mecanizadas no campo. Para a ASAE, o custo de oportunidade varia amplamente dependendo da região, das culturas, variedades, tipo de operação e prazo de execução. O ASAE Standards D497.4 (ASAE, 2000 b) relaciona valores de coeficientes de oportunidade de operações e culturas válidos para algumas regiões dos EUA.

No Brasil, SEDIYAMA et al. (1972 a), LIN e SEVERO (1982) e VIEIRA et al. (1982) desenvolveram ensaios para avaliar o efeito do atraso na colheita sobre as características de qualidade das sementes de soja porém não propuseram coeficientes ou funções que correlacionassem o atraso com as perdas de produtividade ou qualidade do material colhido.

Para seu o modelo empírico para seleção de máquinas agrícolas na cultura da soja baseado na pontualidade da operação de semeadura, VEIGA (2000) determinou experimentalmente para dois cultivares a função quadrática de produtividade considerando sete diferentes épocas de semeadura.

Os autores citados concordam que a conclusão das operações agrícolas fora do prazo ótimo gera uma perda denominada custo de oportunidade, que é atribuída às máquinas como um custo indireto, relacionada à sua capacidade. Em seus trabalhos, a maioria utilizou coeficientes ou funções de perdas por oportunidade obtidos através de observações experimentais. Porém, ressaltaram as dificuldade na sua obtenção e a restrição no seu uso em outros locais além daqueles onde foram obtidos, visto que resultam de condições específicas da operação, de solo e clima.

2.3 Dias úteis de campo

As perdas por oportunidade dependem do tempo transcorrido após o período agronomicamente ótimo de execução das operações agrícolas. O prazo de conclusão, porém, depende da quantidade e distribuição dos dias úteis de campo, cuja estimativa é uma das limitações de todos os modelos de seleção e otimização de máquinas. O dimensionamento da capacidade adequada seria uma tarefa menos difícil se o administrador pudesse estimar o número de dias disponíveis no intervalo de tempo ótimo. Na prática, a incerteza ou risco climático é contornado com aumento da capacidade instalada de forma que mesmo nos anos de chuva intensa, as operações sejam concluídas no período recomendado. Porém, esta “política de seguro” resulta num aumento do custo fixo da maquinaria que pode não ser compensado pelos eventuais ganhos de pontualidade nas operações.

Dias úteis são definidos como os dias apropriados ao trabalho de campo em um período no qual as operações programadas devem ser executadas. Geralmente as estimativas são obtidas através de modelos de simulação baseados em séries de temperatura e precipitação, balanço da água no solo para determinar a trafegabilidade ou ainda através de séries históricas de dias úteis observados numa região, visando obter a distribuição de probabilidades de ocorrência. Porém, a maioria dos modelos de seleção de maquinaria

converteram esta distribuição numa expectativa de dias úteis para o período calendário de operações.

TULU et al. (1974), por exemplo, utilizaram uma equação de temperatura e precipitação para estimar os dias úteis de campo e aplicaram-na a numa série temporal longa, porém não estabeleceram níveis de probabilidade apropriados ao cálculo da capacidade e desempenho da maquinaria agrícola. Em seu modelo de seleção de máquinas, BURROWS e SIEMENS (1974) determinaram a fração de tempo útil que pode ser esperado num intervalo com uma probabilidade mínima de 80%. BOISVERT (1976) também utilizou séries de temperatura e precipitação para estimar a ocorrência de dias úteis no período de realização das operações com níveis de probabilidade de ocorrência de 70% e 80%.

Outro procedimento para estimativa é através de séries históricas de dias úteis observados numa região. AYRES (1975), citado por OSKAN et al. (1990), usou a cadeia de primeira ordem de Markov numa série para prever o número de dias de úteis de trabalho com sete níveis de probabilidade para colheita de milho.

FULTON et al. (1976) determinaram o número de dias úteis de campo calculados semanalmente para quatro níveis de probabilidade ao longo do período de semeadura do milho séries de dias úteis observados no estado de Iowa (EUA). EDWARDS e BOEHLJE (1980) utilizaram no seu modelo de seleção de máquinas uma série de dias de campo úteis observados entre 1958 e 1977 em Iowa (EUA). Os autores ressaltaram a preferência desta metodologia sobre qualquer outra baseada em simulações com parâmetros climáticos.

OZKAN et al. (1986) utilizaram dados coletados a partir de 1958 pelo Iowa Crop Livestock and Reporting Service para estimar o número de dias de campo esperados com 75% de probabilidade em nove distritos de Iowa (EUA). No modelo, os valores semanais foram divididos por sete para calcular a fração de cada dia útil para as operações realizadas no período. Com base na capacidade de campo das máquinas, calcularam o tempo demandado para cada operação. A data provável de conclusão de cada operação foi calculado pela soma dos dias calendário calculados com base no número de dias úteis esperados e a jornada de trabalho no período. Os autores ainda consideraram no modelo o número de úteis semanais, a jornada de trabalho e o tempo gasto em manutenção para examinar o impacto do atraso na conclusão das operações.

Outra metodologia para determinar o número de dias úteis de campo se baseia nas condições de trafegabilidade a partir da textura e umidade do solo. Em seu modelo de seleção de máquinas considerando o risco climático, WHITSON et al. (1981) determinaram os dias úteis de campo esperados em períodos críticos com níveis de probabilidade de 50%, 75% e 90%, utilizando um simulador de trafegabilidade baseado no balanço da umidade do solo a partir da evapotranspiração. No modelo foi utilizado uma série climatológica de 21 anos de observações e os resultados foram expressos em dias úteis semanais, convertidos em horas disponíveis para cada período crítico considerando a jornada de trabalho.

ROSENBERG et al. (1982), citado por OSKAN et al. (1990), desenvolveu um modelo baseado nas propriedades do solo e observações climáticas para estimar os dias úteis de campo considerando as condições de trafegabilidade e umidade do solo para seis tipos de operações no estado de Michigan (EUA). O modelo foi testado comparando os resultados da simulação com os dias úteis observados revelando boa correlação. OSKAN et al. (1990) associaram o seu modelo de seleção de máquinas TIMECOST a um outro modelo denominado DRAINMOD, capaz de estimar os dias úteis de campo a partir das características do solo e dados climáticos de chuva e temperatura para áreas de solos drenados.

Em seu modelo de seleção de máquinas, VEIGA (2000) utilizou dias agronomicamente secos, isto é, com déficit hídrico na zona das raízes da cana-de-açúcar em solos argilosos e arenosos na região de Piracicaba (SP), mensalmente esperados com 90% de probabilidade. Este critério foi apresentado por MIALHE (1974).

POPP et al. (2003) desenvolveram ensaios de campo para analisar estratégias visando minimizar a influência do risco climático sobre a pontualidade da semeadura de soja no delta do Rio Mississippi (EUA). Para determinar os dias úteis semanais para operação com tratores e a respectiva probabilidade, utilizaram uma série de 40 anos de chuvas e os seguintes critérios: a) após três dias consecutivos de chuva, o terceiro e o quarto dias não foram considerados úteis; b) o dia foi considerado não produtivo se a umidade do solo nos primeiros 10 cm fosse superior a 80% da capacidade de armazenamento; c) o dia também seria considerado não útil para chuvas iguais ou superiores a 38mm.

Tratando exclusivamente de dias úteis para colheita mecanizada de grãos na Inglaterra, WITNEY (1988) pondera que o critério mais simples disponível para estimar o tempo para colheita mecânica são os dias livres de chuva. Este critério, porém, não leva em conta o efeito

da intensidade de chuva na umidade do grão durante os dias que precedem a colheita ou da influência da umidade na lavoura no desempenho de máquina.

PHILIPS e O'CALLAGHAN (1974) e AUDSLEY e BOYCE (1974) contornaram esta restrição em modelos de simulação para colheita de cereais na Inglaterra estabelecendo o critério da “chuva limite”, acima da qual a colheita deveria ser interrompida. PHILIPS e O'CALLAGHAN (1974) estabeleceram que a colheita poderia acontecer numa hora qualquer entre 10 e 19 horas desde que a umidade do grão estivesse abaixo de 24%, nenhuma chuva ocorresse durante a operação, menos de 1,25mm uma hora antes e menos de 2,25 mm duas horas antes do início da colheita.

AUDSLEY e BOYCE (1974) aprimoraram o critério da “chuva limite” introduzindo a “soma descontada de chuva”, na qual a colheita só poderia ter início no dia em que o seu valor fosse menor que 1,27mm. Os autores definiram a “soma descontada de chuva” como a chuva das últimas 24 horas mais 20% da soma descontada do dia anterior. Estudos conduzidos na Escócia Oriental por GLASBEY e McGHECAN (1986) e McGHECAN et al. (1989), resultaram na substituição do critério da “soma descontada de chuva” por uma chuva limite única, ligeiramente mais alta. Uma comparação estatística entre as simulações a partir de registros climáticos e observações de dias de campo úteis permitiu estabelecer que a operação de colheita poderia acontecer quando a chuva nas 24 horas prévias fosse inferior a 1,4 mm.

Os modelos para determinar os dias úteis relatados, de forma generalizada, aplicaram-se às operações diretamente envolvidas com a implantação de lavouras, como a mobilização do solo e a semeadura. Porém, raros trabalhos foram dirigidos para estabelecer critérios específicos para as operações de condução, sobretudo colheita da lavoura. Ademais, embora a quantificação dias úteis de campo para um período crítico de acordo com um critério específico possa ser determinada estocasticamente para níveis de confiança, a sua distribuição num período determinado é um problema muito mais complexo. Esta limitação tem sido contornada em alguns modelos buscando padrões de comportamento climático das chuvas para intervalos geralmente semanais. Todavia, no caso da soja, dependendo do atraso já acumulado e das características arquitetônicas do cultivar, este intervalo pode resultar em perdas muito significativas.

2.4 Pontualidade na colheita da soja

Para a Empresa Brasileira de Pesquisa Agropecuária – Embrapa (EMBRAPA 2002), a colheita constitui uma etapa crítica no processo produtivo da soja, principalmente pelos riscos climáticos a que está sujeita a lavoura destinada à produção de grãos ou de sementes. Recomenda que operação deve ser iniciada tão logo a soja atinja o estágio de desenvolvimento fenológico R8, ou seja, apresentando 95% das vagens maduras. A partir desse ponto de maturidade, a tendência é aumentar a deiscência e a deterioração dos grãos em intensidade proporcional ao tempo que a soja permanecer no campo. Em lavouras destinadas à produção de grãos, além da deiscência, as perdas podem ser maiores devido à redução da qualidade industrial. Em lavouras destinadas a produção de sementes, o retardamento da colheita para reduzir a umidade pode resultar na deterioração das sementes e elevação da incidência de patógenos pela exposição prolongada aos elementos climáticos.

LIN e SEVERO (1982) relataram que quando a colheita é protelada por chuvas intensas e prolongadas, as perdas de rendimento aumentam consideravelmente devido à deterioração e debulha das vagens. Pouco tempo depois da maturação, pode ocorrer a queda natural dos grãos por desgrana ou deiscência das vagens, sobretudo quando a fase da maturação ocorre sob temperaturas altas e umidade baixa. A queda dos grãos também é causada, frequentemente, pela ação de meios mecânicos.

Para SEDIYAMA et al. (1972, b), as perdas de sementes pela deiscência natural das vagens, fragmentação na debulha mecânica e pela diminuição do peso são fatores que diminuem o rendimento econômico da cultura. Ressaltaram, porém, que na maioria das variedades de soja a deiscência não é rápida havendo alguma tolerância no tempo em que as plantas podem permanecer no campo. Recomendaram, todavia, que a colheita seja efetivada tão logo atinja a maturidade, a fim de reduzir a ação das intempéries.

PHILBROOK (1989) conduziu estudos de campo para determinar os efeitos do atraso da colheita de soja sobre as perdas de grãos no campo, na localidade de Arlington (Winsconsin, EUA), nos anos de 1984 a 1986. Foram utilizadas seis cultivares de três grupos de maturação com suscetibilidade ao acamamento. A primeira colheita foi executada entre 3 e 7 dias após o estágio R8 de maturação, seguindo-se três outras aos 14, 28 e 42 dias. A perda média de produtividade potencial observado foi de 10%, mas variou entre 5,5% em 1983 e

12,7% em 1984. A taxa de perda observada variou de 14 kg.ha⁻¹.dia⁻¹ em 1984 e 18 kg.ha⁻¹.dia⁻¹ em 1986. O autor constatou que atrasos de 42 dias causaram a deterioração das plantas resultando em perdas de pré-colheita devidas à deiscência, quebra de talos e quebra de grãos. As condições de umidade durante a colheita foi o fator que mais influenciou as perdas por grãos quebrados, seguido pela deterioração das plantas. Constatou também que os cultivares apresentaram diferentes taxas de perdas e que a relação inversa entre a proporção da perda de rendimento com a produtividade potencial, indicou que eficiência de colheita poderia ser melhorada com altas produtividades.

No Brasil, ensaios de campo também foram realizados para examinar o efeito do atraso na colheita da soja, sobretudo nos parâmetros de qualidade das sementes. SEDIYAMA et al. (1972 a) testaram o cultivar “Viçoja” para estimar o seu tempo de permanência no campo sem comprometer a qualidade e o poder germinativo da semente após a maturação. O ensaio foi realizado em Viçosa (MG) no ano de 1972 e o delineamento experimental consistiu de 6 colheitas semanais a partir do 18º dia após a maturação de 95% das vagens, entre 28/04 e 01/06, com cinco repetições em blocos casualizados. Os resultados mostraram que 32 dias depois da primeira colheita, a produtividade média foi reduzida em 16,4% e a germinação das sementes em 85%.

SEDIYAMA et al. (1972 b) realizaram um trabalho para examinar os efeitos do retardamento da colheita sobre os cultivares de soja “Viçoja”, “Mineira” e “IAC – 2” e determinar as melhores épocas de colheita respectivas. Dentre os parâmetros relacionados, foram observados a produtividade, umidade das sementes, deiscência das vagens, sementes danificadas na debulha mecânica, qualidade das sementes, poder germinativo, percentagem de óleo e proteína bruta. O delineamento experimental do ensaio consistiu de parcelas subdivididas com 4 repetições e 7 épocas de colheita sendo a primeira quando 95% das vagens se apresentavam maduras, e as demais aos 3, 7, 14, 28 e 35 dias depois. Os autores encontraram épocas ótimas de colheita muito distintas entre os cultivares. Para a variedade IAC – 2, o período de colheita se prolongou por duas semanas após o estágio de maturação sem perdas significativas nas variáveis monitoradas. Para a variedade “Viçoja”, o período ótimo de colheita se prolongou por 10 dias após a maturação. Para a variedade “Mineira”, que apresentou deiscência precoce das vagens, o melhor período de colheita foi na semana

seguinte à maturação. Finalmente, os autores verificaram ainda que o retardamento da colheita não afetou significativamente os teores de óleo e proteína bruta das sementes.

Experimento semelhante foi conduzido por VIEIRA et al. (1982) para estudar os efeitos do retardamento da colheita sobre a qualidade da semente da variedade de soja “UFV 2”. O ensaio foi realizado em Viçosa (MG) no ano de 1980 e o delineamento experimental consistiu de blocos causalizados com três repetições e 15 épocas de colheita a partir do estágio R8 de maturação, em intervalos de três dias até 42 dias de atraso, entre 20/04 e 10/06. O retardamento da colheita além de 21 dias após a maturação reduziu significativamente todos os parâmetros de qualidade das sementes. Os autores, porém, não determinaram as perdas de produtividade de em função do atraso na colheita.

LIN e SEVERO (1982) conduziram experimento de campo com o objetivo de estudar os efeitos do atraso da colheita na produtividade e parâmetros de qualidade da semente dos cultivares de soja “Bragg” e “Prata”. O ensaio foi conduzido em Guaíba (RS) no ano de 1978 e consistiu de 5 épocas de colheita para cada cultivar, distribuídas semanalmente a partir do estágio de maturação fisiológica da semente (entre 40% e 50% de umidade), correspondendo a 0, 7, 14, 21 e 28 dias. O pico de produtividade foi obtido na colheita aos 14 dias após a maturidade fisiológica e a partir daí, ambos cultivares apresentaram queda de 26,6% no rendimento médio até a última colheita. Os melhores índices de poder germinativo, vigor e emergência foram encontradas quando as sementes foram colhidas na época da maturação fisiológica, decrescendo significativamente com o atraso da colheita. Os valores significativos da interação entre cultivares e épocas de colheita revelou que cada cultivar respondeu segundo suas características próprias de resistência à deiscência natural, danos mecânicos e tolerância às condições climáticas durante a permanência no campo. Com base nos resultados, os autores concluíram que as condições de temperaturas e umidade elevadas foram os principais fatores que causaram a redução do rendimento e qualidade das sementes de soja.

Ainda no Brasil, MESQUITA et al. (2001, 2002) desenvolveram trabalhos para estudar o perfil da colheita de soja no estado do Paraná e no Brasil. Os trabalhos, porém, tiveram o objetivo de estudar as perdas que ocorrem durante a operação, e não antes dela, por conta de atrasos na operação. Os autores concluíram que os níveis de perdas de grãos, danos mecânicos e grãos quebrados aparentemente independem das marcas e da idade das máquinas avaliadas até 15 anos, embora haja uma tendência maior de perdas com máquinas mais antigas. As

perdas médias superiores à tolerância de 60 kg.ha⁻¹ e de 8% de grãos quebrados observadas durante a colheita, indicaram a provável necessidade de cuidados operacionais e/ou ajustes na colhedora e, possivelmente também, problemas na capacitação da mão-de-obra de operação. Os níveis de grãos quebrados observados aparentemente tenderam a aumentar com a redução do teor de água dos mesmos, independentemente da rotação do cilindro de trilha. Finalmente, as perdas de grãos tenderam a aumentar de forma acentuada com velocidades de trabalho superiores a 7 km.h⁻¹.

2.5 A soja no Estado do Paraná

A primeira notícia que se tem da soja no Brasil data do final do século XIX, na Bahia, mais precisamente em 1882. Foi trazida dos Estados Unidos como uma planta destinada mais ao uso forrageiro do que à produção de grãos para a indústria de farelos e óleos vegetais. Testes de adaptação de cultivares trazidos daquele país realizados pela então “Estação Agronômica de Campinas” (hoje Instituto Agronômico de Campinas) levaram a primeira distribuição de sementes em 1900 para produtores paulistas (MEDINA, 1981). Em 1901 ocorreu o primeiro registro de cultivo de soja no Rio Grande do Sul, no município de Dom Pedrito, e em 1917, no município de Santa Rosa (MAGALHÃES, 1981). Com o estabelecimento do programa oficial de incentivo à triticultura em meados dos anos 50, a cultura da soja foi igualmente incentivada por ser, do ponto de vista técnico (leguminosa sucedendo gramínea) quanto econômico (melhor aproveitamento das máquinas, implementos, infra-estrutura e mão-de-obra), a melhor alternativa de verão para suceder o trigo semeado no inverno (EMBRAPA, 2002).

Como lavoura comercial, a soja chegou no Paraná em meados dos anos 50. Até então, no Estado, a produção era pequena, destinada ao consumo doméstico na alimentação de suínos, principalmente. No Sul do Paraná, a soja surgiu como alternativa para rotacionar com o arroz de sequeiro, cujo manejo era comprometido pela rápida infestação de invasoras após alguns anos de cultivo. No norte e noroeste do Paraná, após as grandes geadas de 1953 e 1955, a soja foi recomendada pelo Instituto Agronômico de Campinas para minorar as consequências socio-econômicas do fenômeno. O incentivo baseou-se na premissa de que se tornava necessário, naquela emergência, contar com uma cultura cuja produção pudesse ser

comercializada a bons preços no mercado internacional e causasse o menor impacto no cafezal em recuperação (KASTER, QUEIRÓZ e TERESAWA, 1981). No sudoeste e oeste do Estado, a cultura desenvolveu-se com a migração de colonos vindos do Rio Grande do Sul, onde a soja era cultivada há mais tempo. A partir da década de 60, o crescimento da produção foi explosivo, passando de 8 mil toneladas em 1960 para 150 mil toneladas na média dos anos 60, para 3,5 milhões de toneladas na média dos anos 70, para 4,15 milhões de toneladas na média dos anos 80, e para 6,5 milhões de toneladas na média dos anos 90. Na safra 2001/02 foram colhidas 9,4 milhões de toneladas, ou 22,4% da produção nacional, sendo ultrapassado apenas pelo estado do Mato Grosso (EMBRAPA, 2002).

2.5.1 Produção e produtividade no Estado e na região de Ponta Grossa

Na Tabela 1 observa-se que, na safra 2001/02, o estado do Paraná produziu 9,407 milhões de toneladas em 3,294 milhões de hectares, representando 22,43 % da produção brasileira desta oleaginosa. Na produtividade, o Paraná supera em 11% a produtividade brasileira. A região de Ponta Grossa representa 9,38 % da área de cultivo e 9,98 % da produção paranaense. Na produtividade, a região supera a do estado em 6,4 % e a brasileira em 18,24 %.

Tabela 1 Evolução da cultura da soja no Estado do Paraná, na região de Ponta Grossa e Brasil - Safra 94/95 a 2001/02 (Fonte: IBGE 2002; SEAB, 2002).

Safra	Paraná			Ponta Grossa			Brasil		
Ano	10. ³ ha	10. ³ t	kg.ha ⁻¹	10. ³ ha	10. ³ t	kg.ha ⁻¹	10. ³ ha	10. ³ t	kg.ha ⁻¹
94/95	2.206	5,694	2.581	203	572	2.818	11.675	25.682	2.200
95/96	2.386	6.440	2.699	209	576	2.756	10.291	23.155	2.250
96/97	2.540	6.582	2.591	216	579	2.681	11.486	26.391	2.298
97/98	2.858	7.313	2.559	244	689	2.824	13.303	31.307	2.353
98/99	2.786	7.752	2.782	249	773	3.104	13.061	30.987	2.372
99/00	2.859	7.199	2.518	250	711	2.844	13.656	32.820	2.403
00/01	2.821	8.628	3.058	253	743	2.937	13.930	37.683	2.705
01/02	3.294	9.407	2.856	309	939	3.039	16.315	41.936	2.570

Isto se deve ao uso de tecnologias inovadoras com o uso intenso da mecanização centrado no sistema de produção em plantio direto, auferindo elevadas produtividades em área de cultivo relativamente pequenas.

2.5.2 Aspectos gerais do manejo

2.5.2.1 Semeadura

A época de semeadura é um dos fatores que mais influenciam a produtividade da soja. Como se trata de uma espécie termo e fotossensível, está sujeita a alterações fisiológicas e morfológicas quando as suas exigências, nesse sentido, não são satisfeitas. A época de semeadura determina a exposição da soja à variação dos fatores climáticos limitantes, como a disponibilidade hídrica, temperatura e fotoperíodo. Assim, semeaduras em épocas inadequadas podem afetar o porte, o ciclo, a produtividade das plantas e aumentar as perdas na colheita. (EMPRAPA, 2002).

No Estado do Paraná, a época de semeadura indicada, para a maioria dos cultivares, estende-se de 15/10 a 15/12. Os melhores resultados, para o rendimento e altura de plantas, na maioria dos anos e para a maioria das cultivares, são obtidos nas semeaduras realizadas de final de outubro a final de novembro. De modo geral, as semeaduras da segunda quinzena de outubro apresentam menor porte e maior produtividade do que as da primeira quinzena de dezembro (EMBRAPA, 2002). Na região de Ponta Grossa, a semeadura é realizada preferencialmente entre 15/10 e 15/11.

O número de cultivares inscritas no Registro Nacional de Cultivares e indicadas para o Estado do Paraná, na safra 2002/03, foi de 91. Destas, cerca de 36 são do grupo de maturação precoce (até 115 dias), 28 semiprecoces (116 a 125 dias), 23 médio (126 a 137 dias), 2 semitardio (138 a 150 dias) e 2 tardio (acima de 150 dias) (EMPBRAPA, 2002).

No Paraná, a maioria dos produtores adotaram a tecnologia de semeadura direta na palha. Na região de Ponta Grossa, pioneira neste sistema, é o padrão há muito adotado pelas vantagens que apresenta. No sistema convencional, o preparo do solo exige um grande número de operações que, se executadas com implementos e condições de solo inadequados, causa

desestabilização dos agregados e redução da matéria orgânica resultando em erosão, com perdas de solo e nutrientes. O sistema de semeadura direta na palha é a melhor alternativa para reverter o quadro de degradação do solo gerado pelo sistema convencional. Reduz a erosão, mantém ou aumenta a matéria orgânica a qual, por sua vez, melhora a CTC e a estabilidade das características físicas dos solos com agregados estáveis, relação adequada entre macro e microporos, retenção de água, níveis de fertilidade, principalmente de fósforo, que resultam em maior produtividade da lavoura. Paralelamente, a semeadura direta reduz os custos de produção em razão da ausência das operações de preparo do solo, permitindo a racionalização do uso de máquinas e implementos e permitindo que outras culturas sejam instaladas nas épocas oportunas (EMBRAPA, 2002).

2.5.2.2 Colheita

A predominância de materiais precoces e semiprecoces inscritas no Registro Nacional de Cultivares e recomendadas para o Estado do Paraná na safra 2002/03, indica uma preferência pelos agricultores. A semeadura da soja entre 15/10 e 15/11 resulta na colheita nos meses de março e abril, sucedendo a colheita do milho sem comprometer a pontualidade para as operações de instalação das culturas de inverno. Porém, a colheita dos cultivares precoces em março, na região de Ponta Grossa, sofre restrições devido à menor disponibilidade de dias de campo úteis para colheita que em abril e maio, quando se colhem os cultivares semiprecoces e médios.

Quando às especificações de desempenho e qualidade do serviço de colheita, as máquinas operam a velocidades de 4,0 a 6,5 km.h⁻¹ devido as restrições de topografia, todavia compatível com as recomendações de 3,0 a 7,0 km.h⁻¹ para perdas inferiores a 60 kg.ha⁻¹ proposto por MESQUITA et al. (2001).

O custo de colheita é resultante da interação de inúmeras variáveis sobre os custos fixos e variáveis da operação. Os valores levantados para o Estado são estimativos e não esclarecem as condições em que a operação é executada, particularmente no que se refere às características da máquina, velocidade média de avanço, eficiência de campo e a logística associada. São resultados do simples produto entre o custo hora máquina para um padrão de

uso anual pelo consumo estimado de tempo máquina por hectare segundo uma produtividade média colhida, sem considerar os custos de oportunidade eventualmente envolvidos. O custo de colheita, pois, é o custo máquina, apenas.

Para a safra de soja 2002/03, no Estado do Paraná, a Secretaria de Estado da Agricultura e do Abastecimento estimou o custo médio de colheita de US\$ 30,95. ha⁻¹ com um consumo de 0,6 hora máquina. ha⁻¹ (SEAB, 2003). A FNP Consultoria e Comércio (FNP Consultoria e Comércio, 2003) estimou um custo de US\$ 26,23. ha⁻¹ considerando um consumo de 0,65 hora máquina. ha⁻¹, para uma produtividade média de 50 sacas de 60 kg. ha⁻¹. [¹]

¹ Cotação média em Agosto de 2002: US\$ 1,00 = R\$ 3,10

3 MATERIAL E MÉTODOS

Para melhor entendimento, este capítulo foi dividido em três partes, cada qual contendo material e métodos respectivos. A primeira, tratou da determinação das perdas devidas ao atraso na colheita que, de acordo com o objetivos do trabalho, referiu-se à cultura da soja (*Glycine max* (L.) Merrill). A segunda, tratou do desenvolvimento do programa computacional para a seleção da colhedora considerando a pontualidade na operação de colheita. A terceira, tratou da validação do programa simulando cenários de colheita de soja numa propriedade referência na região de Ponta Grossa (PR).

3.1 Determinação das perdas por atraso na colheita da soja

Para a determinação das perdas físicas e qualitativas decorrentes do atraso na colheita da soja, optou-se pela instalação de um experimento conduzido como lavoura comercial, cujas variedades cobrissem ciclos de maturação precoce, semiprecoce e média, e fossem economicamente representativas dos cultivos na região de Ponta Grossa (PR).

3.1.1 Localização e solo da área experimental

A área experimental localizou-se no município de Ipiranga, Estado do Paraná (longitude 50° 29' 13" W, latitude 25° 01' 36" S e altitude de 869m), distante 27 km da sede do município de Ponta Grossa e 136 km da sede do município de Curitiba. De acordo com EMBRAPA (1999 a), o solo foi classificado como Latossolo Vermelho Amarelo, de relevo suave ondulado, com as características texturais e químicas para profundidade de amostragem de 0 – 20 cm, conforme resultados da análise apresentados na Tabela 2, determinadas no Laboratório de Solos do Departamento de Solos e Engenharia Agrícola da UFPR.

Tabela 2 Características texturais e químicas do solo da área experimental (0 – 20 cm).

Características Texturais			
Argila	20 %	Areia Grossa	5 %
Areia Fina	34 %	Silte	31 %
Características Químicas			
pH (CaCl ₂)	4,62	MO	3,46 %
P	12,42 mg.dm ⁻³	V	40,39 %
K	0,32 cmol	S	15,31 mg.dm ⁻³
Ca	2,82 cmol	Na	2,16 mg.dm ⁻³
Mg	1,16 cmol	B	0,18 mg.dm ⁻³
Al	0,38 cmol	Fé	57,06 mg.dm ⁻³
H + Al	6,33 cmol	Mn	3,40 mg.dm ⁻³
SB	4,30 cmol	Cu	1,69 mg.dm ⁻³
CTC	10,62 cmol	Zn	3,86 mg.dm ⁻³

3.1.2 Clima

Segundo Koeppen, o clima da região foi classificado como Cfb, ou seja, subtropical úmido, mesotérmico, com verões frescos (temperatura média do mês mais quente < 22 ° C), geadas severas demasiado frequentes (temperatura média do mês mais frio < 18 ° C), sem estações secas (IAPAR, 1978). Na Figura 1 se encontram as médias mensais de temperatura máxima e mínima absolutas e a precipitação pluviométrica observadas entre 1997 – 2002 pelo Instituto Tecnológico – SIMEPAR.

3.1.3 Delineamento experimental

O delineamento experimental adotado foi o de blocos ao acaso com 3 repetições seguindo o esquema de parcelas subdivididas, conforme descrito em PIMENTEL-GOMES (2000). Foi considerado como fator A os cultivares de soja, em número de 4, compondo as

parcelas ou tratamentos (Figura 3). Como fator B, foi considerado as épocas ou atrasos de colheita, em número de 10, compondo as subparcelas ou subtratamentos.

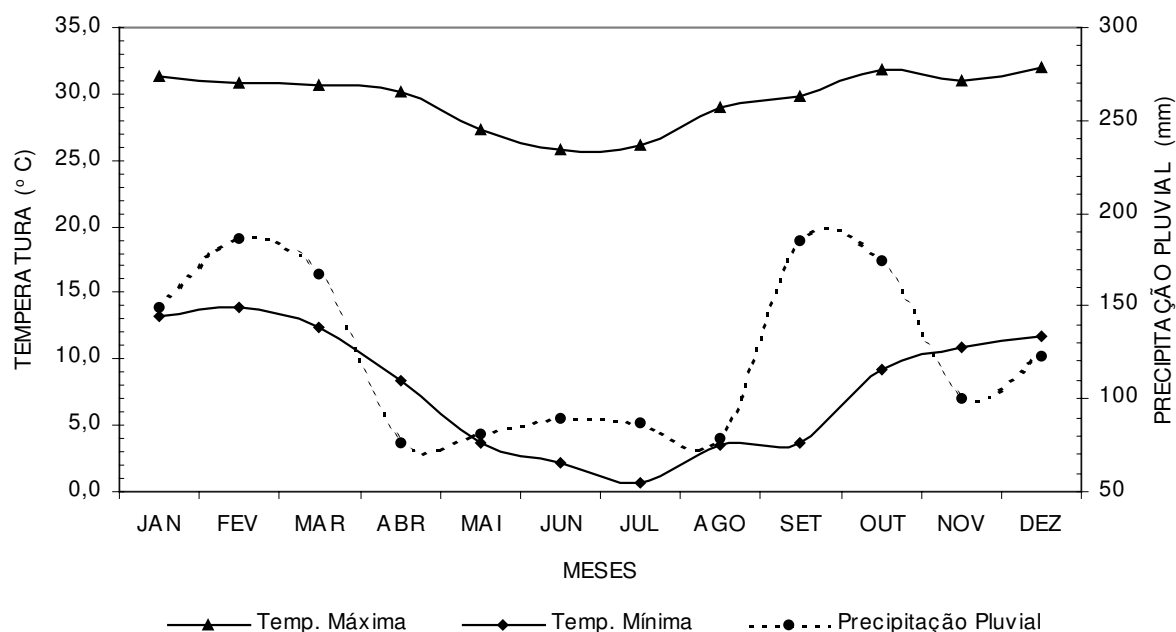


Figura 1 Temperatura absoluta máxima e mínima médias mensais, e precipitação média mensal no período de 1997 - 2002, para a região de Ponta Grossa (Fonte: Instituto Tecnológico – SIMEPAR, 2003).

Os cultivares foram FT Cometa (C1) e M – SOY 5942 (C2), do grupo de maturação precoce (≤ 115 dias) e hábito de crescimento indeterminado; BRS 133 (C3), de ciclo semiprecoce (116 – 125 dias) e FT Abyara (C4), de ciclo médio (126 – 137 dias), ambas de crescimento determinado. Todos os cultivares estavam inscritos no Registro Nacional de Cultivares e recomendados para o Estado do Paraná, segundo a EMBRAPA (1999 b).

O experimento foi implantado e conduzido como lavoura comercial. A sua instalação e condução por conseguinte, subordinou-se à logística disponível na propriedade. A semeadura dos cultivares foi executada em faixas de 76,5m largura obtidas com 17 passadas paralelas ao longo de cada parcela. Posteriormente foram piqueteados os vértices das parcelas e blocos experimentais com 306 m de largura por 160m de comprimento, bem como os acessos de cabeceira (Figura 2 e Figura 3).

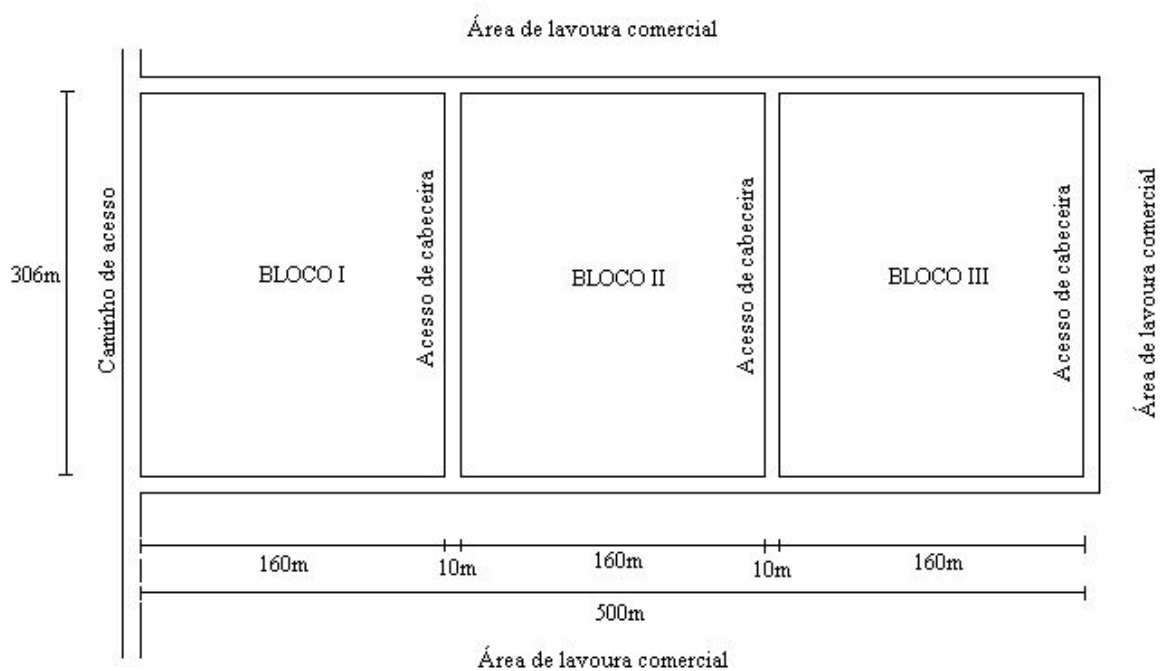


Figura 2 Representação do croqui da área de ensaio.

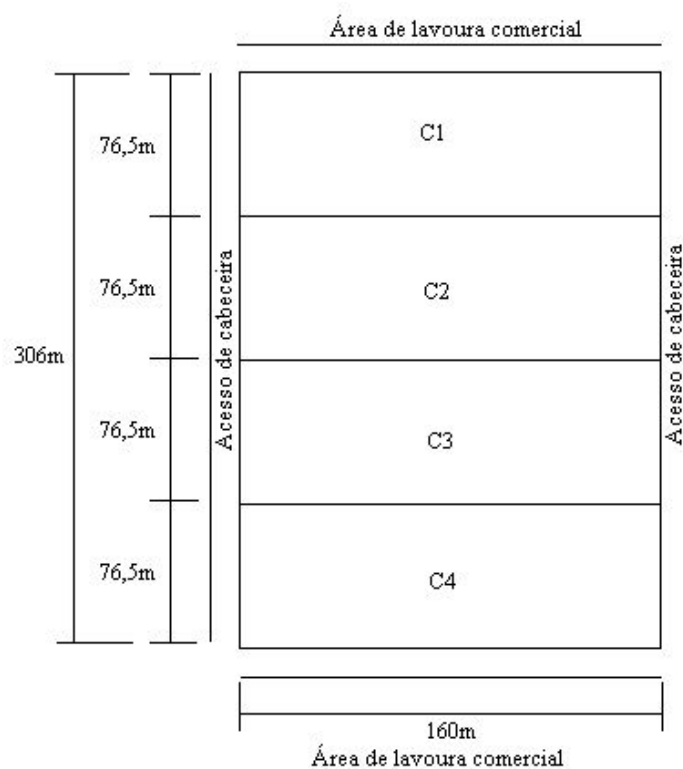


Figura 3 Representação do bloco experimental.

As parcelas foram subdivididas em 12 faixas de 6,37 m de largura por 160m de comprimento constituindo as subparcelas ou unidades experimentais (T0 a T11), 10 das quais foram efetivamente utilizadas (Figura 4). A área útil da subparcela foi de 4,5 m de largura (de corte da plataforma utilizada na colhedora), por 150 m de comprimento (675m^2). Os excedentes em ambas extremidades (5,0m) foram considerados bordadura e no primeiro momento de colheita foram incorporadas aos acessos de cabeceira da parcela (Figura 5).

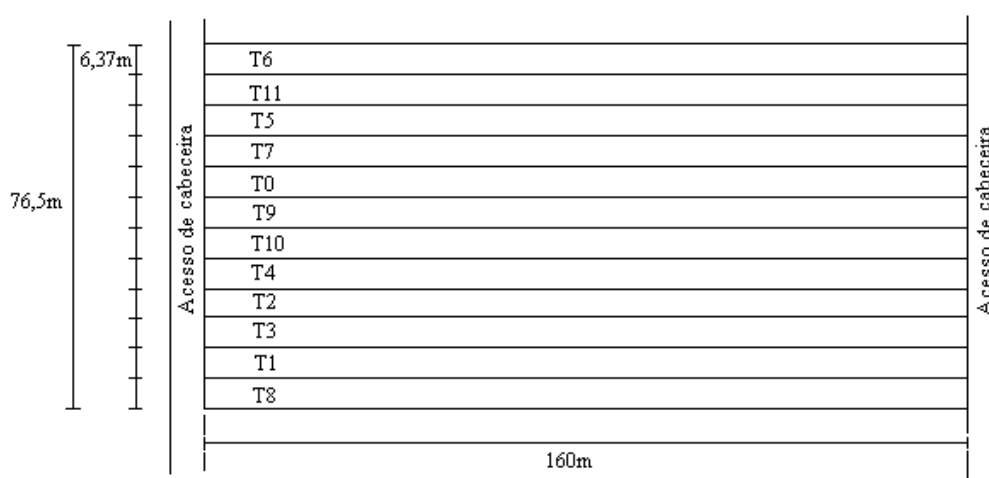


Figura 4 Representação da parcela experimental.

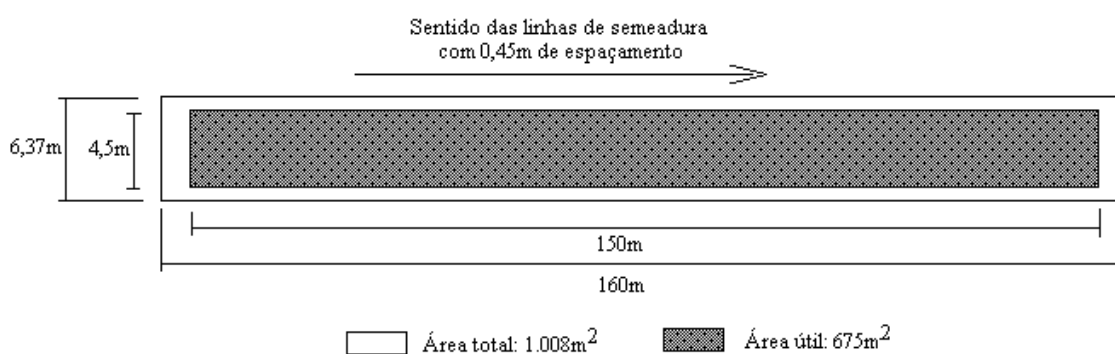


Figura 5 Representação da subparcela experimental.

3.1.4 Implantação e condução do ensaio

O talhão da área experimental foi ocupado no inverno de 2001 com lavoura comercial de cevada cervejeira. Sobre os resíduos desta, a semeadura de todos os cultivares do ensaio realizada em 27/10/2001, adotando-se em seguida o manejo e os tratamentos usuais para soja comercial no sistema de produção de semeadura direta na região, conforme Tabela 3.

Paralelamente, foi observado o desenvolvimento das cultivares segundo a Escala Fenológica descrita por FEHR e CAVINESS (1977), coletando 20 plantas aleatoriamente nas parcelas duas vezes por semana ao longo do ciclo, para identificar o início de cada estágio fenológico da soja. Também foram observados os atributos climatológicos diários de insolação, temperatura média e precipitação, coletados na estação meteorológica de Ponta Grossa, do Instituto Tecnológico – SIMEPAR.

3.1.5 Colheita e processamento do material

A colheita de cada tratamento foi executado em 10 épocas ou atrasos (T0 a T9), buscando preferencialmente intervalos semanais, fazendo-se a primeira (T0) de 12 a 15 dias após ter a cultivar atingido o estágio R8 da escala fenológica. Os subtratamentos foram escolhidos ao acaso nas datas de colheita, balizados e, em seguida, colhidos em todos os blocos.

Foi utilizada uma colhedora New Holland TC55 (99,3kW) com plataforma Superflex de 4,5m de largura de corte e as regulagens principais conforme a Tabela 4, de acordo com recomendações New Holland Latino Americana (New Holland, 1998).

O material colhido em cada subparcela foi imediatamente descarregado e pesado numa carreta graneleira com capacidade para 4.500 kg, apoiada sobre 03 células de carga de 4.500 kg de uma balança eletrônica com subdivisão de 2,0kg, previamente aferida ².

² Marca Vesta, Modelo 3402.

Tabela 3 Eventos da implantação e condução da soja na área experimental.

Data – Evento	Dosagem	Produto Comercial/Alvo
27/10/2001 – Semeadura		
FT – Cometa; M – SOY 5942:	22 sementes.m ⁻¹	
FT – Abyara; BRS 133:	17 sementes.m ⁻¹	
Profundidade:	0,045 m	
Espaçamento entre linhas:	0,45 m	
Tratamento de sementes		
Inoculante:	150 ml.saca 60 kg ⁻¹	Cellthec
Fungicida:	200 ml.100 kg ⁻¹	Vitavax Thiram 200SC (Carboxim + Thiran)
Micronutrientes:	200 ml.100 kg ⁻¹	Basfoliar Comol (Cobalto-Molibdenium)
Fertilizante:	120 kg.ha ⁻¹	KCl
Herbicida pós-plantio:	2,6 l.ha ⁻¹	Roundup Original (Glyphosate, Isopropilamina) – Controle não seletivo do complexo de invasoras da soja.
Herbicida pós-plantio, pré-emergente:	200 g.ha ⁻¹	Scepter 70 DG (Imazaquin) – Controle do complexo de invasoras de folha larga da soja.
21/11/2001 – Aplicação herbicida/inseticida		
Herbicida pós-emergente seletivo:	1,25 l.ha ⁻¹	PoastBASF (Sethoxydim) – Controle de Brachiaria plantaginea (Papuã)
Herbicida pós-emergente seletivo:	0,6 l.ha ⁻¹	Flex (Fomesafen) – Controle do complexo de invasoras de folha larga da soja.
Adjuvante:	0,25 % v/v	Dash HC
Inseticida:	33 g.ha ⁻¹	Dimilin (Diflubenzuron) – Controle de Anticarsia gemmatilis (Lagarta da soja)
20/12/2001 – Aplicação de herbicida		
Herbicida pós-emergente seletivo:	0,35 l.ha ⁻¹	Select 240 CE (Clethodim). Controle de Brachiaria plantaginea (Papuã)
Adjuvante:	0,8 l.ha ⁻¹	Lanzar
08/01/2002 – Aplicação de inseticida		
Inseticida (piretróide):	25ml.ha ⁻¹	Pounce 384CE (Permethrin). Controle de Anticarsia gemmatilis (Lagarta da soja); Nezara viridula (Percevejo verde); Piezodorus guildinii (Percevejo verde pequeno da soja); Euschistus heros.
Inseticida (fisiológico):	50ml.ha ⁻¹	Nomolt 150 (Teflubenzuron) Controle de Anticarsia gemmatilis (Lagarta da soja)

Tabela 4 Regulagens principais da colhedora utilizada (Fonte: New Holland Latino Americana Ltda, 1998).

Item	Unidade	Valores
Motor	rotações.min ⁻¹	2.200
Velocidade média de avanço	km.h ⁻¹	4,5
Controle automático de altura e flutuação lateral da plataforma	-	Acionado
Rotação do Sem-fim	rotações.min ⁻¹	177
Rotação na caixa de navalhas	rotações.min ⁻¹	550
Rotação do cilindro de debulha	rotações.min ⁻¹	750
Abertura do côncavo	-	6ª posição
Rotação do Rotary Separator	rotações.min ⁻¹	400
Abertura do côncavo do Rotary Separator	mm	30
Abertura da peneira inferior	mm	8
Abertura da peneira superior	mm	10
Abertura da extensão da peneira superior	mm	15
Rotação do ventilador	rotações.min ⁻¹	900

Do material colhido e pesado, foram coletadas duas amostras homogêneas com massa mínima de 1,0 kg e acondicionados para análise imediata da qualidade em laboratório conforme a Norma de Qualidade para Classificação e Comercialização da Soja em Grão estabelecido pela Portaria Nº. 262 de 23/11/1982, do Ministério da Agricultura. Foram determinados os parâmetros umidade (padrão: 14%); grãos quebrados (máximo: 30%); impurezas e/ou matérias estranhas (máximo: 1,0%); grãos avariados (máximo: 8,0%) e grãos esverdeados (máximo: 10,0%). O fator umidade, de acordo com o procedimento usual adotado pelas moageiras de soja da região, recebeu desconto de 1,5% para cada 1,0% encontrado na amostra acima do padrão básico. Descontados os excedentes conforme os padrões acima, a massa colhida na subparcela foi convertida em produtividade de soja comercial expressa em kg.ha⁻¹.

Aplicou-se análise de variância segundo PIMENTEL-GOMES (2000) para examinar as diferenças estatísticas obtidas no ensaio. Finalmente, baseado em EDWARDS e BOELHJE (1980), WITNEY (1988) e VEIGA (2000) ajustou-se os dados coletados a função quadrática para expressar as perdas da produtividade de soja comercial segundo o atraso aplicado na colheita de cada cultivar até o limite de 45 dias.

3.2 Desenvolvimento do programa computacional

Para a seleção colhedora considerando a pontualidade na colheita, foi desenvolvido um programa computacional capaz de simular a operação de colheita interagindo as funções de perdas por atraso na operação, risco climático e características da máquina. O programa foi elaborado utilizando linguagem de programação orientada ao objeto Borland® Delphi 5.0 e um programa para produção de diagramas da UML (Unified Modeling Language) Rational Rose 2000, buscando uma interface acessível ao usuário. Foi utilizado um processador Athlon XP 1700+. A metodologia e o programa desenvolvido são a seguir descritos.

3.2.1 Metodologia de desenvolvimento

As primeiras versões foram desenvolvidas utilizando algumas ferramentas da UML (Unified Modeling Language). Essa metodologia descrita por FURLAN (1998), permitiu a criação de modelos que especificaram o domínio dos problemas do sistema. Para isso, empregou-se uma série de diagramas textuais e/ou gráficos que descreveram o comportamento e a funcionalidade do mesmo. Essas ferramentas possibilitaram a melhor compreensão dos requisitos verificando se as exigências do usuário foram atendidas e proporcionando uma visão antecipada do programa, o que permitiu classificá-lo como satisfatório ou não ao usuário. O emprego dessa metodologia foi, pois, essencial para obter um desempenho satisfatório durante as etapas preliminares de desenvolvimento.

Também foi utilizado o diagrama de entidades e relacionamentos descrita por YOURDON (1990), que ofereceu um excelente suporte para a definição da estrutura do banco de dados relacional. Apesar de não pertencer à UML, essa ferramenta foi fundamental para entender a sua importância. Devido à praticidade e facilidade de compreensão, empregou-se o fluxograma que, não sendo essencial ao desenvolvimento, serviu para representar relativamente bem o funcionamento do sistema.

3.2.2 Funcionalidade do sistema

O usuário dispõe de duas opções de simulação: PROPRIEDADE e CENÁRIO. A simulação PROPRIEDADE corresponde à situação real de uma fazenda que supostamente existe. O usuário está habilitado a especificar uma frota que utiliza diferentes marcas e modelos de colhedoras, informando as respectivas quantidades e velocidades de operação. Define, por conseguinte, a capacidade instalada e o ritmo de operacional. O usuário não especifica, porém, o prazo de colheita, que será limitado pelo ritmo ou capacidade da frota. Com isso, o processamento avança o calendário até que a área de lavoura seja totalmente colhida. Portanto, a variável funcional da simulação PROPRIEDADE é o prazo de colheita.

A simulação CENÁRIO é uma modificação da PROPRIEDADE no qual o sistema não permite a combinação de diferentes modelos de colhedoras. Essa limitação é consequência da natureza do dimensionamento proposto, que procura relacionar a capacidade da máquina com o prazo de colheita. É necessário, pois, que toda a operação seja realizada por um modelo de colhedora apenas. O usuário somente especifica o modelo de colhedora e a respectiva velocidade de operação. Isto, porém, não define a capacidade instalada. Para defini-la, é necessário informar o prazo de colheita. O calendário é fixo, portanto. O ritmo de trabalho e, por conseguinte, a capacidade expressa pelo número de colhedoras são definidos com base no número de dias úteis compreendidos nesse prazo. Logo, o número de colhedoras é a variável funcional na simulação CENÁRIO. Ressalte-se, porém, que número de colhedoras resultante da simulação CENÁRIO é um valor inteiro, considerando-se no custo do serviço as frações de utilização produtiva e improdutiva.

CENÁRIO, pois, é inverso de PROPRIEDADE. Naquele, define-se um prazo para determinar o número de colhedoras, enquanto neste, define-se a quantidade de colhedoras para determinar um prazo, de modo que ambos compõe um par de simulações cujos resultados se complementam.

3.2.2.1 Entrada de dados

O usuário inicia a série de simulações especificando inicialmente o cenário de colheita na opção PROPRIEDADE. O sistema requisita a seleção da região agroclimática, o turno de trabalho, a precipitação crítica e o período de simulação. São atributos da região agroclimática, além da sua identificação por nome e sigla, o turno de trabalho máximo e o período recomendado de semeadura do cultivar. Em seguida, o usuário seleciona a colhedora, especificando o número de máquinas e a velocidade de trabalho. São atributos da colhedora a marca, modelo, potência motora, largura da plataforma e o preço de aquisição. O usuário deve ainda especificar os talhões a serem colhidos selecionando os cultivares, as áreas, as datas de semeadura e datas prováveis de início da colheita respectivos. São atributos dos cultivares o ciclo de maturação e os coeficientes das funções de perdas respectivos. Finalmente, o usuário deve também especificar os indicadores econômicos do cenário atribuindo valores ao preço unitário do produto, salário do operador e combustível, vida útil do equipamento e valor residual, taxas de juros, seguro e alojamento e eficiência de campo. Com isso, o sistema cria o arquivo temporário que contém as definições da propriedade e está pronto para iniciar o processamento.

A simulação CENÁRIO pressupõe a existência de uma simulação PROPRIEDADE. Nesta opção, o sistema recupera a simulação contendo vínculos, atributos e resultados respectivos criando o arquivo temporário que contém as definições da propriedade original. Desta forma, usuário está habilitado para alterar o turno de trabalho, selecionar outra marca e modelo de colhedora e ainda alterar a definição dos talhões, com novas variedades, área de cultivo, data de semeadura, início e término da colheita (prazo). Com isso, o sistema cria o arquivo temporário contendo as definições do cenário e está pronto para iniciar o processamento.

Os respectivos fluxogramas de entrada de dados das simulações PROPRIEDADE e CENÁRIO encontram-se na Figura 6 e Figura 7, respectivamente.

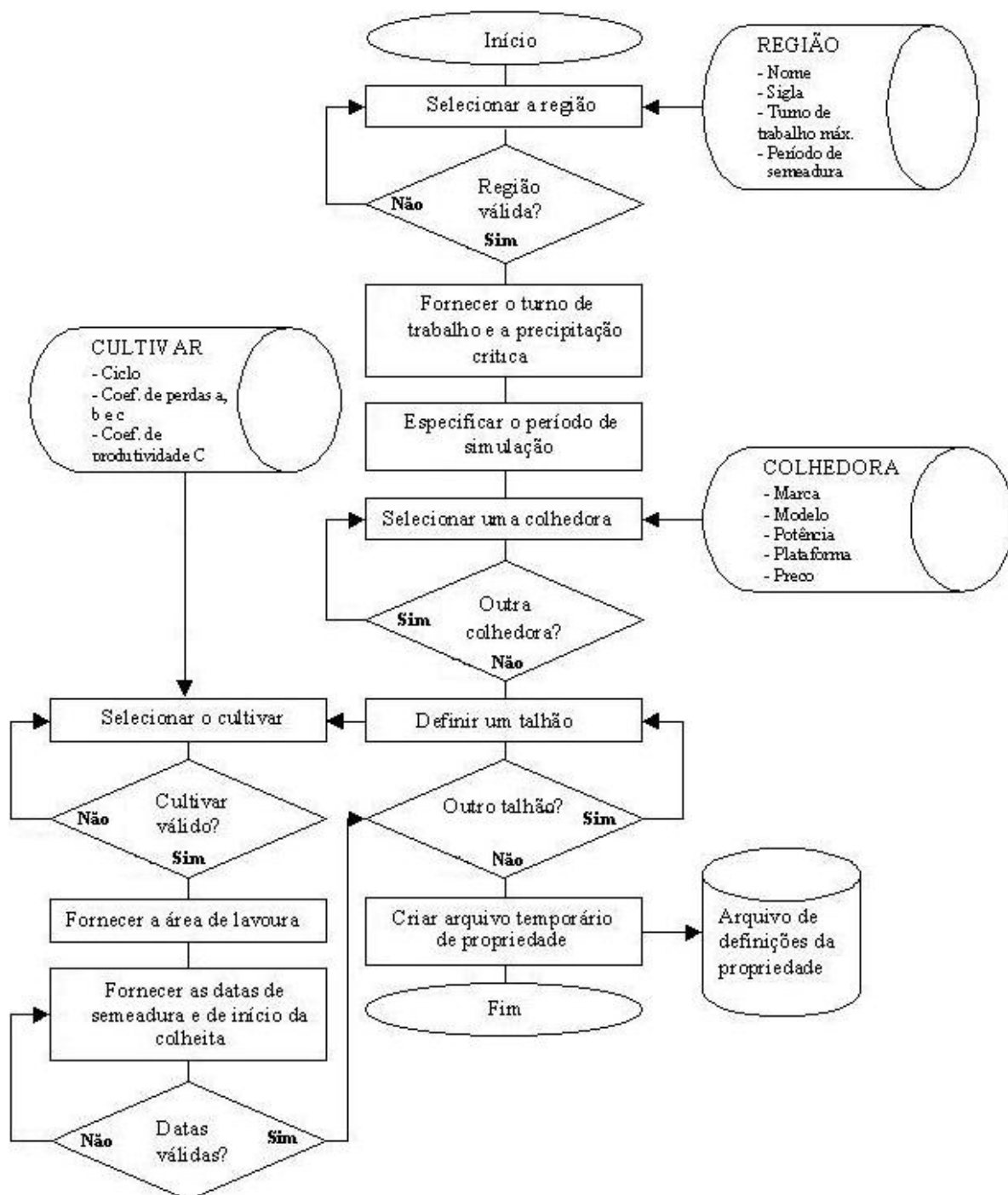


Figura 6 Fluxograma da entrada de dados da simulação PROPRIEDADE.

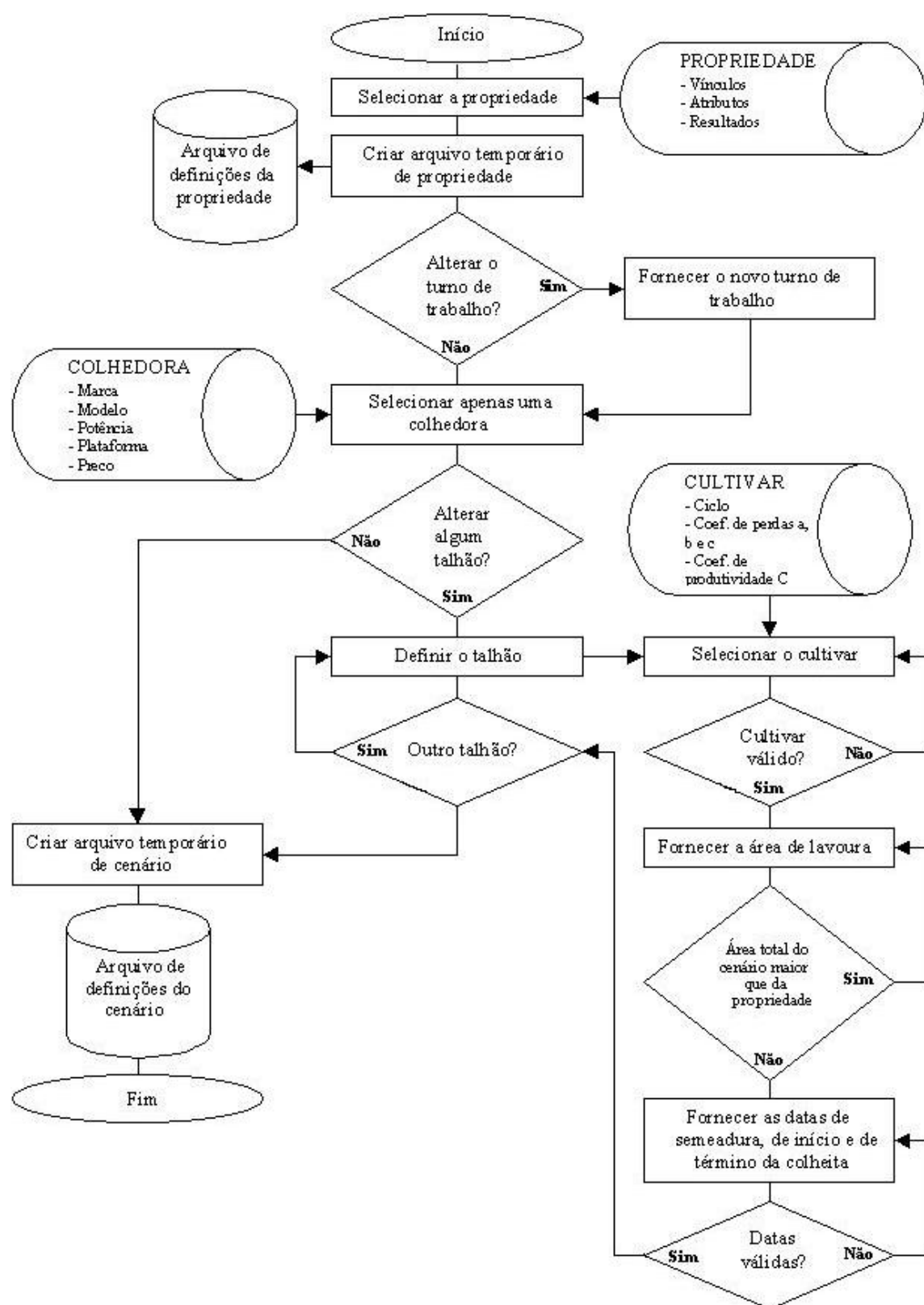


Figura 7 Fluxograma da entrada de dados de simulação CENÁRIO.

3.2.2.2 Processamento

O processamento da simulação PROPRIEDADE inicia com a leitura dos atributos armazenados no arquivo de definições da propriedade e o cálculo da capacidade ou ritmo operacional da frota de colhedoras, calculado segundo ASAE (2000 a). Segue então para a simulação da colheita nos anos do período selecionado usando um banco de dados que contém uma série histórica de chuva diária da região agroclimática selecionada, e um critério para classificar os dias úteis para a operação, baseado em GLASBEY e McGHECAN (1986) e McGHECAN et al. (1989). O dia será considerado útil para colheita quando a chuva acumulada do dia anterior com a do dia atual for menor que a precipitação crítica especificada no cenário. Assim, o processamento avança a operação através do calendário a partir da data de início da colheita num ritmo operacional igual à capacidade da frota especificada, até o limite de 44 dias de atraso. Com base no tempo gasto para a conclusão da colheita, o sistema calcula o custo das perdas por atraso, o custo do serviço mecanizado, custo da colheita e a renda líquida da operação do ano simulado.

O custo das perdas por atraso é o somatório das perdas diárias do saldo diário da área não colhida, calculadas segundo a função de perdas do cultivar utilizado no talhão. O custo do serviço mecanizado é calculado com base no custo horário de propriedade e utilização do equipamento adotando a metodologia descrita pela ASAE (2000 b) e utilizando os indicadores econômicos pertinentes especificados no cenário. O custo de colheita é calculado pela soma do custo das perdas por atraso com o custo do serviço mecanizado. Finalmente, a renda líquida, é a diferença entre o potencial de renda da área (atraso nulo) e o custo de colheita.

A rotina é aplicada aos demais anos até concluir a série de simulação. Concluída a série, o sistema efetua o tratamento estatístico e cria o arquivo temporário contendo os resultados da simulação PROPRIEDADE.

O processamento da simulação CENÁRIO se inicia com a leitura dos atributos armazenados no arquivos de definições do cenário. Segue então para a simulação da colheita nos anos do período selecionado aplicando ao prazo calendário especificado o critério de identificação de dia útil para calcular o ritmo operacional, a capacidade operacional requerida expressa pelo número de colhedoras, as perdas por atraso no período, o custo do serviço mecanizado, custo da colheita e a renda líquida do ano simulado.

Conforme a recomendação de HUAN-WEN e HUNT (1985), o programa expressa a quantidade de colhedoras em números inteiros. As frações, portanto, foram arredondadas para o número inteiro imediatamente acima. Esta medida requereu a adoção do “fator de utilização horária” da frota dimensionada. Este recurso há muito é usado na construção rodoviária para determinação do custo horário de equipes mecanizadas (MINISTÉRIO DOS TRANSPORTES - DNER, 1972). É denominado “custo horário produtivo” a fração utilizada da capacidade da frota, e “ custo horário improdutivo” a fração não utilizada. Na fração produtiva o programa contabiliza todos os custos horários de propriedade e utilização. Na fração improdutiva, apenas os custos horários de propriedade acrescido da mão-de-obra de operação. As parcelas do custo horário de propriedade e utilização também foram calculados pela metodologia ASAE (2000 b). Assim, o custo horário da frota resulta dos seus fatores de utilização pelos respectivos custos produtivo e improdutivo. E o custo direto do serviço mecanizado resulta do custo horário calculado na forma descrita pelo tempo de utilização anual na colheita da soja. É evidente, todavia, que o arredondamento da fração para a unidade imediatamente acima acarreta uma “distorção” do custo do serviço mecanizado e, por conseguinte, da renda líquida.

Ressalte-se, ainda, que a receita líquida produzida por uma frota dimensionada para colher uma área num prazo calendário definido sempre vai diferir daquela utilizada com prazo livre porque o fator de utilização daquela sempre será inferior à unidade, enquanto que esta, será igual à unidade.

A mesma rotina é aplicada aos demais anos até concluir a série de anos da simulação. Concluída a série, o sistema efetua o tratamento estatístico e cria o arquivo temporário contendo os resultados da simulação CENÁRIO.

Finalmente, é importante ressaltar também que ambas simulações envolvem a colheita de uma lavoura anual, apenas. Por conseguinte, a expectativa de tempo de uso anual do equipamento será baixo, resultando na redução da renda líquida devido à elevação do custo do fixo serviço mecanizado. Numa propriedade real, ao contrário, as máquinas utilizadas na colheita das lavouras de verão soja são as mesmas utilizadas na colheita lavouras de inverno, o que eleva o tempo de utilização anual e reduz o custo do serviço.

Os fluxogramas de processamento das simulações PROPRIEDADE e CENÁRIO encontram-se na Figura 8 e Figura 9, respectivamente.

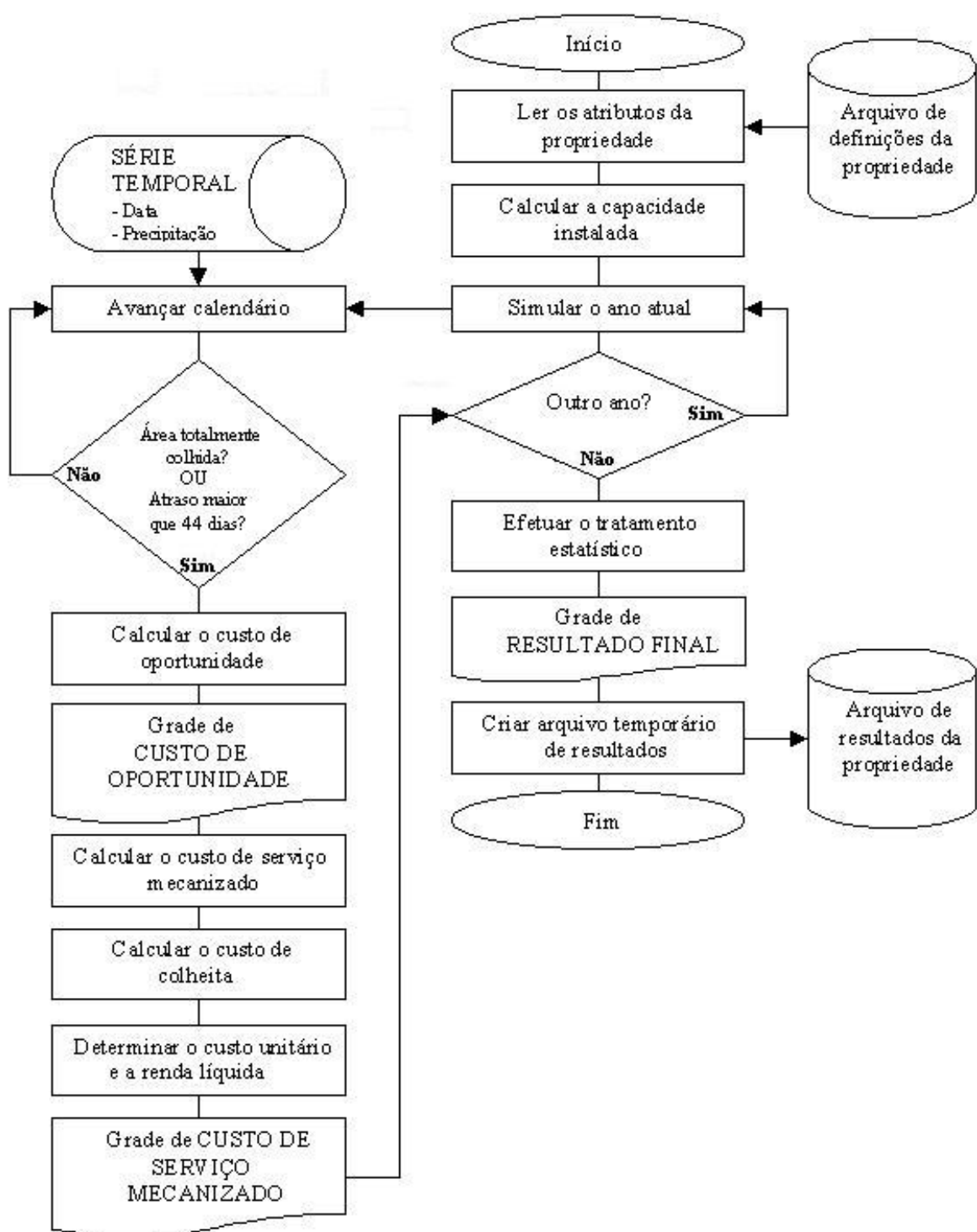


Figura 8 Fluxograma do processamento da simulação PROPRIEDADE.

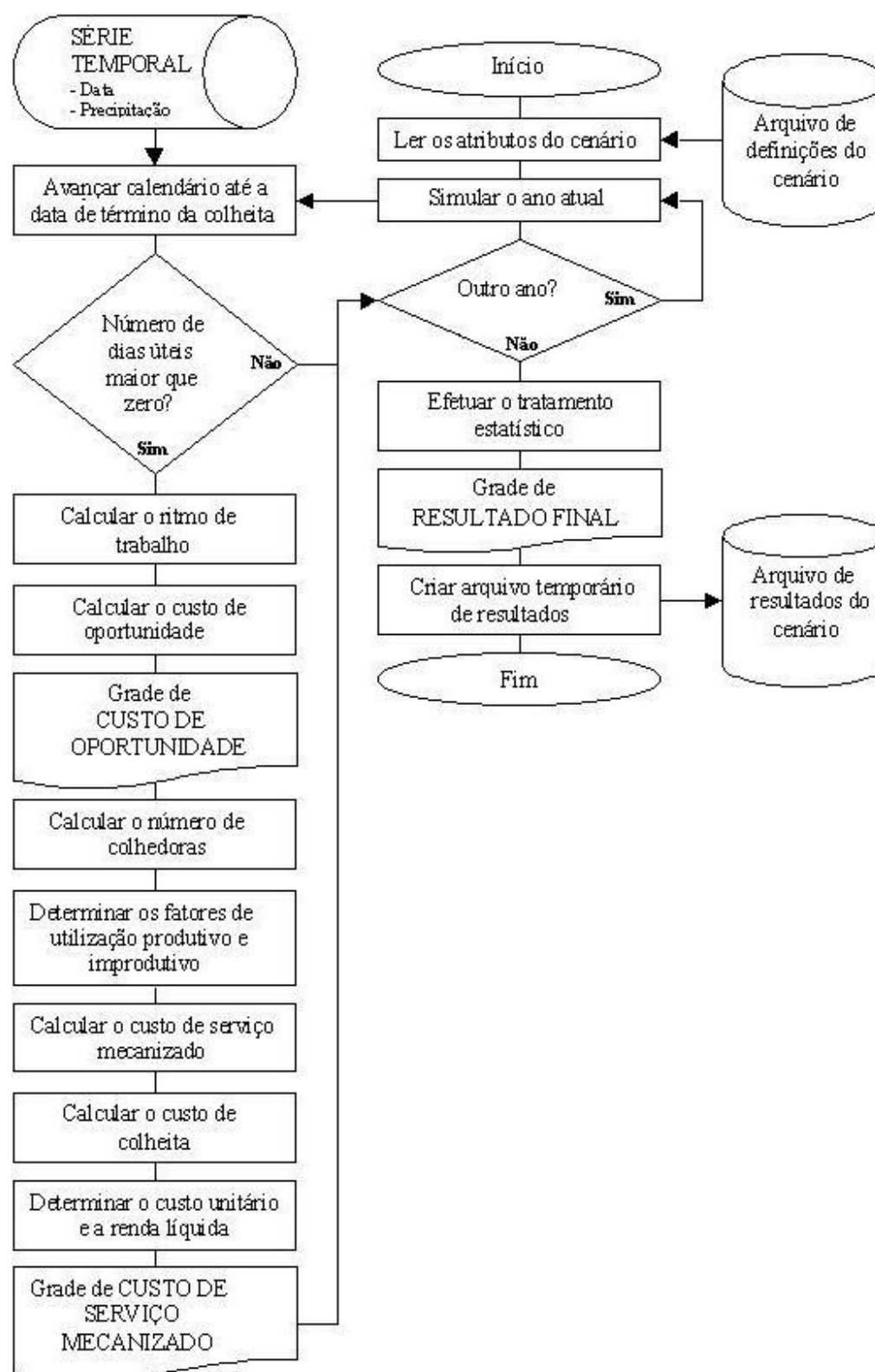


Figura 9 Fluxograma do processamento da simulação CENÁRIO.

3.2.2.3 Saída de dados

Ao término do processamento, dispõe-se do comportamento da colheita em cada ano do período simulado para PROPRIEDADE e CENÁRIO. O sistema efetua um tratamento estatístico simples contendo os valores máximo, mínimo, médio e o desvio padrão das perdas, do custo de colheita, da renda líquida e do custo unitário do serviço dos anos válidos da série simulada, apresentados numa grade de resultados.

Além dessas variáveis, para as simulações CENÁRIO, a grade contém ainda os detalhes de cada talhão como o prazo de colheita, o número de dias úteis e a distribuição do número de colhedoras pela frequência acumulada de demanda na série de anos simulada e os respectivos fatores de utilização produtivo e improdutivo.

O programa possibilita o armazenamento das simulações de PROPRIEDADE e CENÁRIO com seus vínculos, atributos e resultados, bem como a exportação da grade de resultados respectivos para uma planilha do Microsoft® Excel de forma que possam ser melhor examinados. Os fluxogramas da Figura 10 e Figura 11 ilustram o processo de armazenamento e exportação.

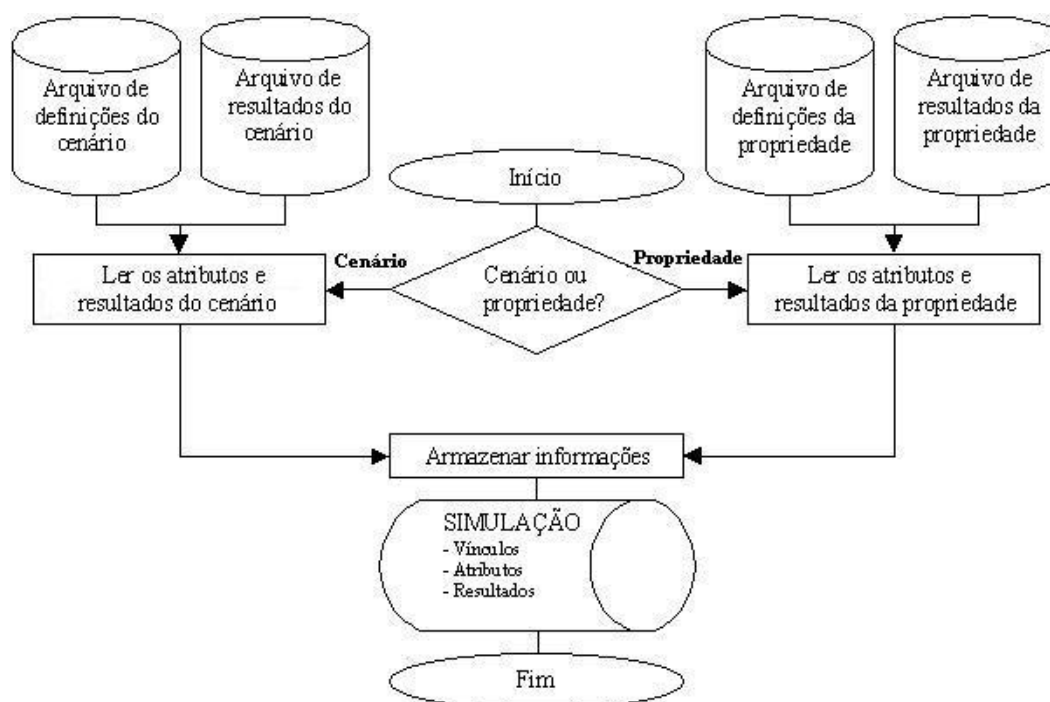


Figura 10 Fluxograma de armazenamento da simulação.

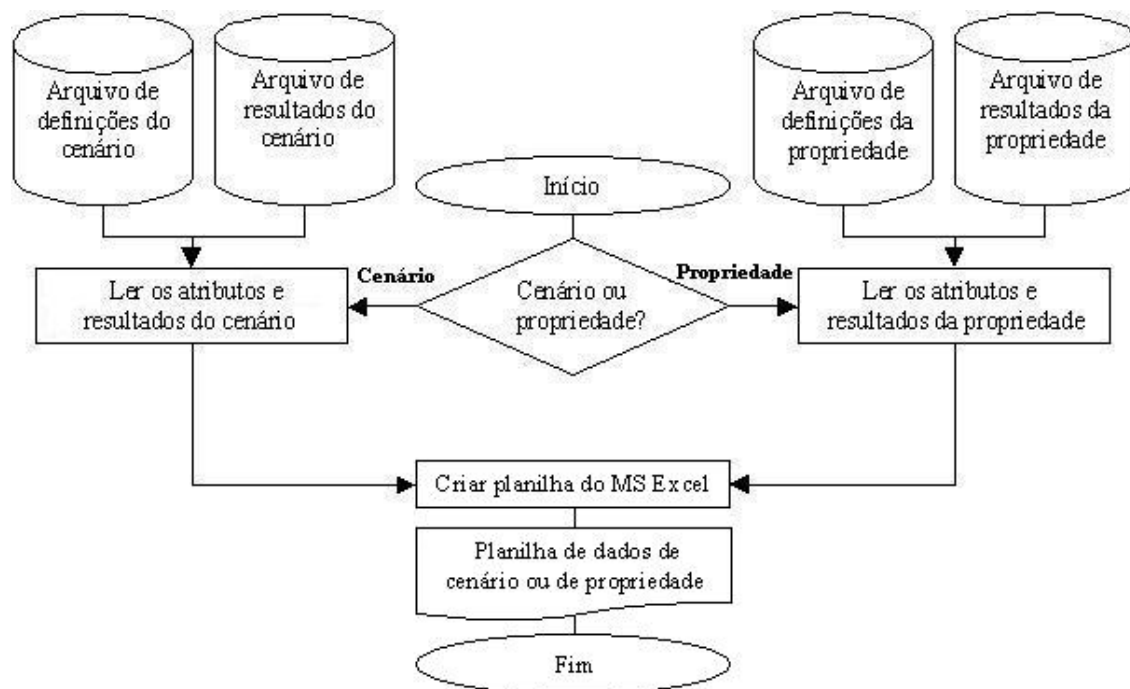


Figura 11 Fluxograma de criação de planilha do Microsoft® Excel.

3.2.2.4 Apresentação das telas

3.2.2.4.1 Hierarquia

Na Figura 12 está representada a hierarquia de acesso às telas do programa. O início se dá na tela SIMULAÇÃO. A partir dela é possível navegar para as telas de PROPRIEDADE e CENÁRIO bem como para REGIÃO, COLHEDORA e CULTIVAR. Por sua vez, das telas PROPRIEDADE e CENÁRIO para os INDICADORES e EXPORTAÇÃO. A descrição das telas encontram-se a seguir.

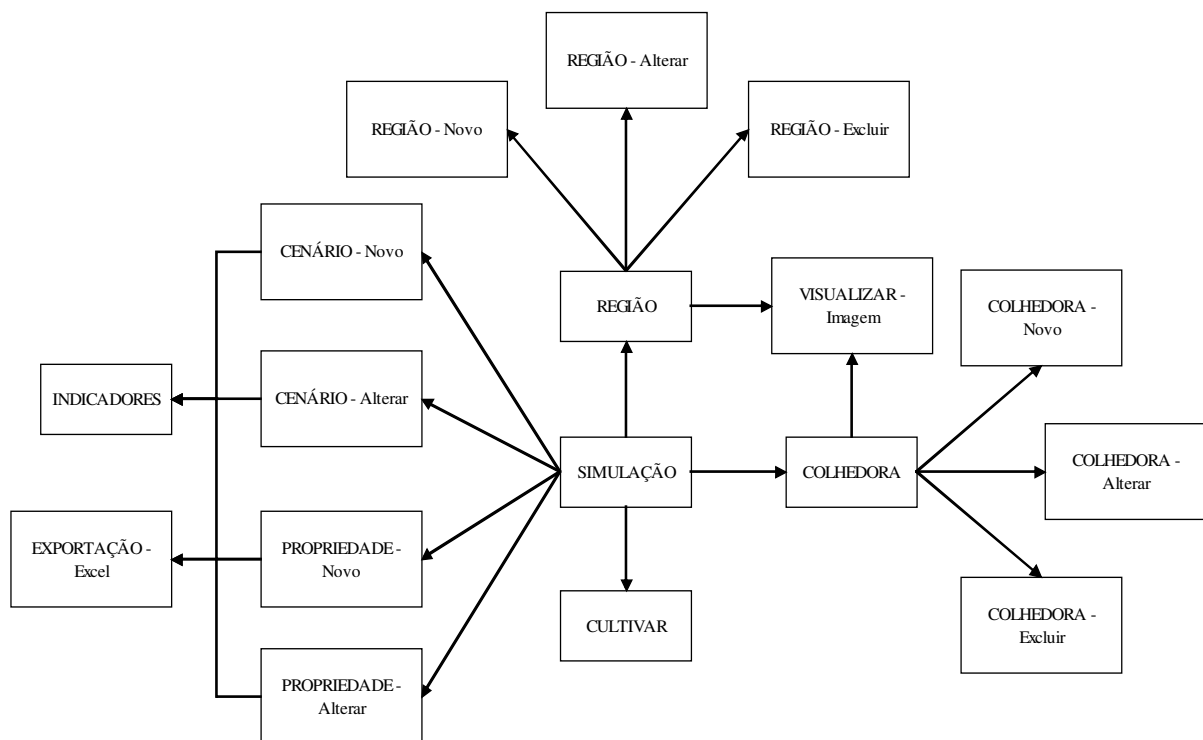


Figura 12 Hierarquia de acesso às telas do programa.

3.2.2.4.2 Descrição das telas

3.2.2.4.2.1 Tela SIMULAÇÃO

É a tela principal do sistema (Figura 13). Seu propósito é o gerenciamento de simulações (menu Opções: Propriedade ou Cenário). Possibilita pesquisar simulações armazenadas (botão Pesquisar) e acessar as demais telas do sistema. Através dela, navega-se às telas de cadastramento (menu Exibir) de REGIÃO (Figura 20), COLHEDORA (Figura 21) e CULTIVAR (Figura 22); abrem-se as telas de criação ou alteração de simulações CENÁRIO (Figura 14 e Figura 15) e PROPRIEDADE (Figuras 16 e 17); exclui-se uma simulação previamente selecionada (botão Excluir) e finaliza-se o programa (botão Sair).

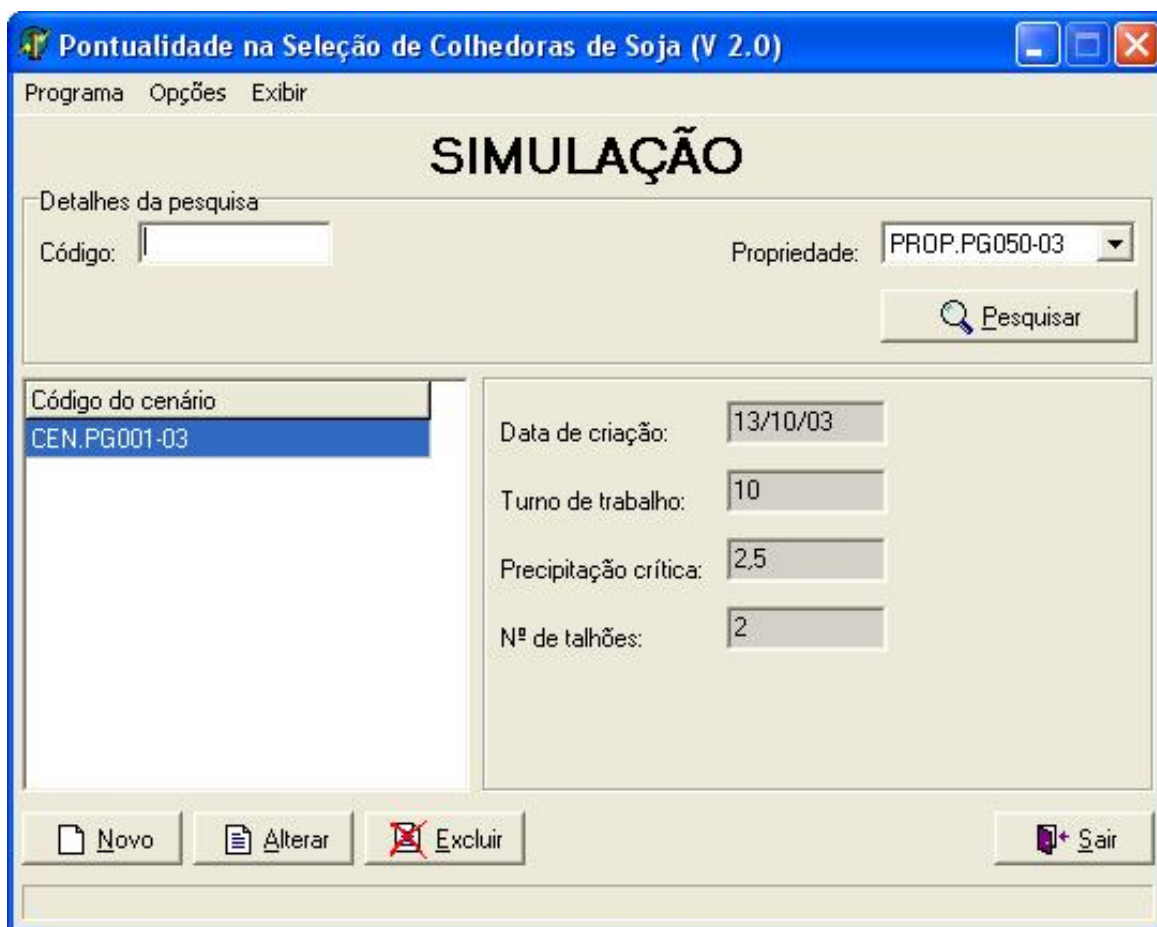


Figura 13 Tela SIMULAÇÃO.

3.2.2.4.2.2 Tela CENÁRIO-Novo

Esta tela tem como propósito a criação de novas simulações CENÁRIO (Figura 14). A partir dela é possível acessar as janelas de edição dos indicadores econômicos (Figura 18) e de exportação para planilha do Microsoft® Excel (Figura 19). Um novo cenário é gerado a partir da propriedade selecionada, cujas informações (atributos e resultados) são recuperadas e exibidas. É permitida a alteração de quaisquer desses atributos, exceto a precipitação crítica, o período de simulação e a área total. Quando o cenário é gerado, é possível salvá-lo (menu Cenário: Salvar). Pode-se iniciar a criação de um novo cenário através da opção Novo (menu Cenário: Novo).

Os resultados da série de anos simulados são mostrados por meio de três grades. A primeira (guia Custo de Oportunidade) relaciona resultados de perdas e produção na colheita. A segunda (guia Custo de Serviço Mecanizado) contém informações sobre o custo do serviço mecanizado produzido pela frota. A terceira e última (guia Resultado Final) exibe a estatística da perda total, renda líquida, custo de colheita por área e por unidade de produção, custo do serviço mecanizado, tempo de utilização anual e número de colhedoras. Exibe ainda dados do talhão relativos ao prazo de colheita, número de dias úteis, e da colhedora, como número, a distribuição de frequência do número de máquinas e respectivos fatores de utilização.

Pontualidade na Seleção de Colhedoras de Soja (V 2.0)

Cenário Editar

CENÁRIO - Novo

Código: Cenário Custo de Oportunidade Custo de Serviço Mecanizado Resultado Final

Detalhes do cenário

Propriedade:

Turno de trabalho:

Precipitação crítica:

Período de simulação

Primeiro ano:

Último ano:

Colhedoras do cenário

- ☐ TC 55 (plataforma de 3,9 metros)
- ☐ TC 55 (plataforma de 4,5 metros)
- ☐ TC 57 (plataforma de 5,1 metros)
- ☐ TC 57 (plataforma de 5,7 metros)
- ☐ TC 57 HYDRO (plataforma de 5,1 metros)
- ☐ TC 57 HYDRO (plataforma de 5,7 metros)
- ☐ TC 59 (plataforma de 7 metros)
- ☐ MF 3640 (plataforma de 4,2 metros)
- ☐ MF 3640 (plataforma de 4,8 metros)
- ☐ MF 5650 (plataforma de 5,1 metros)
- ☐ MF 5650 (plataforma de 5,7 metros)
- ☐ MF 8780 (plataforma de 3,1 metros)
- ☐ MF 34 (plataforma de 7 metros)
- ☐ MF 38 (plataforma de 7,6 metros)
- ☐ MF 38 (plataforma de 3,1 metros)

Talhões do cenário

Talhão	Cultivar	Grupo de maturação	Área de lavoura (ha)	Ciclo de maturação (dias)	Data de semeadura	Data provável de colheita	Início da colheita	Término previsto	Atraso inicial (dia)
A	BRS 133	SEMI-PRÉCOCE	800	170	27/10/02	15/04/03	15/04/03	09/05/03	0

Figura 14 Tela CENÁRIO-Novo.

3.2.2.4.2.3 Tela CENÁRIO-Alterar

O objetivo desta tela é permitir a alteração das simulações CENÁRIO (Figura 15). Quando aberta, ela recupera e exibe as atuais informações (atributos e resultados) do cenário que foi selecionado na tela Simulação (Figura 13). Pode-se alterar qualquer atributo, exceto a propriedade à qual o cenário está vinculado e os valores de precipitação crítica, período de simulação e área total de lavoura. Existem ainda as possibilidades de salvar as alterações efetuadas (menu Cenário: Salvar) e acessar as janelas de edição de indicadores econômicos (Figura 18) e de exportação para planilha do Microsoft® Excel (Figura 19).

Pontualidade na Seleção de Colhedoras de Soja (V 2.0)
Cenário Editar

CENÁRIO - Alterar

Código: CEN.PG001-03

Cenário | Custo de Oportunidade | Custo de Serviço Mecanizado | Resultado Final

Nº de anos da série:	50				
Nº de anos válidos:	50				
ESTATÍSTICA					
Variável	Menor valor	Maior valor	Média	Desvio padrão	
Perda total (US\$/ha)	11,67	31,90	23,81	5,15	
Renda líquida (US\$/ha)	483,35	524,51	499,76	10,43	
Custo de colheita (US\$/ha)	113,79	135,39	126,40	5,33	
Custo médio (US\$/sc)	2,15	2,62	2,42	0,12	
Custo de serviço mecanizado (US\$/ha)	101,96	107,76	102,59	0,97	
Tempo de utilização anual (h)	330,00	620,00	465,60	64,97	
Número de colhedoras	1,00	1,00	1,00	0,00	
DETALHES DO TALHÃO					
	Talhão A	Talhão B			
Prazo (dias)					
Menor valor	23,00	28,00			
Maior valor	36,00	36,00			
Média	35,14	35,06			
Desvio padrão	2,10	1,71			
Nº de dias úteis					
Menor valor	9,00	13,00			
Maior valor	29,00	36,00			
Média	21,58	24,98			

Grade de resultados estatísticos

Confirmar Cancelar

Figura 15 Tela CENÁRIO-Alterar.

3.2.2.4.2.4 Tela PROPRIEDADE – Novo

Esta tela permite a criação de novas simulações PROPRIEDADE (Figura 16). Através dela é possível acessar as janelas de alteração dos indicadores econômicos (Figura 18) e de exportação para planilha do Microsoft® Excel (Figura 19). É necessário que a propriedade seja vinculada a uma região válida cuja série temporal apresenta, no mínimo, um ano cadastrado. Também é necessário fornecer o turno de trabalho e a precipitação crítica, o período de simulação, especificar a frota de máquinas e definir os talhões da propriedade. Quando gerada, é possível salvar a propriedade (menu Propriedade: Salvar) e/ou iniciar a criação de uma nova (menu Propriedade: Novo).

Assim como nas simulações CENÁRIO, os resultados de PROPRIEDADE são exibidos através de três grades. A primeira (guia Custo de Oportunidade) relaciona resultados de perdas e produção na colheita. A segunda (guia Custo de Serviço Mecanizado) contém informações sobre gastos com a frota de máquinas. A terceira grade (guia Resultado Final) exibe, além de informações de prazo de colheita sobre o número de dias úteis, a estatística dos valores da perda por atraso total, renda líquida, custo de colheita por área e por unidade de produção.

3.2.2.4.2.5 Tela PROPRIEDADE-Alterar

O propósito desta tela é permitir a alteração das simulações PROPRIEDADE (Figura 17). Quando aberta, ela recupera e exibe as atuais informações (atributos e resultados) da propriedade que foi selecionada na tela SIMULAÇÃO (Figura 13). É possível alterar qualquer atributo, exceto a região à qual a propriedade está vinculada. Há possibilidade de salvar as alterações efetuadas (menu Propriedade: Salvar) e de acessar as janelas de edição dos indicadores econômicos (Figura 18) e de exportação para planilha do Microsoft® Excel (Figura 19).

Pontualidade na Seleção de Colhedoras de Soja (V 2.0)

Propriedade Editar

PROPRIEDADE - Novo

Código: [Propriedade] Custo de Oportunidade Custo de Serviço Mecanizado Resultado Final

Detalhes da propriedade

Região: PG
 Turno de trabalho:
 Precipitação crítica:

Período de simulação

Primeiro ano: 1954
 Último ano: 1954

Colhedoras da propriedade

- ☐ TC 55 (plataforma de 3,9 metros)
- ☐ TC 55 (plataforma de 4,5 metros)
- ☐ TC 57 (plataforma de 5,1 metros)
- ☐ TC 57 (plataforma de 5,7 metros)
- ☐ TC 57 HYDRO (plataforma de 5,1 metros)
- ☐ TC 57 HYDRO (plataforma de 5,7 metros)
- ☐ TC 59 (plataforma de 7 metros)
- ☐ MF 3640 (plataforma de 4,2 metros)
- ☐ MF 3640 (plataforma de 4,8 metros)
- ☐ MF 5650 (plataforma de 5,1 metros)
- ☐ MF 5650 (plataforma de 5,7 metros)
- ☐ MF 8780 (plataforma de 9,1 metros)
- ☐ MF 34 (plataforma de 7 metros)
- ☐ MF 38 (plataforma de 7,6 metros)
- ☐ MF 38 (plataforma de 9,1 metros)

Talhões da propriedade

Talhão	Cultivar	Grupo de maturação	Área de lavoura (ha)	Ciclo de maturação (dias)	Data de semeadura	Data provável de colheita	Início previsto da colheita	Atraso inicial (dias)
A								

Confirmar Cancelar

Lista de talhões

Figura 16 Tela PROPRIEDADE-Novo

Pontualidade na Seleção de Colhedoras de Soja (V 2.0)

Propriedade Editar

PROPRIEDADE - Alterar

Código: PROP.PG217-03

[Propriedade] Custo de Oportunidade Custo de Serviço Mecanizado Resultado Final

Propriedade

Região: PG
 Turno de trabalho: 10
 Precipitação: 2,5

Período de simulação

Primeiro ano: 1954
 Último ano: 2003

Colhedoras da propriedade

- ☐ TC 55 (plataforma de 3,9 metros)
- ☐ TC 55 (plataforma de 4,5 metros)
- ☐ TC 57 (plataforma de 5,1 metros)
- ☐ TC 57 (plataforma de 5,7 metros)
- ☐ TC 57 HYDRO (plataforma de 5,1 metros)
- ☐ TC 57 HYDRO (plataforma de 5,7 metros)
- ☐ TC 59 (plataforma de 7 metros)
- ☐ MF 3640 (plataforma de 4,2 metros)
- ☐ MF 3640 (plataforma de 4,8 metros)
- ☐ MF 5650 (plataforma de 5,1 metros)
- ☐ MF 5650 (plataforma de 5,7 metros)
- ☐ MF 8780 (plataforma de 9,1 metros)
- ☐ MF 34 (plataforma de 7 metros)
- ☐ MF 38 (plataforma de 7,6 metros)
- ☐ MF 38 (plataforma de 9,1 metros)

Talhões da propriedade

Talhão	Cultivar	Grupo de maturação	Área de lavoura (ha)	Ciclo de maturação (dias)	Data de semeadura	Data provável de colheita	Início previsto da colheita	Atraso inicial (dias)
A	FT-COMETA	PRECOCE	250	128	27/10/02	04/03/03	04/03/03	0
B	FT-ABYARA	MÉDIO	250	166	09/11/02	24/04/03	24/04/03	0

Confirmar Cancelar

Lista de colhedoras

Figura 17 Tela PROPRIEDADE-Alterar

3.2.2.4.2.6 Tela INDICADORES

Esta tela permite a alteração dos valores dos indicadores que são utilizados como parâmetros econômicos de contorno dos cenários simulados (PROPRIEDADE ou CENÁRIO) (Figura 18).



A imagem mostra uma janela de software intitulada "Pontualidade na Seleção de Colhedoras d...". O título da janela é "INDICADORES". A interface contém dois grupos de campos de entrada. O primeiro grupo, "Saca de soja", possui um campo "Preço:" com o valor "12". O segundo grupo, "Colhedora", possui seis campos: "Juros:" (12), "Salário:" (150), "Vida Útil:" (10), "Seg + Aloj:" (3,2), "Eficiência:" (75) e "Diesel:" (0,45). Abaixo dos campos, há dois botões: "Confirmar" (com um ícone de checkmark verde) e "Cancelar" (com um ícone de X vermelho).

Saca de soja	
Preço:	12

Colhedora			
Juros:	12	Salário:	150
Vida Útil:	10	Seg + Aloj:	3,2
Eficiência:	75	Diesel:	0,45
Residual:	20		

Confirmar Cancelar

Figura 18 Tela INDICADORES

3.2.2.4.2.7 Tela EXPORTAÇÃO-Excel

Esta tela tem como propósito a exportação da grade de resultados finais a uma planilha do Microsoft® Excel (Figura 19). Há duas possibilidades de exportação (grupo Opções de Exportação). A primeira opção cria uma nova pasta de trabalho, enquanto que a segunda utiliza uma já existente. Neste caso, é necessário selecionar o arquivo (*.xls) no qual a planilha será adicionada.



Figura 19 Tela EXPORTAÇÃO-Excel.

3.2.2.4.2.8 Tela REGIÃO

O objetivo desta tela é o gerenciamento do cadastro de regiões (Figura 20). Ela possibilita pesquisar regiões armazenadas (botão Pesquisar); registrar datas na série temporal (guia Série Temporal); acessar as telas de inclusão, alteração e exclusão de regiões; e também retornar à tela anterior – SIMULAÇÃO (Figura 13) – através do botão Fechar.

Em cadastro de série temporal, é necessário informar à qual região as datas serão vinculadas. A entrada de dados ocorre a partir de um período especificado informando-se a primeira e a última data a serem registradas. O calendário então avança requisitando as precipitações diárias ocorridas até que a última data seja registrada.

Ao excluir determinada região, a série temporal, as funções de perdas e as propriedades que estão associadas a ela serão também removidas. Além disso, como cenário parte de propriedade, a remoção desta implica apagar todos os cenários dela provenientes.



Figura 20 Tela REGIÃO.

3.2.2.4.2.9 Tela COLHEDORA

Esta tela tem como propósito o gerenciamento do cadastro de colhedoras caracterizadas pela marca, modelo, potência, largura da plataforma e preço de aquisição do equipamento padrão (Figura 21). Permite pesquisar colhedoras armazenadas (botão Pesquisar) e acessar as telas de inclusão, alteração e exclusão de colhedoras. Através do botão Fechar é possível retornar à tela anterior – SIMULAÇÃO (Figura 13).

Ao excluir uma colhedora, os resultados de custo de serviço mecanizado a ela associados serão também removidos.

Cadastro de Colhedora

COLHEDORA

Detalhes da pesquisa

Marca: Modelo:

 Pesquisar

Marca	Modelo	Potência (kw)	Plataforma (m)
MASSEY FERGUSON	MF 8780	206	9,1
MASSEY FERGUSON	MF 34	172	7
MASSEY FERGUSON	MF 38	206	7,6
MASSEY FERGUSON	MF 38	206	9,1
JOHN DEERE	JD 1165	103	4,2
JOHN DEERE	JD 1175	125,1	4,8



Figura 21 Tela COLHEDORA.

3.2.2.4.2.10 Tela CULTIVAR – Função de Perdas

O objetivo desta tela é gerenciar o cadastro de funções de perdas (Figura 22). Ela permite pesquisar as funções armazenadas. Se a função de perdas não existir, há possibilidade de incluí-la. Caso já esteja cadastrada, podem-se alterar os seus atributos ou removê-la. Há possibilidade de retornar à tela anterior – SIMULAÇÃO (Figura 13) – através do botão Fechar.

Cadastro de Cultivar

CULTIVAR - Função de Perdas

Detalhes da pesquisa

Região: PG Cultivar: FT-COMETA

Pesquisar

Ciclo de maturação: 128

Coeficiente de perda (a): 0,0105

Coeficiente de perda (b): -0,3032

Coeficiente de perda (c): 2,1868

Coef. de produtividade (C): 48,803

Novo Alterar Excluir Fechar

Figura 22 Tela CULTIVAR-Função de Perdas.

3.3 Validação do modelo computacional

A validação do modelo computacional foi feito através de duas séries simulações de cenários de colheita tomando como referência uma propriedade na região de Ponta Grossa. A primeira série – PROPRIEDADE - teve a finalidade de analisar a frota de colhedoras da propriedade referência e cada modelo isoladamente, para determinar-lhes a capacidade ótima para diferentes composições de cultivares de soja, independente do prazo para conclusão do serviço. A segunda série – CENÁRIO - teve a finalidade de dimensionar a demanda de colhedoras de cada modelo para as mesmas composições, porém estabelecendo prazos para conclusão da operação.

3.3.1 Condições contorno dos cenários

Os cenários de colheita foram delimitados basicamente por composições dos cultivares de soja cujas funções de perdas por atraso na colheita foram determinados experimentalmente, colhedoras, área de cultivo, prazos de conclusão da operação e parâmetros gerais especificados pelos indicadores econômicos, turno de trabalho, velocidade de operação das colhedoras, precipitação crítica e período de simulação, a seguir descritos.

3.3.1.1 Composições de cultivares

Todos os cultivares ensaiados foram utilizados para formar cinco composições usualmente praticadas na propriedade referência, mostradas na Tabela 5. No programa, as datas de semeadura dos cultivares podem ser cadastradas entre 15/10 e 15/11, que é o período ótimo para realizar a operação na região. Todavia, nas simulações as datas de semeadura adotadas foram 27/10 para o primeiro talhão ou talhão único, e 09/11 para o segundo talhão. Embora o sistema permita subdividir talhões de mesmo cultivar, as simulações não levaram em conta o ritmo e a capacidade das máquinas na operação de semeadura. Portanto, os talhões foram considerados como inteiramente semeados na mesma data. Finalmente, quando selecionado o cultivar atribuído ao talhão, programa informava a data prevista para o início da colheita com base na data de semeadura e no ciclo do cultivar cadastrado na região.

3.3.1.2 Colhedoras

Embora o programa contenha no banco de dados o cadastro de todas as marcas e modelos de colhedoras comercializadas no Brasil, três categorias de potência e largura de plataforma de corte foram utilizadas: 125,1 kW, com plataforma de 4,8 m; 132,5 kW, com plataforma de 5,7 m e 165,6 kW, com plataforma de 7,0 m de largura, todas de uma mesma marca. Na propriedade referência, a frota de colhedoras se compõe de 2 unidades de 125,1 kW e 1 unidade de 165,6 kW.

Tabela 5 Composição de cultivares utilizadas nas simulações.

Composição	Cultivar	Grupo de maturação	Proporção	Data de semeadura	Início da colheita
I	FT Cometa	Precoce	50%	27/10/02	04/03/03
	M-Soy 5942	Precoce	50%	09/11/02	08/04/03
II	FT Abyara	Médio	50%	27/10/02	11/04/03
	BRS 133	Semiprecoce	50%	09/11/02	28/04/03
III	FT Cometa	Precoce	50%	27/10/02	04/03/03
	FT Abyara	Médio	50%	09/11/02	11/04/03
IV	FT Cometa	Precoce	100%	27/10/02	04/03/03
V	FT Abyara	Médio	100%	27/10/02	11/04/03

3.3.1.3 Área de cultivo

Nas simulações PROPRIEDADE, as áreas de cultivo variam de 300 a 3.700 ha, com incrementos de área 300 ha, para avaliar a capacidade da frota, e de 200 a 1.700 ha com incrementos de área de 100 ha, para avaliar a capacidade de cada colhedora. Já nas simulações CENÁRIO, para determinar a demanda de colhedoras em função o do prazo para conclusão do serviço, as áreas variaram de 200 a 1.000 ha, com incrementos de área de 100 ha. Na propriedade referência, a área cultivada com soja é de 500 ha.

3.3.1.4 Prazos de conclusão da operação

O prazo para a conclusão da operação se refere ao período calendário a partir das datas de início da colheita especificados na Tabela 5.

Nas simulações PROPRIEDADE, o prazo foi determinado pelo programa considerando a capacidade da frota especificada. Nas simulações CENÁRIO, os prazos foram de 10 – 15 – 20 – 25 – 30 – 35 dias.

Em ambas simulações, o sistema observou ainda como limites para conclusão da operação, o prazo máximo de 45 dias e a data de 31/05. O primeiro por estar associado às funções de perdas determinadas e, o segundo, porque a colheita além desta data interfere nas operações de implantação das lavouras de inverno na região.

3.3.1.5 Parâmetros gerais

3.3.1.5.1 Indicadores econômicos

Para todas as simulações, os indicadores econômicos foram padronizados conforme a Tabela 6, tomando-se como base os valores médios praticados no mercado. Particularmente difícil foi a atribuição de valor à eficiência de campo. Inúmeras variáveis intervêm na operação de colheita passando pelas características da máquina quanto à capacidade teórica e manobrabilidade, características da área de trabalho quanto a natureza do solo, topografia, forma e extensão, características da lavoura quanto ao sistema de produção, condições da lavoura, rendimento, sobretudo da logística da operação. HUNT (1986) especifica valores entre 63 e 81% e a ASAE (2000 b), de 65 a 80%, com um valor típico de 70%. Para este trabalho, em se tratando da lavoura de soja na região de Ponta Grossa, logística de colheita com transbordo e considerando que o sistema exclui os dias considerados impróprios para a operação, a eficiência de campo adotada foi de 75% em todos os cenários.

Tabela 6 Indicadores econômicos utilizados.

Indicador	Unidade	Valor
Preço da soja	US\$. sc ⁻¹ 60kg	12,00
Mão-de-obra de operador	US\$.mês ⁻¹	150,00
Preço do Diesel	US\$.L ⁻¹	0,45
Vida útil do equipamento	Anos	10
Taxa de juros	% aa	12
Valor residual do equipamento	%	20
Seguro + Alojamento	% aa	3,2
Eficiência de campo	%	75

3.3.1.5.2 Turno de trabalho

O turno de trabalho adotado para todas as simulações foi de 10 horas.dia⁻¹, que corresponde à jornada média diária praticada na região de Ponta Grossa. Turnos superiores são possíveis, porém geralmente ocorrem na colheita de milho cuja umidade de início e final de período não compromete significativamente o desempenho da colhedora nem da qualidade do material colhido.

3.3.1.5.3 Velocidade de operação

Para todas as simulações, as colhedoras operaram no intervalo de velocidade recomendado por MESQUITA et al (2001) que resultam, para esta variável, perdas mecânicas inferiores a 60 kg.ha⁻¹. Considerando as restrições fisiográficas e as velocidades médias usualmente praticadas na região, para colhedoras de 125,5 kW e plataforma de 4,8 m, a velocidade adotada foi de 4,5 km.h⁻¹; para colhedoras de 132,5 kW e plataforma de 5,7 m, 5,0 km.h⁻¹ e, para a colhedora de 165,6 kW e plataforma de 7,0 m, 6,0 km.h⁻¹. Nas simulações, porém, não foram consideradas reduções da velocidade em virtude do estado da lavoura em função do atraso, particularmente a incidência de inços e ao acamamento das plantas de soja.

3.3.1.5.4 Precipitação crítica

Nas simulações, adotou-se um critério baseado em GLASBEY e McGHECAN (1986) e McGHECAN et al. (1989) para classificar o dia como útil para a operação de colheita. Embora o programa permita especificar outros valores, a magnitude da chuva limite ou precipitação crítica, porém, foi alterada para 2,5mm para todas as simulações. O dia foi considerado impróprio para a colheita de soja quando a chuva acumulada do dia anterior com o dia atual superar este valor. Tal altura pode, evidentemente, ser contestada tanto pelo valor nominal como pela intensidade e o momento de ocorrência. Todavia, a escassez de informações disponíveis não permitem uma estimativa com precisão maior. Ademais, é quase

certo que os cultivares de soja, dependendo da sua arquitetura, do atraso imposto e das condições microclimáticas, respondam de forma distinta para chuvas de mesma altura no que tange à aptidão para a colheita. Trata-se, pois, de um parâmetro aberto à pesquisa.

Para a determinação dos dias úteis observados considerando a precipitação crítica, o sistema utilizou uma série histórica de chuva diária acumulada cadastrada entre os dias 01/02 a 31/05, de 1954 a 2003, para a região de Ponta Grossa, fornecida pelo Instituto Tecnológico – SIMEPAR.

3.3.1.5.5 Período de simulação

Embora o programa permita simular intervalos menores, para todos os cenários, o período de simulação foi de 1954 a 2003, para o qual o sistema dispunha de uma série histórica de chuva diária acumulada já descrita no item anterior.

3.3.2 Dimensionamento

Para a seleção da colhedora utilizou-se o critério da renda líquida da operação. A conformação parabólica da função renda líquida por área de trabalho indica um custo maior da capacidade instalada antes do ponto de inflexão e, após, o custo maior das perdas por oportunidade.

Assim, nas simulações PROPRIEDADE, a capacidade ótima da máquina ou da frota de colhedoras será indicada pela área de trabalho que corresponde à máxima renda líquida média anual, segundo a composição de cultivares.

Nas simulações CENÁRIO, a capacidade ótima, expresso pelo número de máquinas, será indicada pela área e o prazo de conclusão do serviço que correspondem a máxima renda líquida média anual, segundo a composição de cultivares. O número de colhedoras, porém, foi determinado como sendo aquele que suprisse a demanda, no mínimo, de 90% dos anos da série simulada (45 anos) para um prazo de colheita, considerando uma área e a respectiva composição de cultivares.

4 RESULTADOS E DISCUSSÃO

4.1 Perdas por atraso na colheita

4.1.1 Desenvolvimento da lavoura

De acordo com a Escala Fenológica de FEHR e CAVINESS (1977), a Tabela 7 apresenta as datas de ocorrência dos principais estádios dos cultivares de soja e o respectivo tempo acumulado em dias após a semeadura para os subperíodos vegetativos entre a emergência e o início do florescimento (VE – VC ; VC – V1; V1 – R1), para os subperíodos reprodutivos do florescimento (R1 – R3), granação (R5 – R7) e maturação (R7 – R8) e, finalmente, colheita.

Para todos os cultivares, o ciclo de maturação não correspondeu àquele especificado no Registro Nacional de Cultivares, citado em EMBRAPA (1999 b, 2002). Para os cultivares precoces FT Cometa e M – Soy 5942, cujo ciclo deveria ser inferior a 115 dias, apresentou 128 e 150 dias, respectivamente. Para o cultivar semiprecoce BRS 133, que deveria variar entre 116 e 125 dias, apresentou 170 dias, enquanto que o FT Abyara, de ciclo médio, que deveria variar entre 126 e 137 dias, apresentou 166 dias. Tal discrepância, conhecida dos produtores da região, se deve ao fato de que o ciclo especificado no registro nacional de cultivares refere-se àquele observado nos ensaios de campo efetuados pela Embrapa Soja, em Londrina, Paraná. Para a região de Ponta Grossa, mais ao sul, os atributos climáticos diferem significativamente, o que prolonga o ciclo desta cultura termo e fotossensível.

A Figura 23 apresenta a evolução dos atributos climáticos de insolação, temperatura média e precipitação durante o período de desenvolvimento observados na região pelo Instituto Agrônomo do Paraná – IAPAR e Instituto Tecnológico - SIMEPAR. Exceto um curto período de estiagem que se seguiu à semeadura, todavia sem comprometer a emergência e o desenvolvimento das plântulas, não foram registrados eventos atípicos durante o ensaio. Foi observado, contudo, a infestação de *Brachiaria plantaginea* (Papuã) sobretudo nas

parcelas de FT Cometa e FT Abyara. Não foi constatado ataque de pragas ou doenças de final de ciclo a níveis comprometedores em quaisquer dos cultivares ensaiados.

Tabela 7 Data de ocorrência dos principais estádios fenológicos dos cultivares do ensaio.

Cultivar	Estádio	Data	DAS ¹	Cultivar	Estádio	Data	DAS ¹
FT – Cometa	S	27/10/01	-	FT – Abyara	S	27/10/01	-
	VE	02/11/01	6		VE	01/11/01	5
	VC	06/11/01	10		VC	05/11/01	9
	V1	11/11/01	15		V1	10/11/01	14
	R1	17/12/01	51		R1	07/01/02	72
	R3	28/12/01	62		R3	21/01/02	86
	R5	11/01/02	76		R5	07/02/02	103
	R7	09/02/02	105		R7	12/03/02	136
	R8	20/02/02	116		R8	27/03/02	151
	Colheita	04/03/02	128		Colheita	11/04/02	166
Cultivar	Estádio	Data	DAS ¹	Cultivar	Estádio	Data	DAS ¹
BRS 133	S	27/10/01	-	M – Soy 5942	S	27/11/01	-
	VE	02/11/01	6		VE	02/11/01	6
	VC	06/11/01	10		VC	06/11/01	10
	V1	11/11/01	15		V1	11/11/01	15
	R1	17/01/02	82		R1	19/12/01	53
	R3	31/01/02	96		R3	03/01/02	68
	R5	12/01/02	108		R5	21/01/02	86
	R7	14/03/02	138		R7	26/02/02	122
	R8	29/03/03	153		R8	12/03/02	136
	Colheita	15/04/02	170		Colheita	26/03/02	150

¹ DAS: Dias após a semeadura.

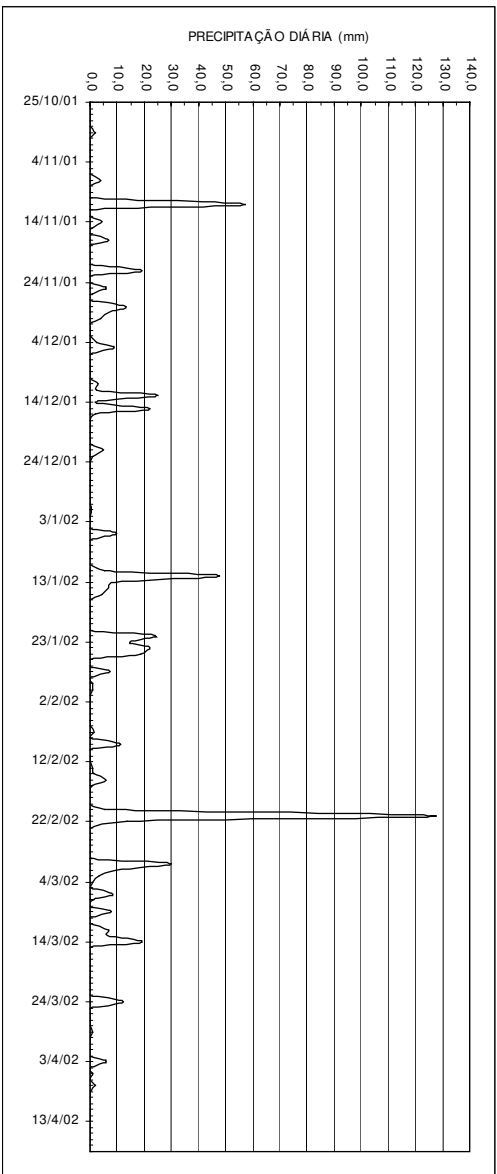
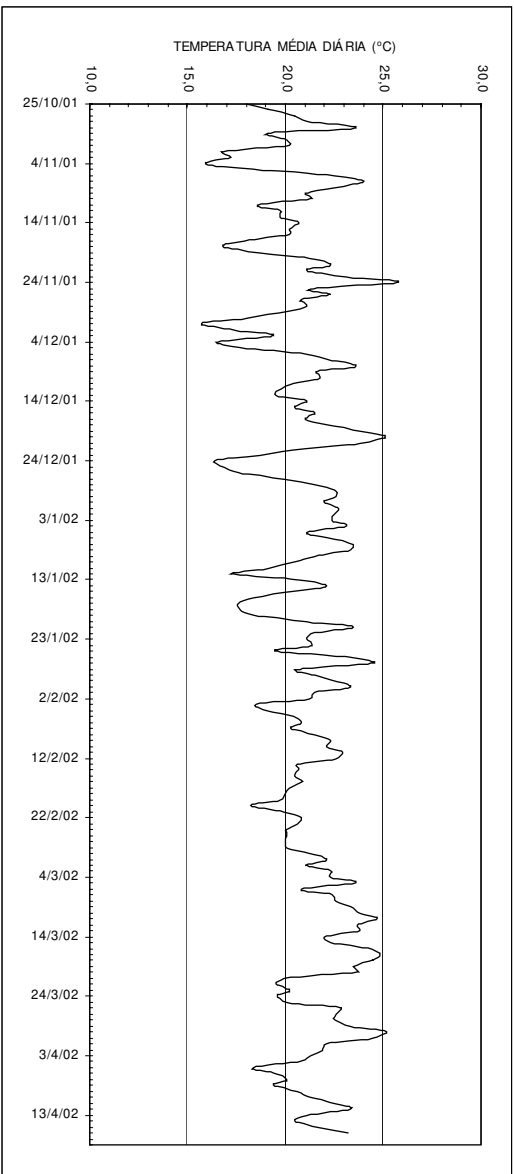
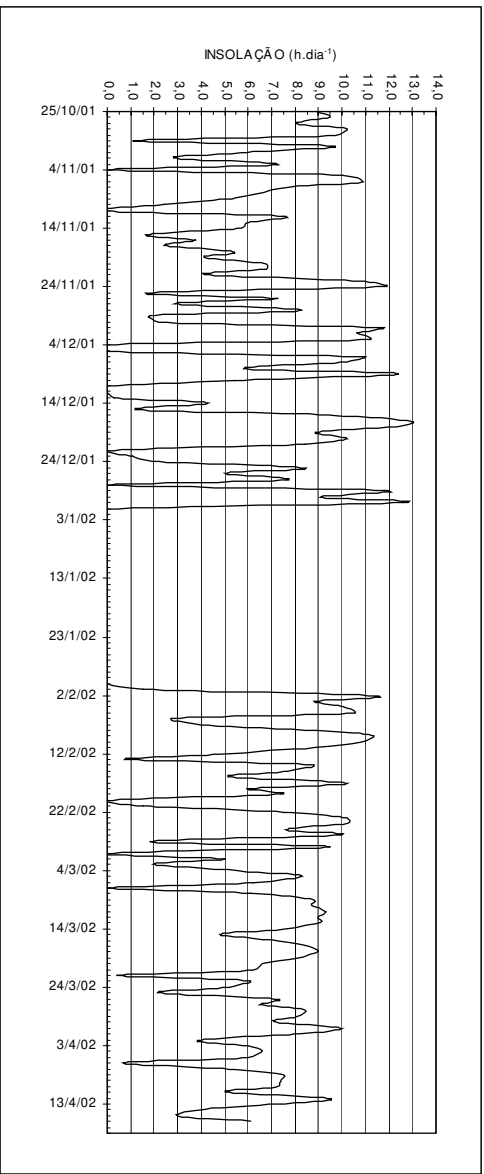


Figura 23 Atributos climáticos durante o período de desenvolvimento vegetativo dos cultivares no período de ensaio (Fonte: Instituto Agrônômico do Paraná – IAPAR; Instituto Tecnológico – SIMEPAR, 2003)

4.1.2 Datas de colheita e atrasos

O projeto previa a colheita das subparcelas em intervalos pré-fixados semanais porém, isso não foi possível devido aos eventos climáticos, sobretudo a distribuição irregular das chuvas associada à temperatura e umidade. Todavia, foi possível executar as dez colheitas pretendidas que produziram um atraso máximo de 42 dias no cultivar FT Cometa, 48 dias no cultivar M – Soy 5942 e 52 dias no FT Abyara e BRS 133, conforme a Tabela 8. Na Figura 24 encontra-se o desenvolvimento dos eventos climáticos de temperatura média, umidade relativa e precipitação durante o período de colheita observado na região pelo Instituto Agrônômico do Paraná – IAPAR e Instituto Tecnológico – SIMEPAR.

Tabela 8 Datas e atrasos na colheita das subparcelas.

Cultivar	Subparcela	Data	Atraso (dias)	Cultivar	Subparcela	Data	Atraso (dias)
FT Cometa	T0	04/03	0	FT Abyara	T0	11/04	0
	T1	07/03	3		T1	15/04	4
	T2	11/03	7		T2	18/04	7
	T3	14/03	10		T3	22/04	11
	T4	18/03	14		T4	25/04	14
	T5	26/03	22		T5	10/05	29
	T6	04/04	31		T6	13/05	32
	T7	08/04	35		T7	24/05	43
	T8	11/04	38		T8	27/05	46
	T9	15/04	42		T9	03/06	52
Cultivar	Subparcela	Data	Atraso (dias)	Cultivar	Subparcela	Data	Atraso (dias)
BRS 133	T0	15/04	0	M- Soy 5942	T0	26/03	0
	T1	18/04	3		T1	04/04	9
	T2	22/04	7		T2	08/04	13
	T3	25/04	10		T3	11/04	16
	T4	10/05	25		T4	15/04	20
	T5	13/05	28		T5	18/04	23
	T6	24/05	39		T6	22/04	27
	T7	27/05	42		T7	25/04	30
	T8	03/06	49		T8	10/05	45
	T9	06/06	52		T9	13/05	48

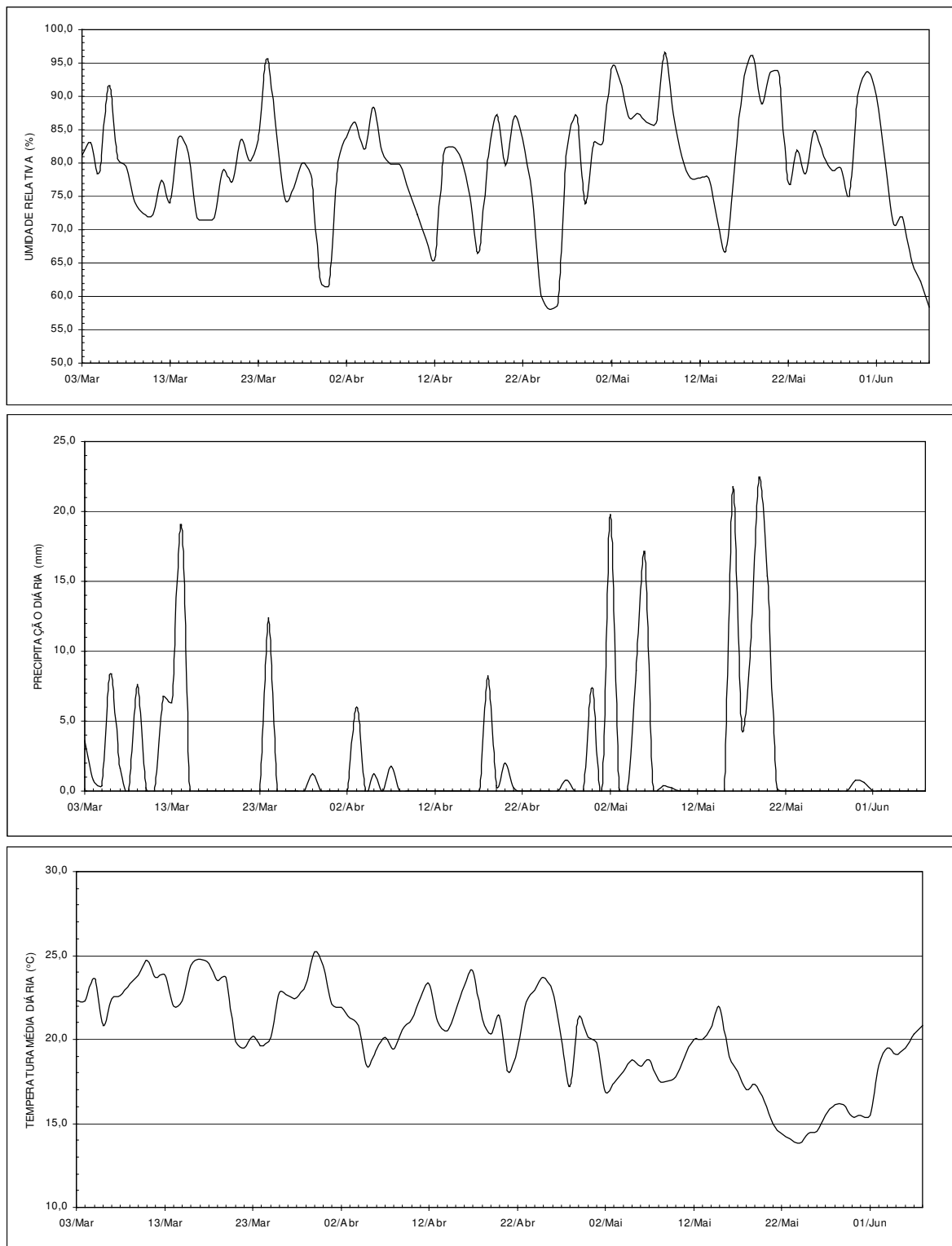


Figura 24 Atributos climáticos durante o período de colheita no ensaio (Fonte: Instituto Agrônômico do Paraná – IAPAR; Instituto Tecnológico – SIMEPAR, 2003).

4.1.3 Produtividade comercial e perdas

Os resultados de produtividade de soja comercial em função do atraso na colheita obtidos para os cultivares testados nas condições edafoclimáticas do local de ensaio na região de Ponta Grossa, encontram-se nas Figuras 25 a 28 (Anexo A – Tabela 1). Contatou-se que, embora não tenha sido possível proceder colheitas com atrasos regulares devido à interferência do clima em considerável número de subparcelas, os atrasos aplicados mostraram a redução da produtividade de soja comercial em todas os cultivares.

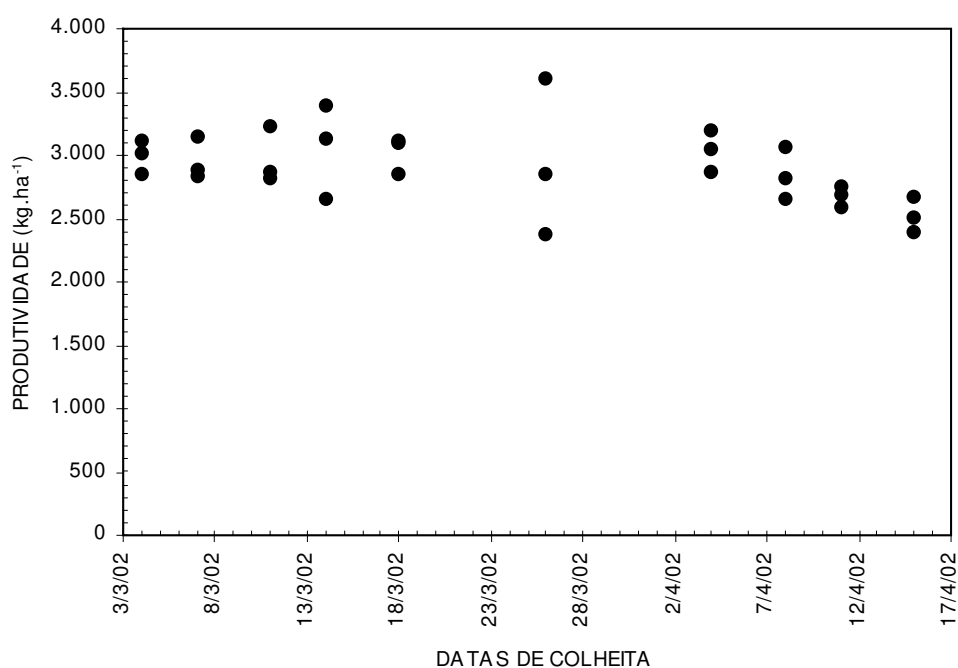


Figura 25 Tendência de produtividade de soja comercial do cultivar FT Cometa.

A análise de variância dos resultados (Anexo A – Tabela 2) mostrou que no ensaio não houve diferença estatística significativa entre as repetições (blocos). Porém, nos tratamentos (cultivares), subtratamentos (atrasos na colheita), bem como na interação entre ambos fatores, houve diferença significativa ao nível de 1%. A redução da produtividade de soja comercial depende, portanto, não só da variedade considerada, mas também do atraso aplicado.

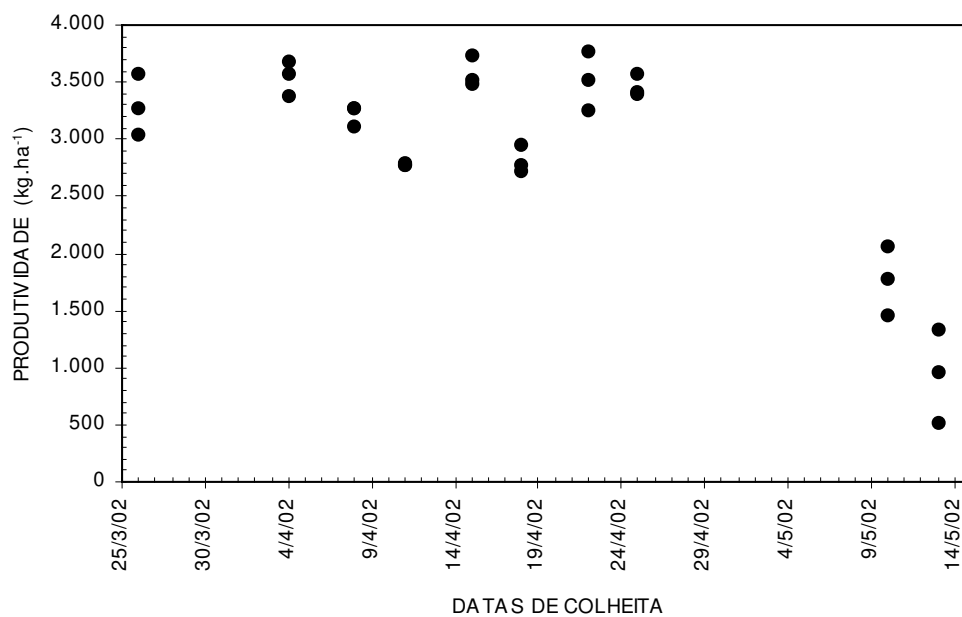


Figura 26 Tendência de produtividade de soja comercial do cultivar M - Soy 5942.

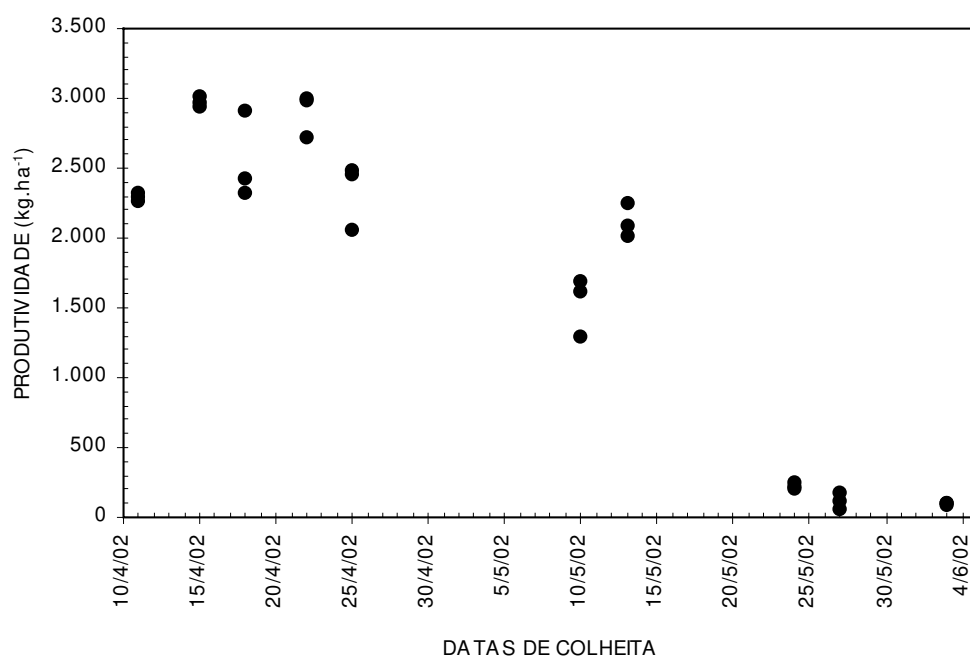


Figura 27 Tendência de produtividade de soja comercial do cultivar FT Abyara.

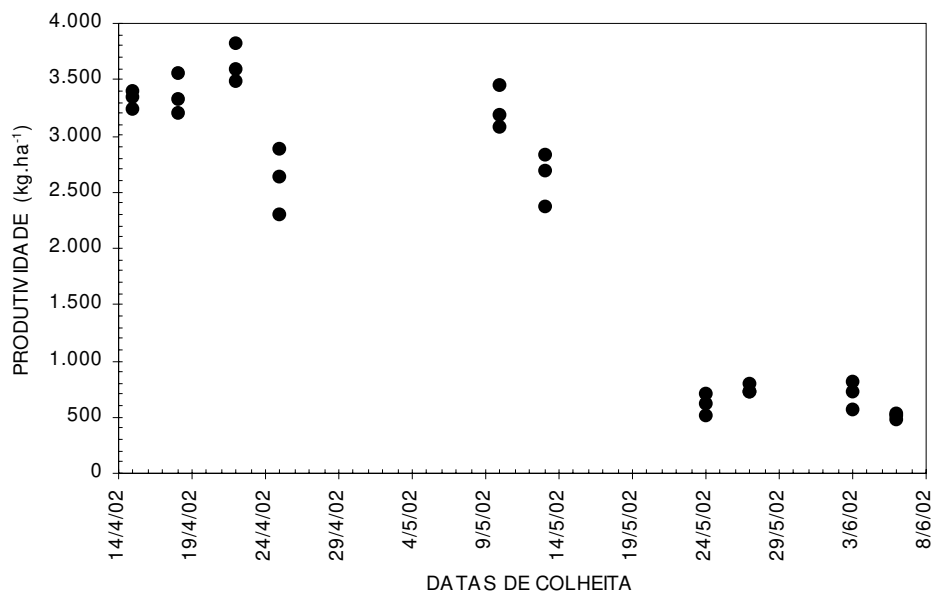


Figura 28 Tendência de produtividade de soja comercial do cultivar BRS 133.

A comparação das médias (Tabela 9) mostrou que a produtividade dos cultivares FT Cometa e M – Soy 5942, de ciclo precoce, não diferiram estatisticamente entre si ao nível de 1%. Porém houve diferenças significativas entre os cultivares BRS 133 e FT Abyara, esta apresentando a menor produtividade entre os cultivares testados.

A comparação das médias dos subtratamentos mostrou que a produtividade decresceu segundo o atraso (Tabela 9), muito embora os atrasos iniciais só foram estatisticamente diferentes quando agrupados em categorias. Isto decorreu porque os cultivares FT Cometa e M – Soy 5942 não mostraram reduções de produtividade significativas nos primeiros atrasos.

Analisando o isoladamente o comportamento de cada cultivar (Tabela 11 e Anexo A – Tabela 3), verificou-se que os subtratamentos (atrasos) aplicados na variedade FT Cometa não apresentaram diferenças estatisticamente significativas entre si ao nível de 1%, enquanto que para a M – Soy 5942 houve diferença. O ensaio, porém, apresentou coeficiente de variância elevado denotando algum erro experimental. Para as cultivares BRS 133 e FT Abyara os subtratamentos produziram resultados estatisticamente significativos.

Tabela 9 Comparativo das médias de produtividade de soja comercial dos tratamentos.

Tratamentos	Média (kg.ha ⁻¹)	
FT – Cometa	2.904	A
M – Soy 5942	2.891	A
BRS 133	2.136	B
FT – Abyara	1.716	C

Médias seguidas das mesmas letras maiúsculas nas linhas não diferem significativamente entre si pelo teste de Duncan ao nível de 1%.

Tabela 10 Comparativo das médias de produtividade de soja comercial nos subtratamentos.

Subtratamentos	Média (kg.ha ⁻¹)	
T1	3.209	A
T2	3.093	AB
T4	3.045	B
T0	2.976	BC
T3	2.840	C
T5	2.840	D
T6	2.323	E
T7	1.820	F
T8	1.315	G
T9	1.014	H

Médias seguidas das mesmas letras maiúsculas nas linhas não diferem significativamente entre si pelo teste de Duncan ao nível de 1%.

A tendência produtividade média dos cultivares em função do atraso na colheita, foi ajustada e representada através de funções quadráticas conforme as Figuras 29 a 32.

Com o modelo polinomial foi possível comparar as perdas de produtividade em função do atraso na colheita dos cultivares testados, até o limite de 45 dias (Figura 33).

Os cultivares precoces FT Cometa e M – Soy 5942 tiveram as menores perdas em relação ao respectivo rendimento comercial máximo. Após 15 dias de atraso, as perdas relativas do cultivar M – Soy 5942 superaram o FT Cometa e, a partir de aproximadamente 30 dias, a sua produtividade comercial também se tornou menor. Portanto, até 30 dias de atraso, o cultivar M – Soy 5942 apresentou o melhor desempenho em termos de rendimento potencial. Após este prazo, o cultivar FT Cometa revelou-se o melhor cultivar em razão da sua menor

tendência de perdas por atraso que lhe conferem maior produtividade, não obstante possuir menor rendimento potencial.

Tabela 11 Comparativo das médias de produtividade de soja comercial dos subtratamentos para cada cultivar.

FT Cometa			M – Soy 5942		
Subtratamento	Média (kg.ha ⁻¹)		Subtratamento	Média (kg.ha ⁻¹)	
T3	3.064	A	T4	3.582	A
T6	3.040	A	T1	3.547	AB
T4	3.024	A	T6	3.518	AB
T0	2.994	A	T7	3.460	AB
T2	2.971	A	T2	3.219	AB
T1	2.956	A	T5	2.815	AB
T5	2.947	A	T3	2.782	AB
T7	2.844	A	T0	2.290	ABC
T8	2.677	A	T8	1.764	BC
T9	2.525	A	T9	933	C

FT Abyara			BRS 133		
Subtratamento	Média (kg.ha ⁻¹)		Subtratamento	Média (kg.ha ⁻¹)	
T1	2.972	A	T2	3.361	A
T3	2.907	A	T1	3.363	A
T2	2.554	B	T0	3.322	A
T4	2.335	BC	T4	3.239	A
T0	2.294	BC	T5	2.631	B
T6	2.122	C	T3	2.606	B
T5	1.536	D	T7	748	C
T7	228	E	T8	702	C
T8	117	E	T6	613	C
T9	93	E	T9	506	C

Médias seguidas das mesmas letras maiúsculas nas linhas não diferem significativamente entre si pelo teste de Duncan ao nível de 1%.

Por outro lado, o cultivar semiprecoce BRS 133 e médio FT Abyara se comportaram de maneira semelhante nas perdas relativas, embora tenham apresentado produtividades diferentes no ensaio. O cultivar BRS 133 revelou-se mais interessante devido à sua produtividade potencial maior que o cultivar FT Abyara. Porém, em relação aos cultivares

precoces M – Soy 5942 e FT Cometa, a sua produtividade tornou-se inferior após 15 dias de atraso, em face da sua característica de maior tendência de perdas.

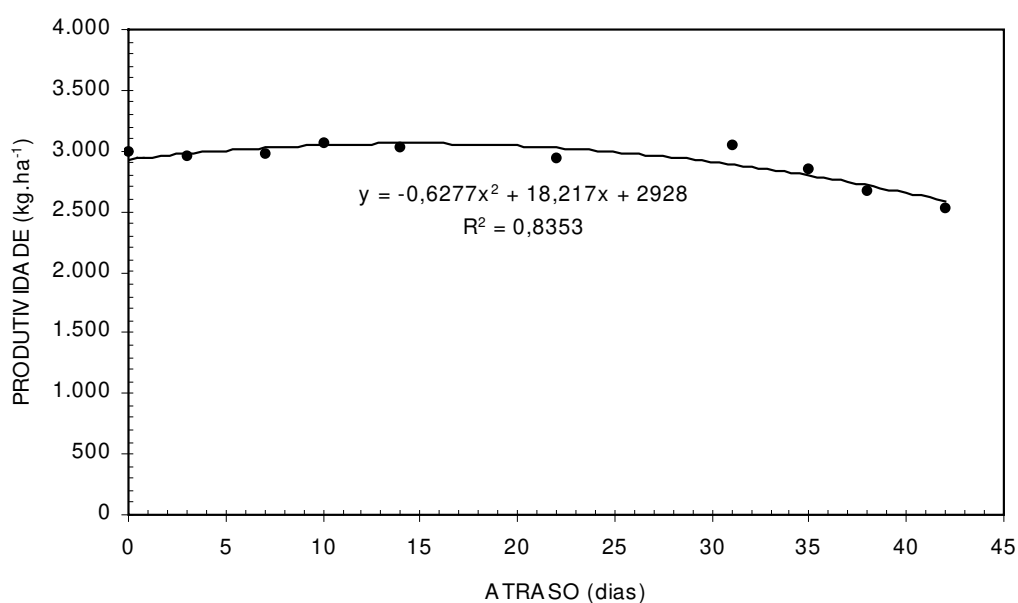


Figura 29 Produtividade média de soja comercial do cultivar FT Cometa.

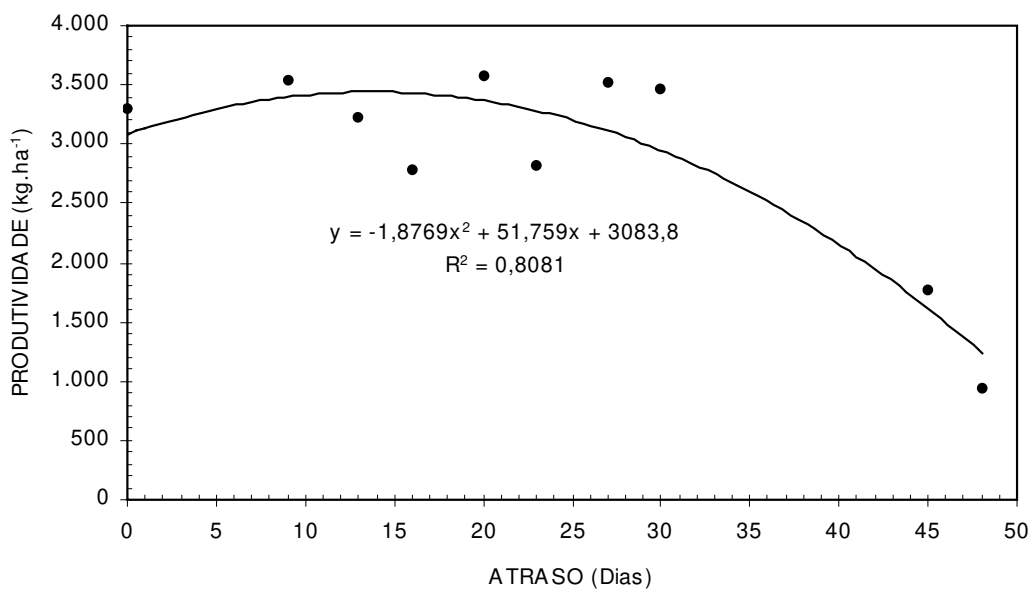


Figura 30 Produtividade média de soja comercial do cultivar M – Soy 5942.

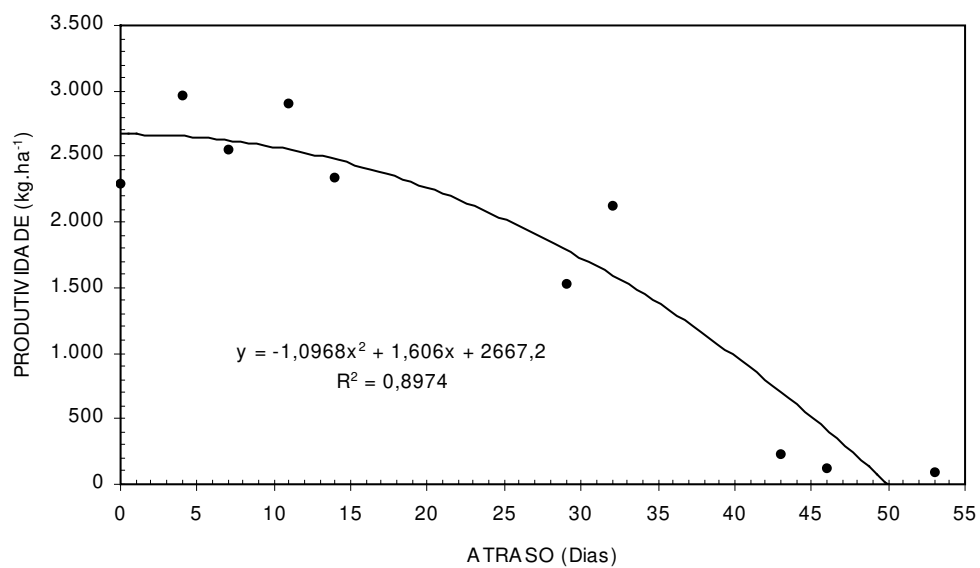


Figura 31 Produtividade média de soja comercial do cultivar FT Abyara.

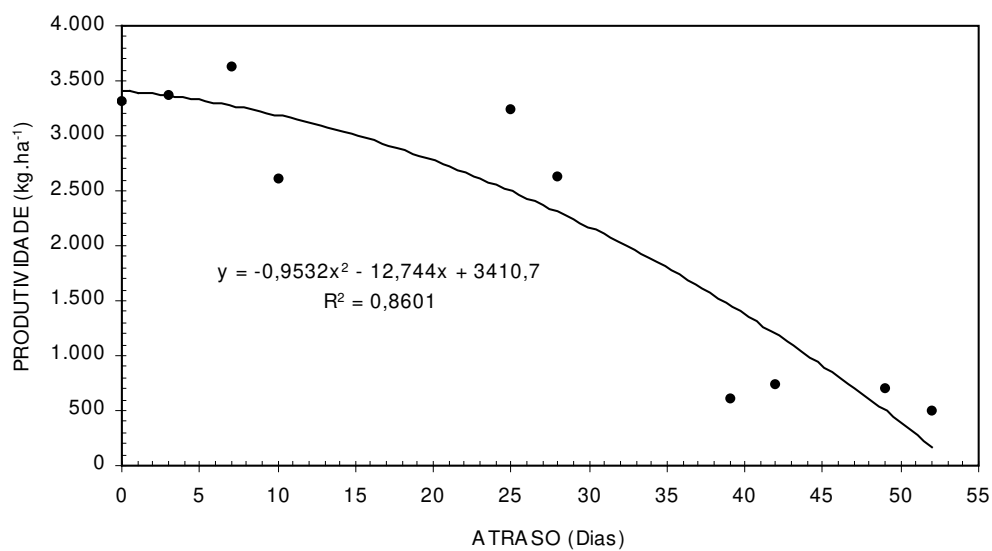


Figura 32 Produtividade média de soja comercial do cultivar BRS 133.

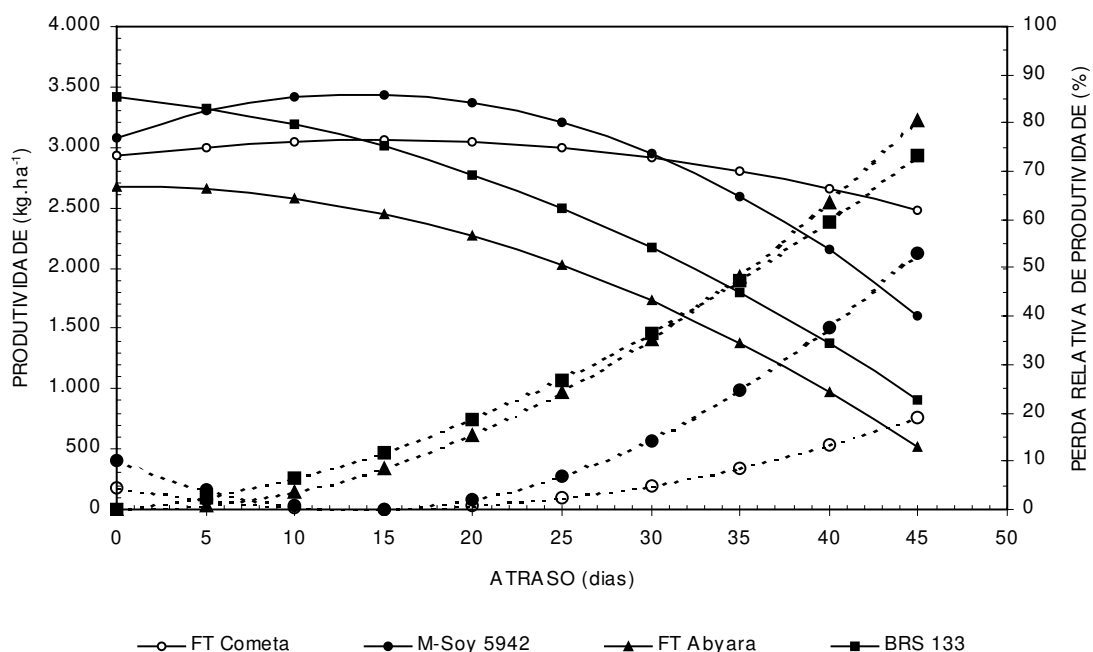


Figura 33 Comparativo de produtividade de soja comercial dos cultivares e perdas relativas respectivas.

Verifica-se, pois, que os cultivares apresentaram diferentes tendências de perdas por atraso na colheita, o que já havia sido observado por PHILBROOK (1989) e por SEDIYAMA et al. (1972, b) e LIN e SEVERO (1982), nos trabalhos desenvolvidos no Brasil. Todavia, a magnitude dessas perdas dependem do seu rendimento potencial indicando que a eficiência de colheita sob condições de atraso pode ser melhorada com o emprego de cultivares de alta produtividade, corroborando a afirmação BOEHLJE e EIDMAN (1984), para os quais o custo de oportunidade só é afetado pelo rendimento. Para ASAE (2000 b), além do rendimento, o custo de oportunidade é afetado também o valor comercial do produto.

Ressalta-se, como já foi afirmado, que tais observações resultam das condições edafoclimáticas específicas do ensaio conduzido apenas na safra 2001/02. Embora a análise de variância tenha revelado coeficientes de variância aceitáveis, somente uma série de ensaios nas mesmas condições poderia fornecer curvas de perdas mais confiáveis que as obtidas. Embora as funções de perdas e os coeficientes de oportunidade sempre resultaram de observações experimentais (EDWARD e BOEHLJE, 1980; OSKAN e EDWARDS, 1986; WITNEY, 1988;

ASAE, 2000 b; VEIGA, 2000), a determinação quantitativa das perdas por oportunidade em operações agrícolas é uma tarefa que consome tempo porque devem ser feitos testes múltiplos e medidas, muitas das quais sob condições onde o valor das perdas têm a finalidade de validar dados de perda de oportunidade. Por conseguinte, a quantidade de dados disponíveis em coeficientes de oportunidade é muito limitado (BURROWS e SIEMENS, 1974). Além disso, cada valor é devido às condições particulares de solo, tempo e condições de colheita sob os quais foi desenvolvido. Assim, tais valores só podem servir como estimativas (CHANCELLOR e CERVINKA, 1974) e não podem ser utilizados em outros locais (OSKAN, et al., 1990).

Finalmente, ao lado dos trabalhos desenvolvidos no Brasil por SEDIYAMA et al. (1972, a), LIN e SEVERO (1982) e VIEIRA et al. (1982), que avaliaram o efeito do atraso na colheita de soja mas não propuseram funções que correlacionassem o atraso com as perdas de rendimento ou qualidade do material colhido, os resultados obtidos neste trabalho podem ser utilizados no modelo computacional proposto para a seleção de colhedoras.

Finalmente, na Tabela 12 se encontram os coeficientes dos polinômios que expressam as perdas de produtividade comercial com o atraso, utilizados no modelo computacional.

Tabela 12 Coeficientes das funções de perdas por atraso dos cultivares testados.

Cultivares	kg.ha ⁻¹				sacos 60 kg.ha ⁻¹			
	a	b	c	C	a	b	c	C
FT Cometa	0,63	-18,216	131,49	2.928,0	0,0105	- 0,3036	2,1868	48,8000
M- Soy 5942	1,878	-51,756	353,79	3.083,8	0,0313	- 0,8626	5,9416	51,3967
FT Abyara	1,098	-1,59	0,0	2.667,2	0,0183	-0,0265	0,0	44,4560
BRS 133	0,954	12,732	0,0	3.410,7	0,0159	0,2122	0,0	56,8419

4.2 Seleção da colhedora

4.2.1 Simulações PROPRIEDADE.

As simulações de PROPRIEDADE determinaram a renda líquida média anual para cinco composições de cultivares e diferentes áreas considerando o emprego da frota da

propriedade objeto de estudo composta de 02 unidades de colhedora de 125,1 kW (plataforma de 4,8 m) e 01 unidade de colhedora de 165,6 kW (plataforma de 7,0 m), e também para cada modelo de máquina isoladamente, incluindo também o modelo de 132,5 kW (plataforma de 5,7 m) (Anexo B, Tabelas 1 – 8). Tais cenários, ressaltando-se, foram limitados pelas condições contornos já descritos na metodologia. Nas Figuras 33 a 37 estão plotados os resultados por composição. O comportamento parabólico da renda líquida reflete o custo maior da capacidade instalada antes do ponto de inflexão e, após, o custo maior das perdas.

A análise dos mesmos permite determinar a utilização da capacidade instalada da frota e de cada colhedora isoladamente, tomando como referência uma área trabalhada de 500 ha.

Para a composição I, FT Cometa + M – Soy 5942 (Figura 34), dois cultivares com menor tendência de perdas por atraso, a colhedora de 125,1 kW (plataforma de 4,8m) apresentou a maior renda líquida até 500 ha de lavoura. No estreito intervalo de 500 – 600 ha, a colhedora de 132,5 kW (plataforma de 5,7 m) produziu a maior renda, pouco superior à colhedora anterior. Entre 600 ha e 1.200 ha, a colhedora de 165,6 kW (plataforma de 7,0m) apresentou o melhor resultado, superior inclusive às demais máquinas. A frota, porém, só apresentou renda líquida superior a quaisquer das outras máquinas acima de 1.200 ha, atingindo o valor máximo em 1.900 ha de lavoura. Nesta composição de cultivares, utilização da frota para uma área de 500 ha de lavoura significou uma ociosidade de 73 % da capacidade em relação à área de renda líquida máxima.

Considerando a composição II, FT Abyara + BRS 133 (Figura 35), dois cultivares com maior tendência de perdas por atraso na colheita, a disposição das curvas é a mesma, porém a sua amplitude se altera. A colhedora de 125,1 kW apresentou a maior renda líquida para áreas de até 300 ha. A utilização da colhedora de 132,5 kW não seria economicamente interessante visto que entre 300 – 700 ha, a colhedora de 165,6 kW apresentou o maior rendimento. A frota, por outro lado, só produziu renda líquida superior a quaisquer dos modelos acima de 700 ha, atingindo o seu pico em 1.000 ha. Considerando a área de lavoura de 500 ha com esta composição de cultivares, a ociosidade da frota foi de 50%.

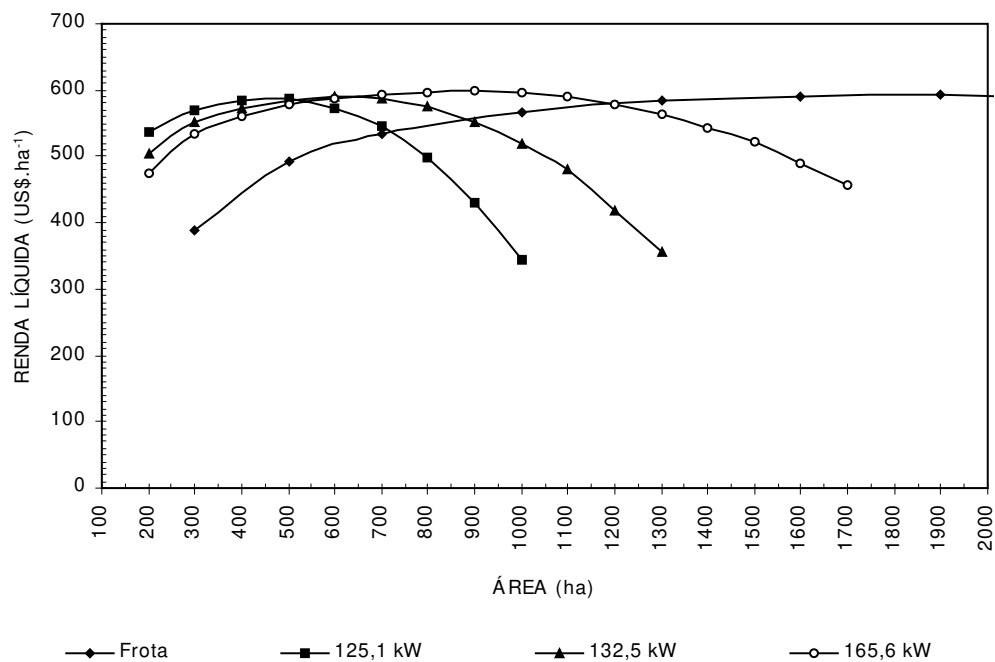


Figura 34 Renda líquida média anual da frota e dos demais modelos de colhedoras para a composição I (FT Cometa + M – Soy 5942).

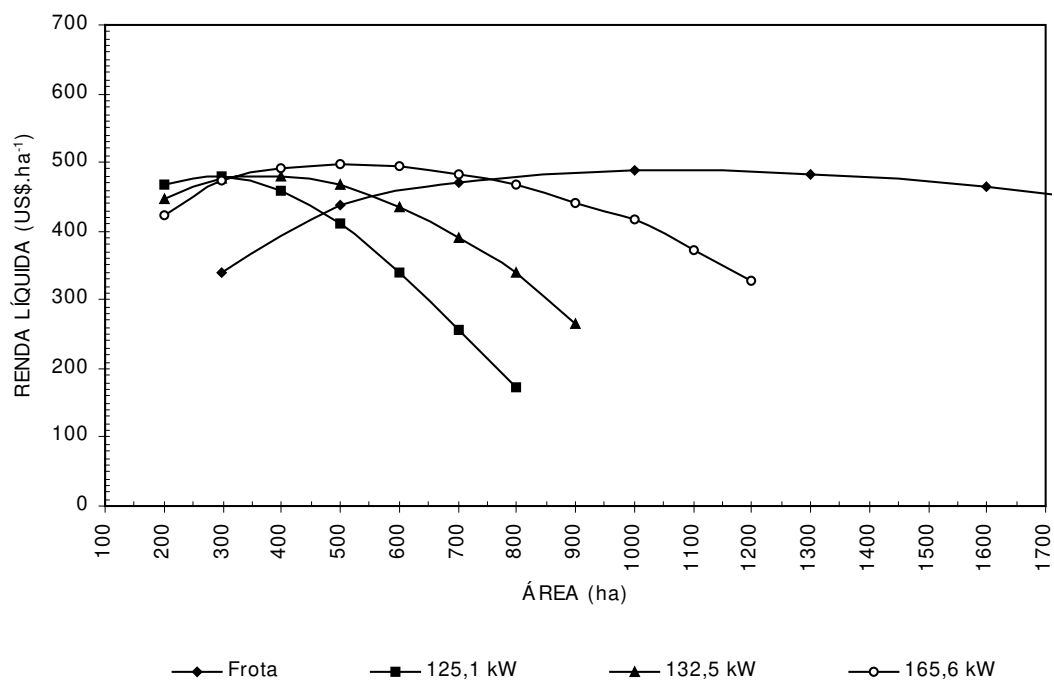


Figura 35 Renda líquida média anual da frota e dos demais modelos de colhedoras para a composição II (FT Abyara + BRS 133).

Na composição III, FT Cometa + FT Abyara (Figura 36), onde apenas o primeiro cultivar possui baixa tendência de perdas por atraso, o comportamento foi muito semelhante ao resultado da composição anterior, porém variando a amplitude de aplicação. A colhedora de 125,1 kW apresenta o maior renda líquida para áreas de até 400 ha. O uso da colhedora de 132,5 kW não seria economicamente interessante porque na faixa entre 400 – 1000 ha a colhedora de 165,6 kW apresentou a maior renda. A frota, por outro lado, só apresentou renda líquida superior às máquinas anteriores acima de 1.000 ha, atingindo o seu máximo em 1.600 ha. Para 500 ha de lavoura com esta composição de cultivares, a ociosidade da frota foi de 69%.

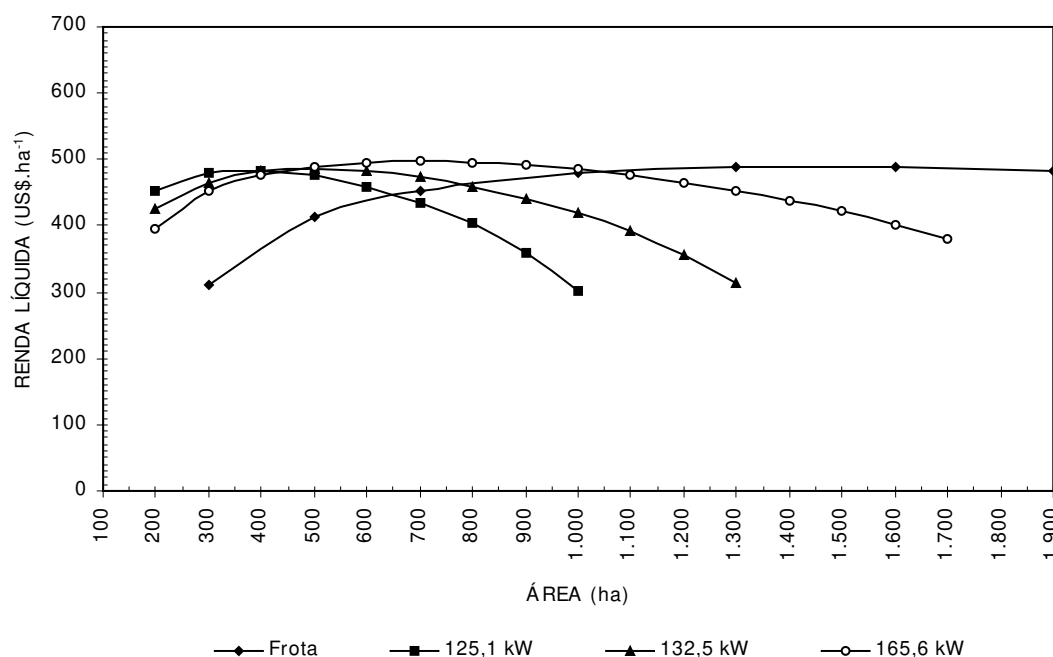


Figura 36 Renda líquida média anual da frota e dos demais modelos de colhedoras para a composição III (Cometa + FT Abyara)

Na composição IV, usando apenas FT Cometa (Figura 37), um cultivar com menor tendência de perdas por atraso na colheita, a colhedora de 125,1 kW apresentou maior renda líquida até 300 ha, apenas. A colhedora de 132,5 kW apresentou uma estreitíssima faixa de aplicação entre 300 – 350 ha. Entre 350 – 750 ha, a colhedora de 165,6 kW apresentou a maior

renda líquida. A frota, por outro lado, apresentou renda maior que as máquinas anteriores acima de 750 ha, atingindo o máximo em 1.000 ha. Para 500 ha de lavoura com FT Cometa, o uso da frota resultou numa ociosidade de 50 %.

Finalmente, na composição IV, usando apenas FT Abyara (Figura 38), um cultivar com alta tendência ao atraso, a colhedora de 125,1 kW revelou-se aplicável apenas uma faixa estreita até 250 ha. A colhedora de 132,5 kW não apresentou faixa de uso, visto que entre 250 – 550 ha a máquina de 165,6 kW apresentou maior renda líquida. A frota, por outro lado, somente apresentou aplicação acima de 550 ha atingindo a renda máxima em 700 ha de lavoura. Para uma área de 500 ha com esta composição, o uso da frota resultou numa ociosidade de 28 %.

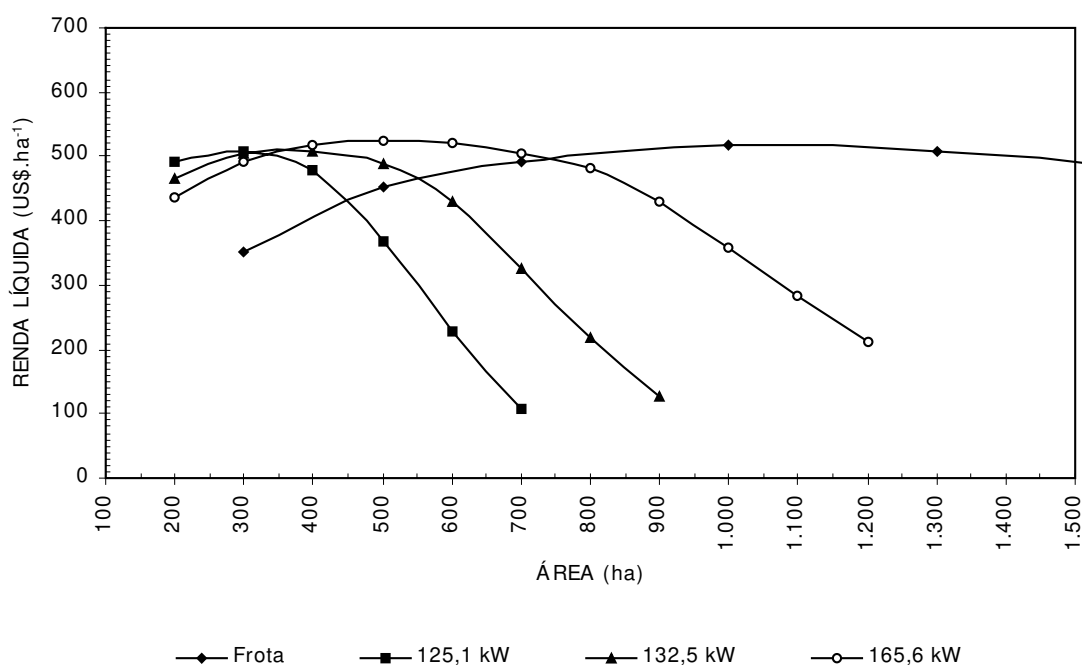


Figura 37 Renda líquida média anual da frota e dos demais modelos de colhedoras para a composição IV (FT Cometa).

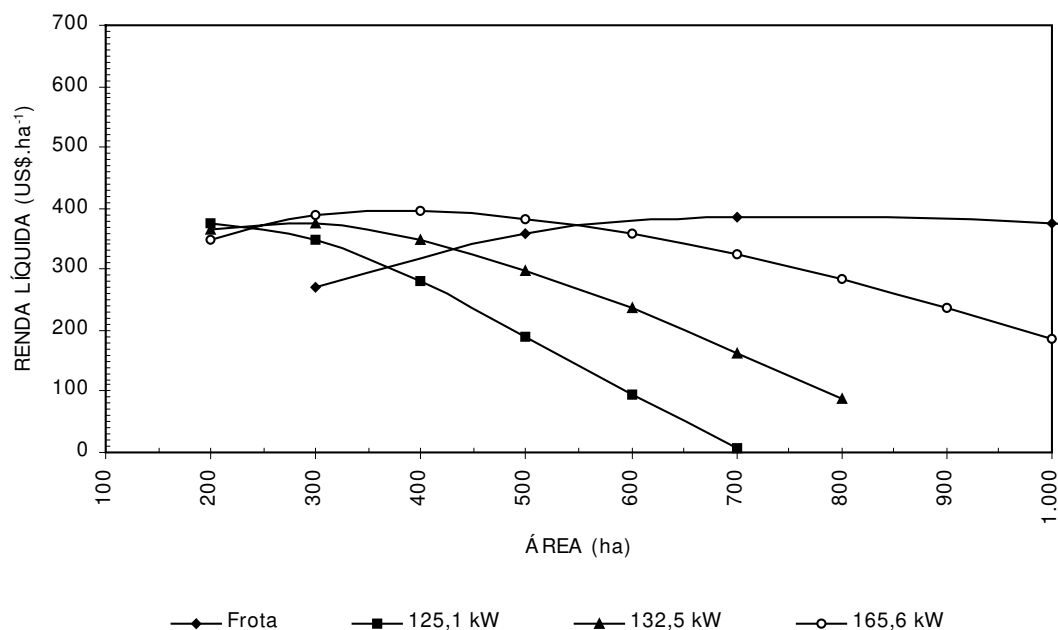


Figura 38 Renda líquida média anual da frota e dos demais modelos modelos de colhedoras para a Composição V (FT Abyara).

A Tabela 13 fornece um resumo dos limites de área para uso da frota e demais colhedoras para as composições de cultivares simuladas. A Tabela 14 fornece a média dos dias úteis de campo para a operação de colheita observados na série histórica de 1954 a 2003, na região de Ponta Grossa, segundo o critério utilizado nas simulações utilizando o modelo computacional.

Tabela 13 Limites de área para uso da frota e demais modelos de colhedoras segundo as composições de cultivares simuladas, em hectare.

Colhedoras	Composições				
	I	II	III	IV	V
125,1 kW	área ≤ 500	área ≤ 300	área ≤ 400	área ≤ 300	área ≤ 250
132,5 kW	500 < área ≤ 600	0	0	300 < área ≤ 350	0
165,6 kW	600 < área ≤ 1.200	300 < área ≤ 750	400 < área ≤ 1.000	350 < área ≤ 750	250 < área ≤ 550
Frota	1.200 < área ≤ 1.900	750 < área ≤ 1.000	1.000 < área ≤ 1.600	750 < área ≤ 1.000	550 < área ≤ 700

Tabela 14 Média de dias úteis de campo para colheita observados na série histórica de 1954 a 2003, região de Ponta Grossa (Fonte: Autor).

Precipitação Crítica (mm)	Fevereiro		Março		Abril		Maio	
	Dias úteis	S	Dias úteis	S	Dias úteis	S	Dias úteis	S
0,5	10,16	4,98	14,32	4,52	18,38	4,90	20,30	5,27
1,0	10,96	5,11	15,38	4,45	19,16	4,27	20,98	5,23
1,5	11,68	5,22	16,26	4,36	19,64	4,34	21,46	5,10
2,0	12,30	5,08	16,84	4,40	20,26	4,26	21,80	5,01
2,5	12,90	4,90	17,44	4,44	20,60	4,29	22,06	4,95
3,0	13,24	4,96	17,96	4,34	21,06	4,29	22,30	4,92
3,5	13,72	4,85	18,68	4,09	21,32	4,08	22,82	4,62
4,0	14,40	4,60	19,08	4,11	21,78	4,00	23,12	4,71
4,5	14,78	4,48	19,30	4,09	22,14	3,78	23,34	4,77
5,0	15,22	4,30	21,60	3,88	22,38	3,71	23,68	4,77

S = Desvio padrão.

Na composição I (Tabela 13) a capacidade da frota ou dos modelos considerados permitiria trabalhar áreas maiores porque ambos cultivares apresentaram menor tendência de perdas por atraso e produtividade potencial maior. Ademais, o ciclo de ambas permite o escalonamento da operação com a colheita do cultivar FT Cometa no mês de março e o cultivar M-Soy 5942 em abril. Nas simulações, o mês de março apresentou uma quantidade de dias de campo úteis para colheita observados menor que abril e maio (Tabela 14). Assim, apenas uma colhedora de 125,1 kW com plataforma de 4,8 m seria capaz de trabalhar 500 ha. Na composição III, para a colhedora de 125,1 kW, a área trabalhada e a renda são menores porque o cultivar FT Abyara, embora colhido no mês de abril, com disponibilidade maior de dias úteis de campo que o mês de março, possui maior tendência de perdas por atraso e uma produtividade potencial menor. Neste caso, seria economicamente mais interessante o uso de uma colhedora de 165,6 kW com plataforma de 7,0 m, embora operando com ociosidade. Esta colhedora também produziria maior renda líquida na composição II onde os cultivares, embora colhidos no mês de abril e maio, que possuem maior disponibilidade de dias úteis, apresentam maior tendência de perdas por atraso, além de um rendimento potencial menor que as precoces. Por outro lado, quando se trata de uma área composta de apenas um cultivar, no caso da composição IV (FT Cometa), a colhedora de 125,1 kW poderia colher no máximo 300 ha. A capacidade desta máquina se revelou limitada porque não havendo escalonamento da

colheita, esta se concentra toda no mês de março cuja disponibilidade de dias úteis de campo é menor. Por conseguinte, as perdas por atraso seriam elevadas. Para a composição V (FT Abyara), sucede o mesmo com a colhedora de 125,1 kW, sendo capaz de trabalhar apenas áreas menores que 250 ha. Embora colhido no mês de abril, o cultivar apresentou menor produtividade potencial e alta tendência de perdas por atraso. Neste caso, a colhedora de 165,6 kW apresentaria melhor resultado econômico.

A frota da área de referência, por outro lado, apresentou ociosidade em quaisquer das composições, particularmente na I, considerando uma demanda de 500 ha de lavoura de soja, admitindo que o período de colheita, sobretudo o início, quando se trata de cultivares precoces, não interfira no término da colheita de milho.

Esta constatação pode ser explicada em PARSONS et al. (1981). Os pesquisadores demonstraram que os agricultores investem em equipamentos em número e tamanho como um seguro para minimizar o tempo consumido em reparos e diluir os riscos de impontualidade. WHITSON et al. (1981), associando a impontualidade aos riscos climáticos, recomendou uma composição de espécies que, no caso estudado, pode ser traduzido como uma composição de cultivares de forma a estender o período de semeadura e colheita através do escalonamento. Os resultados das simulações mostraram que tal prática de fato produziu as maiores rendas com uma demanda menor de máquinas na medida em que foi possível aumentar a área trabalhada quando os cultivares não coincidiram as épocas de colheita.

4.2.2 Simulações CENÁRIO

Nas Tabelas 15 a 17 encontram-se os resultados das simulações CENÁRIO, considerando modelos de 125,1 kW, com plataforma de 4,8m; 132,5 kW, com plataforma de 5,7 m e 165,6 kW, com plataforma de 7,0m; prazos calendário de colheita variando de 10 a 35 dias, com intervalos de 5 dias, e áreas variando de 200 a 1.000 ha, para cinco composições de cultivares (ANEXO C, Tabelas 1 a 3). Não obstante os prazos, os demais limites de contorno foram os mesmos adotados nas simulações PROPRIEDADE.

O número de colhedoras (frota) determinado para cada cenário foi tomado como sendo aquele que supre a demanda, no mínimo, de 90% dos anos da série histórica simulada (45

anos). Prazos inferiores a 10 dias, embora possíveis de serem simulados no modelo, não foram considerados porque na série ocorreram anos em que não foi possível colher neste prazo quando a composição envolve o cultivar precoce FT Cometa, cuja colheita se inicia nos primeiros dias de março. Sucede que este mês é particularmente chuvoso na região de Ponta Grossa. Também não foram considerados prazos superiores a 30 dias para a composição II, FT Abyara + BRS 133, porque a data de conclusão da colheita para o cultivar BRS 133 ultrapassa a data limite de simulação fixado em 31/05, conforme especificado na metodologia.

Os resultados mostraram que para todas as composições a demanda de colhedoras aumentou na razão inversa do prazo de colheita para qualquer tamanho área. Prazos menores demandaram número maior de máquinas. A renda líquida, porém, teve um comportamento parabólico como já foi observado nas simulações PROPRIEDADE. Embora as perdas devidas a impontualidade foram pequenas ou nulas para prazos de colheita pequenos, os custos de propriedade da frota numerosa elevados reduzindo, por conseguinte, a renda líquida. A isso, soma-se o fato de que, no caso de composições de cultivares colhidos em épocas diferentes todavia com prazo calendário definido, a frota é dimensionada pela maior demanda requerida nos períodos com menor disponibilidade de dias úteis de campo. Dessa forma, o fator de utilização da frota na colheita de um cultivar em março, por exemplo, poderá ser elevado. Porém, para um talhão colhido em abril ou maio, o fator de utilização certamente será baixo pela maior disponibilidade de dias úteis de campo nestes meses. Por outro lado, uma vez maximizada a renda líquida para um determinado tamanho de frota, atrasos maiores reduzem a quantidade de máquinas mas também a renda, tendendo a ser negativa por conta do aumento das perdas.

Os resultados também mostram que a demanda de colhedoras foi maior e a renda líquida menor quando a composição foi de apenas um cultivar. Isso ocorreu porque não houve o efeito positivo do escalonamento, concentrando-se a colheita num prazo relativamente curto para minimizar as perdas. Para o cultivar FT Cometa (IV), por exemplo, a demanda de colhedoras foi elevada porque a sua colheita ocorre em março, um mês chuvoso, com baixa disponibilidade de dias úteis de campo. Por outro lado, quando a área foi composta de dois cultivares, mesmo envolvendo FT Cometa, a demanda diminuiu pelo escalonamento. Quando comparado com o cultivar FT Abyara (V), embora a demanda fosse a mesma, a renda líquida respectiva diminuiu por conta da sua maior tendência de perdas por atraso.

Concluiu-se pois, que o número de colhedoras e a renda líquida depende não só da composição de cultivares, mas também do prazo de colheita.

Analizando especificamente os resultados das simulações para uma área de 500 ha da composição I praticada na propriedade referência, o modelo de 125,1 kW maximizaria a receita líquida média anual com 2 unidades concluindo o serviço num prazo calendário de 30 dias (US\$ 562,89. ha⁻¹). No mesmo prazo, todavia, a frota de 2 unidades poderia colher 800 ha com um ligeiro incremento da receita líquida (US\$ 564,85. ha⁻¹). Aumentando o prazo para 35 dias, 1 unidade seria capaz de colher 500 ha, porém com perda significativa de receita (US\$ 550,68 . ha⁻¹). Por outro lado, 1 unidade produziria uma renda líquida equivalente à frota de 2 unidades se a área fosse reduzida para 400 ha e colhida no prazo de 30 dias (US\$ 563,05.ha⁻¹).

O modelo de 132,5 kW, por sua vez, produziria renda líquida maior que o primeiro colhendo 500 ha com 2 unidades num prazo calendário de 25 dias (US\$ 571,24. ha⁻¹) ou ainda em 30 dias (US\$ 564,14. ha⁻¹). A mesma frota, todavia, seria capaz de colher no mesmo prazo de 30 dias uma área de 700 ha produzindo renda líquida equivalente (US\$ 564,98 . ha⁻¹). A redução para 1 unidade, porém, resultaria em perdas expressivas na renda líquida devido ao aumento do prazo ou redução da área de colheita.

Finalmente, o modelo de 165,6 kW maximizaria a renda líquida colhendo 500 ha com 2 unidades num prazo calendário de 20 dias (US\$ 571,83 . ha⁻¹). Aumentando o prazo para 25 dias, apenas 1 unidade seria necessária, produzindo uma receita líquida ligeiramente menor (US\$ 569,24 . ha⁻¹). A frota de 2 unidades, todavia, poderia maximizar a renda líquida se a área fosse de 800 ha colhidos em 25 dias (US\$ 581,93 . ha⁻¹). Por lado, 1 unidade maximizaria a renda líquida colhendo 800 ha em 30 dias calendário (US\$ 576,66 . ha⁻¹).

Tabela 15 Demanda de colhedoras de 125,1 kW e renda líquida média anual para diferentes combinações de cultivares, áreas e prazos de colheita.

Área (ha)	Prazo Calendário (dias)											
	10		15		20		25		30		35	
	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.há ⁻¹	n	US\$.ha ⁻¹
Composição I FT Cometa + M – Soy 5942												
200	3	456,03	2	507,11	1	536,84	1	534,47	1	522,34	1	499,76
300	4	467,43	3	519,57	2	552,98	2	557,87	1	550,08	1	528,84
400	5	475,87	4	520,44	2	552,98	2	565,98	1	563,05	1	543,39

Tabela 15 (Continuação)

Área (ha)	Prazo Calendário (dias)											
	10		15		20		25		30		35	
	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹
Composição I FT Cometa + M – Soy 5942												
500	6	484,54	4	528,81	3	555,81	2	561,48	2	562,89	1	550,68
600	7	487,91	5	533,78	3	561,31	3	562,66	2	560,35	2	549,48
700	8	488,25	6	533,79	4	562,67	3	569,17	2	561,62	2	549,14
800	9	490,77	7	535,11	4	565,93	3	571,81	2	564,85	2	546,58
900	10	497,14	7	540,10	4	566,87	4	574,25	3	568,16	2	549,05
1000	11	494,29	8	538,41	5	565,12	4	572,26	3	569,01	2	551,76
Composição II FT Abyara + BRS 133												
200	3	413,95	2	439,74	1	419,52	1	367,06	1	299,57	-	-
300	4	418,62	2	454,62	2	441,64	2	393,43	2	324,23	-	-
400	5	429,86	3	452,23	2	441,97	2	399,44	2	332,95	-	-
500	6	439,43	3	458,64	2	447,84	2	397,28	2	330,98	-	-
600	7	439,85	4	464,11	3	447,6	3	397,62	3	329,03	-	-
700	8	441,75	4	465,48	3	454,07	3	402,48	3	334,86	-	-
800	9	445,36	5	466,05	4	453,58	3	405,26	3	337,44	-	-
900	10	449,34	5	470,45	4	456,76	4	407,41	4	339,85	-	-
1000	11	447,58	6	468,27	4	457,16	4	406,61	4	340,34	-	-
Composição III FT Cometa + FT Abyara												
200	3	355,78	2	411,42	1	430,99	1	422,58	1	401,55	1	370,67
300	4	378,61	3	422,66	2	447,11	2	443,60	1	428,08	1	399,76
400	5	382,84	4	425,61	2	446,22	2	452,30	2	438,65	1	413,52
500	6	358,59	4	434,51	3	452,09	2	448,88	2	442,09	2	420,88
600	7	395,48	5	438,07	3	455,44	3	450,18	2	439,56	2	420,41
700	8	398,06	6	439,60	4	457,82	3	454,71	2	439,27	2	421,11
800	9	400,35	7	440,28	4	459,62	4	457,68	3	441,80	2	417,51
900	10	404,71	7	444,78	4	462,19	4	461,57	3	445,76	2	419,97
1000	11	402,35	8	443,76	5	460,68	4	458,94	3	447,85	3	422,32
Composição IV FT Cometa												
200	5	282,86	4	368,74	2	438,72	2	469,45	1	482,22	1	477,38
300	7	306,19	5	391,65	3	488,10	3	471,22	2	487,60	2	495,24
400	9	316,96	7	396,94	4	459,00	3	479,16	2	488,45	2	488,99
500	11	319,12	8	403,64	5	461,31	4	484,67	3	496,80	2	494,48

Tabela 15 (Continuação)

Área (ha)	Prazo Calendário (dias)											
	10		15		20		25		30		35	
	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹
Composição IV FT Cometa												
600	13	321,75	10	405,16	6	464,61	5	488,34	3	500,01	3	501,75
700	15	328,25	11	410,28	7	466,96	5	491,96	4	500,77	3	505,40
800	17	329,08	13	410,16	8	469,17	6	490,69	4	503,56	3	504,57
900	19	330,52	14	413,98	8	470,10	7	495,61	5	504,54	4	506,70
1000	21	333,12	16	414,56	9	471,91	7	494,23	5	505,69	4	506,61
Composição V FT Abyara												
200	3	276,78	2	323,67	2	339,09	1	334,12	1	300,95	1	257,04
300	4	239,22	3	341,90	2	344,95	2	341,85	2	314,68	1	282,67
400	5	306,73	4	349,27	3	354,10	2	340,32	2	312,58	2	283,90
500	7	311,32	4	355,10	3	362,44	3	349,40	2	318,42	2	281,00
600	8	308,49	5	356,04	4	362,08	3	354,26	3	322,96	2	285,04
700	9	318,57	6	361,26	4	366,88	4	354,17	3	328,22	3	289,48
800	10	315,54	7	360,75	5	368,72	4	355,88	4	325,93	3	292,8
900	12	319,06	7	365,47	6	370,54	5	358,40	4	328,11	3	293,00
1000	13	324,00	8	364,64	6	369,52	5	357,57	4	331,27	4	293,15

Tabela 16 Demanda de colhedoras de 132,5 kW e renda líquida média anual para diferentes combinações de cultivares, áreas e prazos de colheita.

Área (ha)	Prazo Calendário (dias)											
	10		15		20		25		30		35	
	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹
Composição I FT Cometa + M – Soy 5942												
200	2	457,45	1	500,90	1	513,79	1	510,18	1	496,25	1	473,65
300	3	462,96	2	511,82	1	545,76	1	544,63	1	532,41	1	513,10
400	4	471,64	3	523,06	2	555,91	2	560,67	1	552,08	1	531,64
500	4	491,04	3	527,93	2	557,32	2	571,24	2	564,14	1	542,77
600	5	485,84	4	528,87	3	554,34	2	565,78	2	561,50	1	548,62
700	6	483,48	5	520,2	3	559,57	2	563,86	2	564,98	2	553,46
800	7	491,76	5	537,02	3	564,08	3	565,36	2	562,86	2	551,79
900	8	487,16	6	533,54	4	562,92	3	569,67	2	562,26	2	549,95
1000	8	499,1	6	540,97	4	568,53	3	574,51	2	565,56	2	547,51

Tabela 16 (Continuação)

Área (ha)	Prazo Calendário (dias)											
	10		15		20		25		30		35	
	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹
Composição II FT Abyara + BRS 133												
200	2	412,06	1	423,76	1	396,41	1	344,52	1	275,29	-	-
300	3	418,26	2	444,80	1	433,10	1	380,52	1	309,81	-	-
400	4	422,50	2	457,63	2	444,44	2	396,17	2	327,06	-	-
500	4	441,76	2	457,85	2	450,3	2	401,81	2	337,40	-	-
600	5	439,14	3	454,88	2	444,06	2	395,41	2	329,44	-	-
700	6	440,59	3	461,44	2	448,95	2	398,2	2	331,78	-	-
800	7	443,40	4	466,94	3	450,28	3	400,28	2	331,76	-	-
900	8	440,96	4	465,53	3	454,44	3	402,94	3	335,41	-	-
1000	8	451,99	4	470,44	3	457,77	3	407,42	3	340,22	-	-
Composição III FT Cometa + FT Abyara												
200	2	371,08	1	408,75	1	411,52	1	398,31	1	375,47	1	344,57
300	3	372,66	2	418,10	2	438,83	1	431,20	1	411,62	1	384,02
400	4	382,90	3	426,25	2	450,14	2	446,46	1	430,87	1	402,56
500	4	399,46	3	432,98	2	450,36	2	456,56	2	440,54	1	412,76
600	5	397,11	4	435,15	3	448,96	2	453,13	2	440,72	2	418,76
700	6	394,07	5	435,39	3	453,95	3	449,32	2	444,87	2	423,73
800	7	399,48	5	441,38	3	458,31	3	452,91	2	442,09	2	422,72
900	8	396,85	6	439,31	4	458,05	3	455,19	2	439,91	2	421,93
1000	8	408,94	6	446,96	4	462,51	3	460,77	3	442,89	2	419,40
Composição IV FT Cometa												
200	4	277,49	3	370,61	2	434,38	2	455,94	1	460,30	1	452,46
300	5	299,17	4	375,03	3	436,18	2	469,56	2	480,19	1	487,11
400	7	311,65	5	396,01	3	451,74	3	474,53	2	490,59	2	497,96
500	8	326,96	6	408,03	4	459,85	3	481,92	2	490,10	2	491,20
600	10	318,33	8	402,18	5	459,86	4	486,13	3	492,11	2	493,67
700	11	328,31	9	409,88	5	464,49	4	488,45	3	502,19	2	500,09
800	13	326,96	10	409,31	6	467,97	5	491,35	3	502,78	3	504,35
900	15	324,88	11	408,35	7	466,03	5	491,54	4	500,65	3	505,59
1000	16	332,15	12	415,9	7	471,90	6	494,01	4	505,45	3	505,63

Tabela 16 (Continuação)

Área (ha)	Prazo Calendário (dias)											
	10		15		20		25		30		35	
	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹
Composição V FT Abyara												
200	2	257,27	2	332,83	1	337,83	1	314,52	1	276,59	1	232,67
300	3	287,32	2	324,31	2	343,29	2	340,14	1	312,35	1	270,06
400	4	297,71	3	345,54	2	348,29	2	344,80	1	317,44	1	285,26
500	5	312,27	3	360,14	3	357,79	2	341,98	2	315,80	2	285,05
600	6	306,24	4	352,11	3	361,06	3	347,09	2	317,01	2	280,99
700	7	318,35	5	359,60	3	368,68	3	354,08	3	320,57	2	285,42
800	8	312,74	5	359,44	4	365,12	3	356,99	3	325,57	2	287,58
900	9	316,59	6	360,35	4	366,46	4	354,08	3	328,42	3	289,80
1000	10	320,13	6	367,55	5	370,34	4	357,33	3	329,30	3	293,43

Tabela 17 Demanda de colhedoras de 165,6 kW e renda líquida média anual para diferentes combinações de cultivares, áreas e prazos de colheita.

Área (ha)	Prazo Calendário (dias)											
	10		15		20		25		30		35	
	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹	n	US\$,ha ⁻¹
Composição I FT Cometa + M – Soy 5942												
200	2	456,85	1	481,56	1	493,43	1	487,17	1	473,02	1	450,24
300	2	493,33	2	528,73	1	538,88	1	534,65	1	520,66	1	498,05
400	3	495,74	2	536,78	1	560,19	1	556,98	1	544,48	1	521,96
500	3	510,98	2	547,25	2	571,83	1	569,24	1	556,50	1	536,30
600	4	504,84	3	546,68	2	572,99	2	575,51	1	566,41	1	544,91
700	4	512,79	3	544,65	2	569,74	2	580,80	1	572,67	1	551,06
800	5	511,57	4	547,38	2	572,99	2	581,93	1	576,66	1	556,38
900	5	516,36	4	551,39	3	571,72	2	579,64	2	573,42	1	558,63
1000	6	518,47	4	554,04	3	575,24	2	578,36	2	576,53	1	562,13
Composição II FT Abyara + BRS 133												
200	2	403,40	1	400,86	1	375,91	1	321,55	1	252,23	-	-
300	2	443,88	1	449,03	1	421,41	1	369,01	1	299,78	-	-
400	3	448,61	2	464,64	1	442,74	1	389,91	1	322,14	-	-
500	3	460,44	2	474,00	1	457,80	1	404,72	1	334,42	-	-

Tabela 17 (Continuação)

Área (ha)	Prazo Calendário (dias)											
	10		15		20		25		30		35	
	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹
Composição II FT Abyara + BRS 133												
600	4	452,34	2	476,46	2	460,28	2	410,81	2	341,66	-	-
700	4	461,08	2	471,73	2	461,24	2	411,9	2	346,82	-	-
800	5	461,29	3	474,56	2	460,54	2	415,57	2	348,56	-	-
900	5	465,83	3	474,24	2	459,97	2	410,19	2	343,58	-	-
1000	6	468,92	3	479,66	2	465,21	2	413,85	2	347,00	-	-
Composição III FT Cometa + FT Abyara												
200	2	364,34	1	389,43	1	391,21	1	375,33	1	352,25	1	321,16
300	2	407,54	2	436,61	1	436,66	1	422,81	1	399,89	1	368,98
400	3	398,92	2	441,83	1	455,14	1	445,15	1	423,72	1	392,89
500	3	422,50	2	453,99	2	466,21	1	456,26	1	435,73	1	407,23
600	4	417,13	3	450,78	2	467,93	2	461,78	1	444,69	1	415,84
700	4	422,34	3	450,10	2	463,46	2	466,53	2	449,46	1	421,18
800	5	420,49	4	453,14	2	467,22	2	468,67	2	453,05	1	426,61
900	5	428,65	4	458,00	3	466,97	2	467,17	2	452,65	2	428,93
1000	6	428,26	4	460,21	3	471,89	2	465,96	2	455,76	2	432,49
Composição IV FT Cometa												
200	3	315,76	2	393,86	1	439,91	1	445,49	1	442,88	1	432,34
300	4	339,62	3	415,44	2	467,19	2	484,1	1	486,26	1	477,67
400	5	348,70	4	417,77	2	473,78	2	497,73	1	406,54	1	500,34
500	6	363,33	4	429,30	3	479,96	2	496,92	2	510,81	1	512,81
600	7	367,34	5	436,06	3	479,38	3	496,34	2	507,98	2	510,76
700	8	366,11	6	436,06	4	480,6	3	499,15	2	506,76	2	509,27
800	9	369,5	7	437,47	4	489,26	3	505,51	2	510,78	2	508,85
900	10	372,76	8	441,06	5	487,85	4	508,57	3	512,04	2	511,05
1000	11	377,67	8	445,63	5	491,80	4	509,89	3	518,14	2	513,94
Composição V FT Abyara												
200	2	312,21	1	336,53	1	320,41	1	294,76	1	254,12	1	209,95
300	2	306,57	2	364,74	1	364,64	1	340,01	1	301,33	1	257,25
400	3	326,26	2	360,51	2	368,44	1	358,4	1	232,53	1	279,49

Tabela 17 (Continuação)

Área (ha)	Prazo Calendário (dias)											
	10		15		20		25		30		35	
	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹
Composição V FT Abyara												
500	4	341,45	2	368,1	2	372,95	2	363,79	1	334,59	1	293,96
600	4	339,38	3	375,05	2	369,35	2	357,94	2	334,42	2	296,06
700	5	341,12	3	380,03	3	374,86	2	358,61	2	331,05	2	298,37
800	6	347,36	4	380,23	3	380,41	2	363,34	2	331,34	2	297,96
900	6	344,70	4	380,38	3	383,47	3	366,41	2	334,08	2	296,38
1000	7	353,83	4	385,58	3	387,05	3	370,56	2	336,85	2	298,51

5 CONCLUSÕES

Na determinação experimental das funções de perdas por atraso na colheita de soja, os cultivares FT Cometa e M – Soy 5942 revelaram menores tendências de perdas, enquanto que os cultivares BRS 133 e FT Abyara maiores tendências, quando comparadas entre si.

O programa computacional desenvolvido foi capaz de simular cenários de colheita interagindo as funções de perdas por atraso, o risco climático e as características das colhedoras, para determinar a renda líquida da operação.

A validação do modelo simulando a colheita de soja, os resultados demonstraram que, dependendo da composição de cultivares e do prazo para conclusão da operação, a capacidade e o número de colhedoras é alterado.

A simulação de cenários numa propriedade referência com 500 ha de lavoura na região de Ponta Grossa (PR) mostrou que, para qualquer composição de cultivares, a frota de colhedoras respectiva composta por duas unidades de colhedoras de 125,1 kW e uma unidade de 165,6 kW, opera na colheita de soja com capacidade ociosa. Admitida a composição de cultivares FT Cometa e M – Soy 5942, apenas uma unidade de colhedora de 121,5 kW com 4,8 m de plataforma atenderia a demanda da área, proporcionando um ganho médio de renda líquida estimado em US\$ 47.255,00. ano⁻¹.

6 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

O programa computacional desenvolvido revelou-se apropriado para o dimensionamento de colhedoras baseado no custo da pontualidade devido à sua rotina universal. Os resultados porém, dependem da precisão dos dados de entrada, particularmente da modelagem das perdas por atraso no retardamento na colheita e do critério de seleção de dia útil de campo para a operação de colheita com base na altura de chuva diária acumulada. Ambas devem ser considerados como variáveis restritivas importantes e, por conseguinte, melhor pesquisadas. Por sua vez, as variáveis econômicas como preço do produto e desempenho das máquinas, embora possam influenciar substancialmente os resultados, são relativamente mais fáceis de se apropriar com precisão maior.

Na validação do modelo para a colheita de soja (*Glycine max* (L.) Merrill), as simulações demonstraram que a demanda de máquinas e a renda líquida dependem da tendência de perdas dos cultivares, quando submetidas ao atraso na colheita. Porém, as funções de perdas utilizadas no modelo resultaram de apenas um ensaio de campo sob condições edafoclimáticas específicas. Por conseguinte, experimentos de campo devem ser repetidos. Conviria, também, para fins desta metodologia de dimensionamento, que todos cultivares de soja tivessem suas funções de perdas por atraso determinadas para as diferentes regiões agroclimáticas nas quais são recomendadas. Exaustivamente se estudou procedimento semelhante com as funções de produtividade segundo a época de semeadura dos cultivares de soja, permitindo o planejamento das operações e a estimativa da demanda de capacidade dos equipamentos nas várias etapas do sistema de produção. Idêntico procedimento se observa em relação aos cultivares de cana-de-açúcar (*Saccharum officinarum*, L.) para determinar a melhor época de colheita para maximizar o rendimento industrial de açúcar. Da mesma forma, sugere-se também estender estudos para determinar as funções de perdas por atraso na colheita de outras espécies de grande interesse econômico como o milho (*Zea mays*, L.) e o feijão (*Phaseolus vulgaris*, L.).

Finalmente, o critério utilizado para identificar o dia útil para operação de colheita, na ausência de outro mais preciso, deve ser objeto de estudo. Difere o dia útil para as operações de preparo do solo e pulverização daquele apropriado à colheita. No preparo, a umidade do

solo é, talvez, o fator mais importante. Outros fatores, porém, influenciam a disponibilidade de dias úteis para a pulverização e a colheita. Na pulverização, a velocidade do vento e a temperatura ambiente limitam a operação por conta da deriva e evaporação do produto. Na colheita, além da altura de chuva, a temperatura ambiente e a umidade relativa do ar afetam a umidade do produto a ser colhido e limitam a operação. Mais especificamente ainda sobre a altura de chuva, devem ser estudados os efeitos da intensidade e do momento de ocorrência do evento sobre o tempo de operação. Ressalte-se que as perdas devidas ao atraso dependem da duração do período de colheita. Por conseguinte, mais do que a expectativa provável de dias úteis num período e numa região, a sua distribuição é igualmente importante. As perdas se aceleram com o atraso na colheita, na medida em que o estado geral da lavoura vai se deteriorando.

É nesse sentido, pois, que devem ser desenvolvidos estudos para modelar critérios específicos e relativamente simples para determinar com precisão a disponibilidade de dias úteis para a operação de colheita de grãos.

7 REFERÊNCIAS BIBLIOGRÁFICAS

AUDSLEY, E; BOYCE, D. S. **A method of minimising the cost of harvesting an high temperature grain drying.** Journal of Agricultural Engineering Research. Southampton: University of Southampton, v. 19, p.173-189. 1974.

AMERICAN SOCIETY OF AGRICULTURAL ENGINEERS. **Agricultural engineers yearbook. ASAE Data: D230.3 and ASAE Engineering Practice: EP 391.** St. Joseph, Michigan: ASAE, p. 196-207. 1983.

AMERICAN SOCIETY OF AGRICULTURAL ENGINEERS. **Agricultural machinery management. ASAE Standards D496.2. Dec 99.** St. Joseph, Michigan: ASAE, p. 344-349. 2000 a.

AMERICAN SOCIETY OF AGRICULTURAL ENGINEERS. **Agricultural machinery management data. ASAE Standards D497.4. MAR 99.** St. Joseph, Michigan: ASAE, p. 350-357. 2000 b.

BOEHLJE, Michael D.; EIDMAN, Vernon. R. **Farm management.** New York: John Wiley and Sons. 806p. 1984.

BOISVERT, Richard N. **Available field time, yield losses and farm planning.** Canadian Journal of Agricultural Economics. Ottawa: Canadian Agr. Econ. & Farm Management Soc. v. 24, n. 1, p. 21-32. 1976.

BURROWS, W. C.; SIEMENS, J. C. **Determination of optimum machinery for corn-soybean farms.** Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 17, n. 12, p. 1130-1135. 1974.

CHANCELLOR, W. J.; CERVINKA, V. **Timeliness coefficients for rice and factors affecting their value**. Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 17, n. 5, p. 841-844. 1974.

CHEN, L. H. e McCLENDON, R. W. **Soybean and wheat double cropping simulation model**. Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 28 n. 1, p. 65-69. 1985.

EDWARDS, William; BOEHLJE, Michael. **Machinery selection considering timeliness losses**. Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 23, n. 4, p. 810-815, 821. 1980.

EMBRAPA. Centro Nacional de Pesquisa de Solos. **Sistema brasileiro de classificação de solos**. Brasília: Embrapa. Produção de Informações, Rio de Janeiro: Embrapa Solos. 412p. 1999 a.

EMBRAPA. **Recomendações técnicas para a cultura da soja no Paraná. Safra 1999/2000**. Londrina: Embrapa Soja., 1999 b. 109p.

EMBRAPA. **Sistemas de Produção 2 - Tecnologias de produção de soja – Paraná 2003**. Londrina: Embrapa Soja, 2002. 195p.

FEHR, W. R.; CAVINESS, C. E. **Estages of soybean development**. Ames: Iowa State University of Science and Technology, 1977. 11p.

FNP CONSULTORIA & COMÉRCIO. **Anuário da Agricultura Brasileira: Agrianual 2003**. São Paulo: Argos Comunicação, p. 495. 2002.

FULTON, C.; AYRES, George E.; HEADY, E. O. **Expected number of days suitable for field work in Iowa**. Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 19, n. 6, p. 1045-1047. 1976.

FURLAN, José Davi. **Modelagem de objetos através da UML – The unified modeling language**. São Paulo: Makron Books, 1998. 329 p.

GLASBEY, C. A; MacGHECAN, M. B. **The assessment of combining work-days criteria and forecasting models**. Journal of agricultural engineering research. Southampton: University of Southampton, v. 33, n. 1, p. 23-31. 1986.

HOOGENBOOM, G.; JONES, J. W.; BOOTE, K. J. **Modeling growth, development, and yield of grain legumes using SOYGRO, PNUTGRO and BEANGRO: A review**. Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 35, n. 6, p. 2043-2056. 1992.

HUAN-WEN, Gao; HUNT, Donell. **Optimum combine fleet selection with power-based models**. Transactions of the ASAE. St. Joseph, Michigan: ASAE v. 28, n. 2, p. 364-368. 1985.

HUGHES, H. A.; HOLTMAN, J. B. **Machinery complement selection based on time constraints**. Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 19, n. 5, p. 812-814. 1976.

HUNT, Donell. **Farm power and machinery management**. 4 th ed. Ames, Iowa: Iowa Estate Univesity Press, 1964. 274p.

HUNT, Donell. **Maquinaria agrícola. Rendimiento económico, costos, operaciones, potencia y selección de equipo**. Séptima edición. Mexico: Limusa, 1986. 451p.

IAPAR – INSTITUTO AGRONÔMICO DO PARANÁ. **Cartas climáticas básicas do Paraná**. Londrina: IAPAR, 1994. 45 p.

IBGE. **Levantamento sistemático da produção agrícola: pesquisa mensal de previsão e acompanhamento das safras no ano civil**. Fundação Instituto Brasileiro de Geografia e Estatística. Rio de Janeiro , v. 14, n. 12, p. 1-79. Dezembro 2002.

KASTER, Milton; QUEIROZ, Emilson França de; TERESAWA, Francisco. Primeiras da soja no Estado do Paraná. In: Miyasaka, Shiro; Medina, Júlio César (Ed.). **A soja no Brasil**. [s. l.: s. n.], p. 22-24. 1981.

LAL H.; PEART, R. M.; JONES, J. W; SHOUP, W. D. **FARMSYS – An intelligent information manager for knowledge-based systems**. Applied Engineering in Agriculture. St. Joseph, Michigan: ASAE, v. 6, n. 4, p. 525-531. 1990.

LAL H.; PEART, R. M.; JONES, J. W; SHOUP, W. D. **An objected-oriented field operations simulator in PROLOG**. Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 34, n. 3, p. 1031-1039. 1991.

LAL H.; PEART, R. M.; JONES, J. W; SHOUP, W. D. **FARMSYS – A whole-farm machinery management decision support system**. Agricultural Systems. England: Elsevier Science Publishers, v. 38, n. 3, p. 257-273. 1992.

LIN, Shiow Shong; SEVERO, Jônez Leal. **Efeito do atraso da colheita sobre a qualidade da semente e rendimento de soja**. Porto Alegre: Agronomia Sulriograndense. v. 18, n. 1, p. 47-49. 1982.

MAGALHÃES, Cláudio M. Primeiras notícias da soja no Rio Grande do Sul. In: Miyasaka, Shiro; Medina, Júlio César (Ed.). **A soja no Brasil**. [s. l.: s. n.], p. 18-20. 1981.

MEDINA, Júlio César. Primeiras notícias da soja no Brasil. In: Miyasaka, Shiro; Medina, Júlio César (Ed.). **A soja no Brasil**. [s. l.: s. n.], p. 17. 1981.

MIALHE, Luiz Geraldo. **Manual de mecanização agrícola**. São Paulo: Ceres, 1974. 301 p.

MILES, G. E.; TSAI, Y. J. **Combine systems engineering by simulation**. Transaction of the ASAE. St. Joseph, Michigan: ASAE, v. 30, n. 5, p. 1277-1281. 1987.

MINISTÉRIO DOS TRANSPORTES – DNER. Gerência de Custos Rodoviários. **Manual de composição de custos rodoviários**. Brasília: Departamento Nacional de Estradas de Rodagem – DNER, v. 1: Metodologia e Conceitos. 106p. 1972.

McGHECAN, M. B.; SAADOUN, T.; GLASBEY, C. A; OSKOUI, Eradt K. **Estimation of combine harvesting work-days from meteorological data**. Agricultural Engineer. [s.l]: v. 44, n. 3. p. 66-71. 1989.

McCLENDON, R. W., WETZTEIN, M. E.; EDWARDS, D. M.; CHEN, L. H. **Risk efficiency of machinery selection for double cropping via simulation**. Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 30, n. 5, p. 1259-1265. 1987.

MESQUITA, Cezar M.; COSTA, Newton P.; PEREIRA, José E.; MAURINA, Antoninho C.; ANDRADE, José G. M. **Caracterização da colheita mecanizada da soja no estado do Paraná**. Engenharia Agrícola. Jaboticabal: SBEA, v. 21, n. 2, p. 195-205. 2001.

MESQUITA, Cezar M.; COSTA, Newton P.; PEREIRA, José E.; MAURINA, Antoninho C.; ANDRADE, José G. M. **Perfil da colheita da soja no Brasil: Safra 1998/1999**. Engenharia Agrícola. Jaboticabal: SBEA, v. 22, n. 3, p. 398-406. 2002.

NEW HOLLAND. **Mecânica I. Manual de Serviços - 06**. New Holland LatinoAmericana Ltda. Curitiba: Centro de Treinamento. 68 p. 1998.

OSKAN, H. E.; EDWARDS, W. M. **A farmer-oriented machinery comparison model**. Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 29, n.3, p. 672-677. 1986.

OSKAN, H. E.; FRISBY, J. C. **Optimizing field machinery system energy consumption**. Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 24, n. 2, p. 296-300. 1981.

OSKAN, H. E.; HE, B; HOLMES, R. G. **Determining machinery timeliness cost in corn planting using DRAINMOD.** Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 33, n. 3, p. 718-724. 1990.

PARMAR, R. S.; McCLENDON, R. W.; WILLIAMS, E. J. **A computer simulation model for peanut machinery management.** Applied Engineering in Agriculture. St. Joseph, Michigan: ASAE, v. 10, n. 4, p. 455-461. 1994.

PARMAR, R. S.; McCLENDON, R. W.; POTTER, W. D. **Farm machinery selection using simulation and genetic algorithms.** Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 39, n. 5, p. 1905-1909. 1996.

PARSONS, S. D.; SMITH, T. W.; KRUTZ, G. W. **Machinery downtime costs.** Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 23, n. 2. p. 541-548. 1981.

PHILBROOK, B. D. **Soybean field losses as influenced by harvest delays.** Agronomy Journal. [s.l.]: v. 81, n. 2, p. 251-258. 1989.

PHILIPS, P. R.; O'CALLAGHAN, J. R. **Cereal harvesting – a mathematical model.** Journal of Agricultural Engineering Research. Southampton: University of Southampton, v. 19, n. 4, p. 415-433. 1974.

POPP, Michael P.; DILLON, Carl. R.; KEISLING, Terry C. **Economic and weather influences on soybean planting strategies on heavy soils.** Agricultural Systems. England: Elsevier Science Publishers, v. 76, n.3, p. 969-984. 2003.

PIMENTEL-GOMES, Frederico. **Curso de estatística experimental.** 14^a Ed. Piracicaba: Pimentel-Gomes. 2000. 477p.

SEAB. **[Acompanhamento da Situação Agropecuária no Paraná (Banco de dados)].** Secretaria de Estado da Agricultura e do Abastecimento do Paraná. DERAL. Curitiba, 2002.

SEAB. **[Acompanhamento da Situação Agropecuária no Paraná.** (Banco de dados)]. Secretaria de Estado da Agricultura e do Abastecimento do Paraná. DERAL. Curitiba, 2003.

SEDIYAMA, Tuneo.; CARDOSO, Antônio Américo; VIEIRA, Clibas. **Testes preliminares sobre os efeitos do retardamento da colheita da soja, cultivar ‘Viçosa’.** Viçosa: Revista Ceres, v. 19, n. 104, p. 306-310. 1972 a.

SEDIYAMA, Carlos. S.; VIEIRA, Clibas; SEDIYAMA, Tuneo.; CARDOSO, Antônio A.; ESTEVÃO, Márcio M. **Influência do retardamento da colheita sobre a deiscência das vagens e sobre a qualidade e poder germinativo das sementes de soja.** Viçosa: Experimentiae. v. 14, n. 5, p. 117-141. 1972 b.

SIEMENS, J.; HAMBURG, K.; TYRRELL, T. **A farm machinery selection and management program.** Journal of Production Agriculture. [s.l.]: v. 3, n. 2, pr. 212-219. 1990.

SHORT, Cameron; GITU, Kangethe W. **Timeliness costs for machinery selection.** Canadian Journal of Agricultural Economics. Ottawa: Canadian Agr. Econ. & Farm Management Soc. v. 39, n. 3, p. 457-462. 1991.

TULU, M. Y.; HOLTMAN, J. B; FRIDLEY, R. B.; PARSONS, S. D. **Timeliness cost and available working days: shelled corn.** Transaction of the ASAE. St. Joseph, Michigan: ASAE, v. 17, n. 5, p. 798-800. 1974.

VEIGA, Cristina Monteiro. **Modelo empírico para a seleção de máquinas agrícolas na cultura da soja considerando a pontualidade na semeadura.** 75p. Dissertação (mestrado). Piracicaba: Escola Superior de Agricultura Luiz de Queiroz, Universidade de São Paulo. 2000.

VIEIRA, Roberval Daiton; SEDIYAMA, Tuneo; FERREIRA DA SILVA, Roberto; SEDIYAMA, Carlos Siguelyuki; THIEBAUT, José Tarcísio L. **Efeito do retardamento da**

colheita sobre a qualidade de semente de soja CV “UFV-2”. Brasília: Revista brasileira de sementes, v. 4, n. 2, p. 9-22. 1982.

TSAI, Y. J.; JONES, J. W.; MISHOE, J. W. **Optimizing multiple cropping systems: a systems approach**. Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 30, n. 6, p. 1554-1561. 1987.

YOURDON, Edward. **Análise estruturada moderna**. Rio de Janeiro: Campus, 1990. 836 p.

WITNEY, B. D. **Choosing & using farm machines**. Edinburgh: Land Technology, 1988. 412p.

WHITSON, Robert E.; KAY, Ronald D.; LePORI, Wayne A.; RISTER, Edward M. **Machinery and crop selection with weather risk**. Transactions of the ASAE. St. Joseph, Michigan: ASAE, v. 24, n. 2, p. 288-291. 1981.

ANEXOS

A - Tabela 1 Produtividade de soja comercial (kg.ha⁻¹) obtido no delineamento experimental.

FT Cometa						
Subparcela	Data	Atraso	Bloco I	Bloco II	Bloco III	Média
T0	04/03	0	3.107	2.860	3.015	2.994
T1	07/03	3	3.152	2.878	2.839	2.956
T2	11/03	7	3.234	2.818	2.862	2.971
T3	14/03	10	3.398	3.134	2.660	3.064
T4	18/03	14	3.098	2.854	3.121	3.024
T5	26/03	22	3.612	2.855	2.376	2.947
T6	04/03	31	3.197	2.875	3.049	3.040
T7	08/04	35	3.062	2.650	2.821	2.844
T8	11/04	38	2.754	2.681	2.597	2.677
T9	15/04	42	2.675	2.502	2.398	2.525
M – Soy 5942						
Subparcela	Data	Atraso	Bloco I	Bloco II	Bloco III	Média
T0	26/03	0	3.572	3.270	3.038	3.293
T1	04/04	9	3.382	3.576	3.682	3.547
T2	08/04	13	3.274	3.116	3.266	3.219
T3	11/04	16	3.279	2.774	2.774	2.782
T4	15/04	20	3.493	3.518	3.737	3.582
T5	18/04	23	2.765	2.728	2.954	2.815
T6	22/04	27	3.524	3.261	3.770	3.518
T7	25/04	30	3.572	3.416	3.391	3.460
T8	10/05	45	1.774	1.460	2.058	1.764
T9	13/05	48	1.329	518	953	933

A –Tabela 1 (Continuação)

FT Abyara						
Subparcela	Data	Atraso	Bloco I	Bloco II	Bloco III	Média
T0	11/04	0	2.321	2.300	2.261	2.294
T1	15/04	4	3.014	2.938	2.963	2.972
T2	18/04	7	2.320	2.429	2.913	2.554
T3	22/04	11	2.725	3.003	2.992	2.907
T4	25/04	14	2.460	2.056	2.489	2.335
T5	10/05	29	1.616	1.295	1.695	1.536
T6	13/05	32	2020	2.095	2.253	2.122
T7	24/05	43	211	248	224	228
T8	27/05	46	177	53	122	117
T9	03/06	52	101	83	96	93
BRS 133						
Subparcela	Data	Atraso	Bloco I	Bloco II	Bloco III	Média
T0	15/04	0	3.393	3.340	3.233	3.322
T1	18/04	3	3.325	3210	3.555	3.363
T2	22/04	7	3.823	3.584	3.485	3.631
T3	25/04	10	2.887	2.638	2.293	2.606
T4	10/05	25	3.184	3.077	3.458	3.239
T5	13/05	28	2.694	2.364	2.835	2.631
T6	24/05	39	704	507	627	613
T7	27/05	42	792	734	719	748
T8	03/06	49	815	569	722	702
T9	06/06	52	510	523	486	506

A - Tabela 2 Análise de variância do ensaio experimental considerando 4 cultivares e 10 épocas ou atrasos ou atrasos de colheita.

Fonte de Variação	G.L.	S.Q.	Q.M	F	F (0,05)	F (0,01)
Repetições	2	629729,7037	314864,8519	3,032	5,14	10,92
Tratamentos (A)	3	30988851,3951	10329617,1317	99,465**	4,76	9,78
Resíduo A	6	623108,0290	103851,3382			
Subtratamentos (B)	9	66259041,2792	7362115,6977	238,972**	2,04	2,72
Interação A x B	27	39026137,4172	1445412,4969	46,918**	1,65	2,03
Resíduo B	72	2218134,9404	30807,4297			
Total	119	139745002,7646	1174327,7543			
Média geral: 2.412 kg . C.V. Fator A: 13,35% C.V. Fator B: 7,27%						

A - Tabela 3 Análise de variância dos tratamentos, considerando 10 épocas ou atrasos de colheita.

FT Cometa						
Fonte de Variação	G.L.	S.Q.	Q.M	F	F (0,05)	F (0,01)
Repetições	2	762450,049	381225,024	9,511**	3,55	6,01
Subtratamento	9	8222731,895	91414,655	2,281	2,46	3,60
Erro experimental	18	721451,011	40080,612			
Total	29	2306626,195	79538,834			
Média geral: 2.904 kg		CV: 6,89%				
M – Soy 5942						
Fonte de Variação	G.L.	S.Q.	Q.M	F	F (0,05)	F (0,01)
Repetições	2	422527,516	211263,758	0,471	3,55	6,01
Subtratamento	9	21338840,681	2370982,298	5,287**	2,46	3,60
Erro experimental	18	8072751,595	448486,200			
Total	29	29834117,182	1028762,661			
Média geral: 2.891 kg		C.V: 23,99%				
FT Abyara						
Fonte de Variação	G.L.	S.Q.	Q.M	F	F (0,05)	F (0,01)
Repetições	2	119308,499	59654,249	2,841	3,55	6,01
Subtratamento	9	36051624,127	4005736,014	190,741**	2,46	3,60
Erro experimental	18	378015,595	21000,866			
Total	29	36548947,074	1260308,520			
Média geral: 1.716 kg		C.V: 8,44%				

A – Tabela 3 (Continuação)

BRS 133						
Fonte de Variação	G.L.	S.Q.	Q.M	F	F (0,05)	F (0,01)
Repetições	2	125745,891	62872,945	2,592	3,55	6,01
Subtratamento	9	47371207,868	5263467,541	217,008**	2,46	3,60
Erro experimental	18	436584,468	24254,693			
Total	29	47933539,155	1652880,661			
Média geral: 2.136 kg		C.V: 7,29%				

B - Tabela 1 Simulações PROPRIEDADE – Frota de colhedoras.

Propriedade	Descrição	Área	Custos		Renda Líquida	%
		(ha)	US\$.sc ⁻¹	US\$.ha ⁻¹	US\$.ha ⁻¹	
PROP.PG194		300	4,82	260,50	389,46	-34,29
PROP.PG195		500	2,92	157,88	492,08	-16,98
PROP.PG196		700	2,13	115,08	534,08	-9,89
PROP.PG197		1000	1,53	82,42	567,21	-4,31
PROP.PG198		1300	1,22	66,02	582,88	-1,66
PROP.PG199	Composição I	1600	1,05	56,42	590,89	-0,31
PROP.PG200	FT Cometa + M-Soy 5942	1900	0,97	51,67	592,73	0,00
PROP.PG201		2200	1,00	52,75	585,33	-1,25
PROP.PG202		2500	1,15	58,09	570,54	-3,74
PROP.PG203		2800	1,44	68,44	546,23	-7,85
PROP.PG204		3100	1,99	86,21	508,09	-14,28
PROP.PG205		3400	2,75	108,68	460,75	-22,27
PROP.PG206		3700	3,68	137,55	400,91	-32,36
PROP.PG207		300	5,30	263,86	340,62	-30,40
PROP.PG208		500	3,31	164,35	437,01	-10,71
PROP.PG209		700	2,54	125,65	471,55	-3,65
PROP.PG210		1000	2,05	100,25	489,43	0,00
PROP.PG211	Composição II	1300	2,10	95,46	481,84	-1,55
PROP.PG212	FT Abyara + BRS 133	1600	2,58	98,99	463,55	-5,29
PROP.PG213		1900	3,55	110,77	432,30	-11,67
PROP.PG214		2200	4,99	139,04	372,65	-23,86
PROP.PG215		2500	6,66	164,35	315,43	-35,55
PROP.PG339		2800	0,00	191,97	256,74	-47,54
PROP.PG216		300	5,50	261,07	311,09	-36,43
PROP.PG217		500	3,35	159,09	412,41	-15,73
PROP.PG218		700	2,48	117,41	452,92	-7,45
PROP.PG219	Composição III	1000	1,87	87,34	480,14	-1,89
PROP.PG220	FT Cometa + FT Abyara	1300	1,62	74,39	488,90	-0,10
PROP.PG221		1600	1,53	68,55	489,38	0,00
PROP.PG222		1900	1,56	67,43	483,95	-1,11
PROP.PG223		2200	1,71	70,75	472,05	-3,54
PROP.PG224		2500	1,94	76,60	456,06	-6,81
PROP.PG225		2800	2,26	84,83	436,18	-10,87

B – Tabela 1 (Continuação)

Propriedade	Descrição	Área	Custos		Renda Líquida	%
		(ha)	US\$.sc ⁻¹	US\$.ha ⁻¹	US\$.ha ⁻¹	
PROPPG.336	Composição III	3100	2,67	95,17	412,99	-15,61
PROP.PG337	FT Cometa + FT Abyara	3400	3,23	109,89	381,18	-22,11
PROP.PG338		3700	3,96	128,49	341,96	-30,12
PROP.PG226	Composição IV FT Cometa	300	5,08	259,24	352,59	-31,72
PROP.PG227		500	3,11	158,46	452,84	-12,30
PROP.PG228		700	2,29	116,46	492,42	-4,64
PROP.PG229		1000	1,76	88,80	516,37	0,00
PROP.PG230		1300	1,78	84,55	507,34	-1,75
PROP.PG231		1600	2,14	92,77	479,82	-7,08
PROP.PG232		1900	3,34	129,03	399,27	-22,68
PROP.PG233		2200	5,21	180,30	290,70	-43,70
PROP.PG234		2500	7,36	230,32	185,95	-63,99
PROP.PG235		2800	9,56	271,20	100,47	-80,54
PROP.PG236		300	5,91	261,75	269,27	-29,97
PROP.PG237		500	3,80	166,37	358,71	-6,70
PROP.PG238	Composição V FT Abyara	700	3,07	131,77	384,48	0,00
PROP.PG239		1000	2,95	119,98	375,68	-2,29
PROP.PG240		1300	3,48	130,33	337,50	-12,22
PROP.PG241		1600	4,50	152,59	281,98	-26,66
PROP.PG242		1900	6,06	183,13	213,03	-44,59
PROP.PG243		2200	8,13	218,20	137,05	-64,35
PROP.PG244		2500	10,50	251,84	65,10	-83,07
PROP.PG245		2800	13,00	281,38	2,27	-99,41

B - Tabela 2 Simulações PROPRIEDADE – Colhedora de 121,5 kW.

Propriedade	Descrição	Área (ha)	Custos		Renda Líquida US\$,ha ⁻¹	%
			US\$.sc ⁻¹	US\$.ha ⁻¹		
PROP.PG050		200	2,11	113,74	536,13	-8,60
PROP.PG051		300	1,46	78,85	570,43	-2,75
PROP.PG052		400	1,17	62,91	584,55	-0,35
PROP.PG053	Composição I	500	1,06	56,57	586,59	0,00
PROP.PG054	FT Cometa + M-Soy 5942	600	1,15	59,45	573,69	-2,20
PROP.PG055		700	1,48	71,25	544,94	-7,10
PROP.PG056		800	2,15	92,28	498,92	-14,95
PROP.PG057		900	3,26	125,71	428,79	-26,90
PROP.PG058		1000	4,64	166,32	344,76	-41,23
PROP.PG334		1100	6,27	207,08	259,54	-55,75
PROP.PG335		1200	7,91	244,08	184,75	-68,50
PROP.PG059		200	2,56	123,30	468,88	-2,06
PROP.PG060		300	2,19	103,62	478,73	0,00
PROP.PG061	Composição II	400	2,69	104,73	458,70	-4,18
PROP.PG062	FT Abyara + BRS 133	500	3,99	123,05	411,35	-14,07
PROP.PG063		600	6,06	155,53	339,12	-29,16
PROP.PG325		700	8,52	194,00	256,96	-46,32
PROP.PG328		800	12,73	234,80	171,29	-64,22
PROP.PG064		200	2,48	116,68	453,01	-6,33
PROP.PG065		300	1,85	85,89	479,11	-0,94
PROP.PG066		400	1,66	74,74	483,64	0,00
PROP.PG067		500	1,7	73,01	476,44	-1,49
PROP.PG068	Composição III	600	1,92	77,88	459,55	-4,98
PROP.PG069	FT Cometa + FT Abyara	700	2,31	87,62	434,93	-10,07
PROP.PG329		800	2,85	100,93	404,42	-16,38
PROP.PG330		900	3,66	121,86	359,4	-25,69
PROP.PG331		1000	4,75	149,19	302,09	-37,54
PROP.PG332		1100	6,01	177,94	242,32	-49,90
PROP.PG070		200	2,29	116,04	492,85	-2,70
PROP.PG071	Composição IV	300	1,87	91,53	506,52	0,00
PROP.PG072	FT Cometa	400	2,22	97,13	477,5	-5,73
PROP.PG073		500	3,87	146,13	368,22	-27,30
PROP.PG074		600	6,51	213,08	226,38	-55,31

B – Tabela 2 (Continuação)

Propriedade	Descrição	Área (ha)	Custos		Renda Líquida	%
			US\$.sc ⁻¹	US\$.ha ⁻¹	US\$.ha ⁻¹	
PROP.PG326	Composição IV	700	9,39	269,82	107,08	-78,86
PROP.PG333	FT Cometa	800	12,28	312,58	17,21	-96,60
PROP.PG075	Composição V FT Abyara	200	3,20	135,60	375,41	0,00
PROP.PG076		300	3,39	131,78	347,73	-7,37
PROP.PG077		400	4,58	156,43	280,67	-25,24
PROP.PG078		500	6,68	196,33	189,89	-49,42
PROP.PG079		600	9,97	241,34	93,15	-75,19
PROP.PG327		700	12,82	281,35	6,30	-98,32

B - Tabela 3 Simulações PROPRIEDADE – Colhedora de 132,5 kW.

Propriedade	Descrição	Área (ha)	Custos		Renda Líquida US\$,ha ⁻¹	%
			US\$.sc ⁻¹	US\$,ha ⁻¹		
PROP.PG080		200	2,68	144,60	505,34	-14,18
PROP.PG081		300	1,84	99,09	550,70	-6,48
PROP.PG082		400	1,41	76,17	573,07	-2,68
PROP.PG083		500	1,18	63,33	584,67	-0,71
PROP.PG084	Composição I	600	1,06	56,75	588,87	0,00
PROP.PG085	FT Cometa + M-Soy 5942	700	1,04	54,85	585,88	-0,51
PROP.PG086		800	1,13	58,07	574,33	-2,47
PROP.PG087		900	1,39	67,69	551,36	-6,37
PROP.PG088		1000	1,82	81,25	519,79	-11,73
PROP.PG089		1100	2,43	100,40	479,98	-18,49
PROP.PG340		1200	3,39	129,49	419,32	-28,79
PROP.PG341		1300	4,41	159,60	357,20	-39,34
PROP.PG090		200	3,09	153,03	446,35	-7,13
PROP.PG091		300	2,33	114,38	477,98	-0,55
PROP.PG092	Composição II	400	2,14	101,32	480,62	0,00
PROP.PG093	FT Abyara + BRS 133	500	2,38	100,82	467,50	-2,73
PROP.PG094		600	3,48	112,44	435,25	-9,44
PROP.PG095		700	4,45	131,22	390,86	-18,68
PROP.PG096		800	6,08	154,88	338,37	-29,60
PROP.PG342		900	8,22	189,14	266,05	-44,64
PROP.PG097		200	3,09	146,33	424,85	-12,57
PROP.PG098		300	2,19	102,92	465,79	-4,14
PROP.PG099		400	1,80	83,36	481,45	-0,92
PROP.PG100		500	1,63	74,08	485,92	0,00
PROP.PG101	Composição III	600	1,62	71,18	482,74	-0,65
PROP.PG102	FT Cometa + FT Abyara	700	1,71	72,27	473,77	-2,50
PROP.PG103		800	1,90	76,62	459,98	-5,34
PROP.PG104		900	2,19	84,04	441,41	-9,16
PROP.PG343		1000	2,56	93,20	419,86	-13,59
PROP.PG344		1100	3,05	105,65	392,30	-19,27
PROP.PG345		1200	3,71	122,69	356,06	-26,72
PROP.PG346		1300	4,53	143,55	312,31	-35,73

B – Tabela 3 (Continuação)

Propriedade	Descrição	Área (ha)	Custos		Renda Líquida	%
			US\$.sc ⁻¹	US\$.ha ⁻¹	US\$.ha ⁻¹	
PROP.PG105	Composição IV FT Cometa	200	2,86	145,71	465,03	-8,46
PROP.PG106		300	2,04	103,33	503,70	-0,84
PROP.PG107		400	1,83	89,49	507,98	0,00
PROP.PG108		500	2,05	92,16	488,76	-3,78
PROP.PG109		600	2,89	116,37	430,80	-15,19
PROP.PG110		700	4,57	164,63	327,10	-35,61
PROP.PG347		800	6,65	215,86	219,13	-56,86
PROP.PG348		900	8,84	259,72	127,07	-74,99
PROP.PG111	Composição V FT Abyara	200	3,62	156,96	364,23	-2,99
PROP.PG112		300	3,08	128,29	375,46	0,00
PROP.PG113		400	3,37	130,31	348,05	-7,30
PROP.PG114		500	4,21	148,17	298,47	-20,51
PROP.PG115		600	5,55	174,98	235,49	-37,28
PROP.PG116		700	7,46	208,76	160,88	-57,15
PROP.PG349		800	9,72	242,70	87,62	-76,66

B - Tabela 4 Simulações PROPRIEDADE – Colhedora de 165,6 kW.

Propriedade	Descrição	Área (ha)	Custos		Renda Líquida US\$,ha ⁻¹	%
			US\$.sc ⁻¹	US\$.ha ⁻¹		
PROP.PG117		200	3,23	1,74	475,36	-20,50
PROP.PG118		300	2,18	117,45	532,49	-10,95
PROP.PG119		400	1,66	89,73	560,12	-6,33
PROP.PG120		500	1,35	72,65	576,96	-3,51
PROP.PG121		600	1,15	62,07	587,10	-1,81
PROP.PG122		700	1,02	55,00	593,38	-0,76
PROP.PG123	Composição I	800	0,93	49,92	597,23	-0,12
PROP.PG124	FT Cometa + M-Soy 5942	900	0,88	47,30	597,94	0,00
PROP.PG125		1000	0,87	46,33	595,92	-0,34
PROP.PG126		1100	0,93	48,37	588,87	-1,52
PROP.PG127		1200	1,03	52,46	578,20	-3,30
PROP.PG128		1300	1,19	58,30	564,18	-5,65
PROP.PG129		1400	1,46	67,90	543,15	-9,16
PROP.PG130		1500	1,77	78,00	521,20	-12,83
PROP.PG360		1600	2,24	92,57	490,62	-17,95
PROP.PG361		1700	2,78	108,50	457,37	-23,51
PROP.PG131		200	3,61	179,44	423,56	-15,04
PROP.PG132		300	2,55	126,07	473,13	-5,10
PROP.PG133		400	2,09	102,85	491,75	-1,36
PROP.PG134		500	1,86	90,79	498,55	0,00
PROP.PG135	Composição II	600	1,87	87,84	493,41	-1,03
PROP.PG136	FT Abyara + BRS 133	700	2,12	90,01	481,18	-3,48
PROP.PG137		800	2,47	93,24	468,41	-6,05
PROP.PG138		900	3,38	104,08	442,17	-11,31
PROP.PG139		1000	3,86	114,91	416,53	-16,45
PROP.PG350		1100	4,96	135,67	371,99	-25,39
PROP.PG351		1200	6,35	157,17	326,45	-34,52
PROP.PG140		200	3,70	175,44	396,45	-20,18
PROP.PG141		300	2,52	119,23	451,70	-9,05
PROP.PG142	Composição III	400	1,97	92,87	476,59	-4,04
PROP.PG143	FT Cometa + FT Abyara	500	1,66	77,70	489,62	-1,42
PROP.PG144		600	1,51	69,51	494,99	-0,34
PROP.PG145		700	1,42	64,72	496,67	0,00

B – Tabela 4 (Continuação)

Propriedade	Descrição	Área (ha)	Custos		Renda Líquida	%
			US\$.sc ⁻¹	US\$.ha ⁻¹	US\$.ha ⁻¹	
PROP.PG146		800	1,40	62,36	495,12	-0,31
PROP.PG147		900	1,43	62,15	490,97	-1,15
PROP.PG148		1000	1,50	63,36	484,59	-2,43
PROP.PG149	Composição III	1100	1,62	66,53	475,29	-4,30
PROP.PG150	FT Cometa + FT Abyara	1200	1,78	70,69	464,47	-6,48
PROP.PG151		1300	1,99	76,06	451,40	-9,11
PROP.PG352		1400	2,23	82,39	436,95	-12,02
PROP.PG353		1500	2,49	89,14	421,71	-15,09
PROP.PG360		1600	2,84	98,17	402,24	-19,01
PROP.PG361		1700	3,25	109,11	378,96	-23,70
PROP.PG152		200	3,41	174,04	437,62	-16,70
PROP.PG153		300	2,33	118,64	492,02	-6,35
PROP.PG154		400	1,83	92,50	516,10	-1,76
PROP.PG155		500	1,58	79,41	525,36	0,00
PROP.PG156	Composição IV	600	1,58	76,63	519,83	-1,05
PROP.PG157	FT Cometa	700	1,74	79,57	505,82	-3,72
PROP.PG158		800	2,08	88,91	480,95	-8,45
PROP.PG159		900	2,84	112,18	429,49	-18,25
PROP.PG160		1000	3,98	146,19	357,39	-31,97
PROP.PG354		1100	5,31	181,31	283,78	-45,98
PROP.PG355		1200	6,77	216,03	211,48	-59,75
PROP.PG161		200	4,07	179,12	349,16	-11,88
PROP.PG162		300	3,01	130,31	390,38	-1,47
PROP.PG163		400	2,68	113,29	396,21	0,00
PROP.PG164	Composição V	500	2,75	111,32	383,22	-3,28
PROP.PG165	FT Abyara	600	3,09	118,43	357,95	-9,66
PROP.PG166		700	3,66	131,64	323,39	-18,38
PROP.PG167		800	4,44	148,72	283,11	-28,55
PROP.PG356		900	5,46	169,56	236,59	-40,29
PROP.PG357		1000	6,75	193,35	185,05	-53,29

B - Tabela 5 Simulações PROPRIEDADE – Frota de colhedoras (RESUMO).

Área (ha)	Composições									
	I		II		III		IV		V	
	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%
300	389,46	-34,29	340,62	-30,40	311,09	-36,43	352,59	-31,72	269,27	-29,97
500	492,08	-16,98	437,01	-10,71	412,41	-15,73	452,84	-12,30	358,71	-6,70
700	534,08	-9,89	471,55	-3,65	452,92	-7,45	492,42	-4,64	384,48	0,00
1000	567,21	-4,31	489,43	0,00	480,14	-1,89	516,37	0,00	375,68	-2,29
1300	582,88	-1,66	481,84	-1,55	488,90	-0,10	507,34	-1,75	337,50	-12,22
1600	590,89	-0,31	463,55	-5,29	489,38	0,00	479,82	-7,08	281,98	-26,66
1900	592,73	0,00	432,30	-11,67	483,95	-1,11	399,27	-22,68	213,03	-44,59
2200	585,33	-1,25	372,65	-23,86	472,05	-3,54	290,70	-43,70	137,05	-64,35
2500	570,54	-3,74	315,43	-35,55	456,06	-6,81	185,95	-63,99	65,10	-83,07
2800	546,23	-7,85	256,74	-47,54	436,18	-10,87	-	-	-	-
3100	508,09	-14,28	-	-	412,99	-15,61	-	-	-	-
3400	460,75	-22,27	-	-	381,18	-22,11	-	-	-	-
3700	400,91	-32,36	-	-	341,96	-30,12	-	-	-	-

B - Tabela 6 Simulações PROPRIEDADE - Colhedora de 125,1 kW (RESUMO).

Área (ha)	Composições									
	I		II		III		IV		V	
	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%
200	536,13	-8,60	468,88	-2,06	453,01	-6,33	492,85	-2,70	375,41	0,00
300	570,43	-2,75	478,73	0,00	479,11	-0,94	506,52	0,00	347,73	-7,37
400	584,55	-0,35	458,70	-4,18	483,64	0,00	477,5	-5,73	280,67	-25,24
500	586,59	0,00	411,35	-14,07	476,44	-1,49	368,22	-27,30	189,89	-49,42
600	573,69	-2,20	339,12	-29,16	459,55	-4,98	226,38	-55,31	93,15	-75,19
700	544,94	-7,10	256,96	-46,32	434,93	-10,07	107,08	-78,86	6,30	-98,32
800	498,92	-14,95	171,29	-64,22	404,42	-16,38	-	-	-	-
900	428,79	-26,90	-	-	359,4	-25,69	-	-	-	-
1000	344,76	-41,23	-	-	302,09	-37,54	-	-	-	-

B - Tabela 7 Simulações PROPRIEDADE - Colhedora de 132,5 kW (RESUMO).

Área (ha)	Composições									
	I		II		III		IV		V	
	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%
200	505,34	-14,18	446,35	-7,13	424,85	-12,57	465,03	-8,46	364,23	-2,99
300	550,70	-6,48	477,98	-0,55	465,79	-4,14	503,70	-0,84	375,46	0,00
400	573,07	-2,68	480,62	0,00	481,45	-0,92	507,98	0,00	348,05	-7,30
500	584,67	-0,71	467,50	-2,73	485,92	0,00	488,76	-3,78	298,47	-20,51
600	588,87	0,00	435,25	-9,44	482,74	-0,65	430,80	-15,19	235,49	-37,28
700	585,88	-0,51	390,86	-18,68	473,77	-2,50	327,10	-35,61	160,88	-57,15
800	574,33	-2,47	338,37	-29,60	459,98	-5,34	219,13	-56,86	87,62	-76,66
900	551,36	-6,37	266,05	-44,64	441,41	-9,16	127,07	-74,99	-	-
1000	519,79	-11,73	-	-	419,86	-13,59	-	-	-	-
1100	479,98	-18,49	-	-	392,30	-19,27	-	-	-	-
1200	419,32	-28,79	-	-	356,06	-26,72	-	-	-	-
1300	357,20	-39,34	-	-	312,31	-35,73	-	-	-	-

B - Tabela 8 Simulações PROPRIEDADE - Colhedora de 165,6 kW (RESUMO).

Área (ha)	Composições									
	I		II		III		IV		V	
	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%	US\$.ha ⁻¹	%
200	475,36	-20,50	423,56	-15,04	396,45	-20,18	437,62	-16,70	349,16	-11,88
300	532,49	-10,95	473,13	-5,10	451,70	-9,05	492,02	-6,35	390,38	-1,47
400	560,12	-6,33	491,75	-1,36	476,59	-4,04	516,10	-1,76	396,21	0,00
500	576,96	-3,51	498,55	0,00	489,62	-1,42	525,36	0,00	383,22	-3,28
600	587,10	-1,81	493,41	-1,03	494,99	-0,34	519,83	-1,05	357,95	-9,66
700	593,38	-0,76	481,18	-3,48	496,67	0,00	505,82	-3,72	323,39	-18,38
800	597,23	-0,12	468,41	-6,05	495,12	-0,31	480,95	-8,45	283,11	-28,55
900	597,94	0,00	442,17	-11,31	490,97	-1,15	429,49	-18,25	236,59	-40,29
1000	595,92	-0,34	416,53	-16,45	484,59	-2,43	357,39	-31,97	185,05	-53,29
1100	588,87	-1,52	371,99	-25,39	475,29	-4,30	283,78	-45,98	-	-
1200	578,20	-3,30	326,45	-34,52	464,47	-6,48	211,48	-59,75	-	-
1300	564,18	-5,65	-	-	451,40	-9,11	-	-	-	-
1400	543,15	-9,16	-	-	436,95	-12,02	-	-	-	-
1500	521,20	-12,83	-	-	421,71	-15,09	-	-	-	-
1600	490,62	-17,95	-	-	402,24	-19,01	-	-	-	-
1700	457,37	-23,51	-	-	378,96	-23,70	-	-	-	-

C - Tabela 1 Simulações CENÁRIO – Colhedora de 125,1 kW.

Cenário	Propriedade (Origem)	Descrição	Área (ha)	Prazo Calendário (dias)											
				10		15		20		25		30		35	
				n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹
CEN.PG001	PROP.PG050	Composição I FT Cometa + M-Soy 5942	200	3	456,03	2	507,11	1	536,84	1	534,47	1	522,34	1	499,76
CEN.PG002	PROP.PG051		300	4	467,43	3	519,57	2	552,98	2	557,87	1	550,08	1	528,84
CEN.PG003	PROP.PG052		400	5	475,87	4	520,44	2	552,98	2	565,98	1	563,05	1	543,39
CEN.PG004	PROP.PG053		500	6	484,54	4	528,81	3	555,81	2	561,48	2	562,89	1	550,68
CEN.PG005	PROP.PG054		600	7	487,91	5	533,78	3	561,31	3	562,66	2	560,35	2	549,48
CEN.PG006	PROP.PG055		700	8	488,25	6	533,79	4	562,67	3	569,17	2	561,62	2	549,14
CEN.PG007	PROP.PG056		800	9	490,77	7	535,11	4	565,93	3	571,81	2	564,85	2	546,58
CEN.PG008	PROP.PG057		900	10	497,14	7	540,10	4	566,87	4	574,25	3	568,16	2	549,05
CEN.PG009	PROP.PG058		1000	11	494,29	8	538,41	5	565,12	4	572,26	3	569,01	2	551,76
CEN.PG010	PROP.PG059	Composição II FT Abyara + BRS 133	200	3	413,95	2	439,74	1	419,52	1	367,06	1	299,57	-	-
CEN.PG011	PROP.PG060		300	4	418,62	2	454,62	2	441,64	2	393,43	2	324,23	-	-
CEN.PG012	PROP.PG061		400	5	429,86	3	452,23	2	441,97	2	399,44	2	332,95	-	-
CEN.PG013	PROP.PG062		500	6	439,43	3	458,64	2	447,84	2	397,28	2	330,98	-	-
CEN.PG014	PROP.PG063		600	7	439,85	4	464,11	3	447,6	3	397,62	3	329,03	-	-
CEN.PG110	PROP.PG168		700	8	441,75	4	465,48	3	454,07	3	402,48	3	334,86	-	-
CEN.PG111	PROP.PG169		800	9	445,36	5	466,05	4	453,58	3	405,26	3	337,44	-	-
CEN.PG112	PROP.PG170		900	10	449,34	5	470,45	4	456,76	4	407,41	4	339,85	-	-
CEN.PG113	PROP.PG171		1000	11	447,58	6	468,27	4	457,16	4	406,61	4	340,34	-	-

C – Tabela 1 (Continuação)

Cenário	Propriedade (Origem)	Descrição	Área (ha)	Prazo Calendário (dias)											
				10		15		20		25		30		35	
				n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹
CEN.PG015	PROP.PG064	Composição III FT Cometa + FT Abyara	200	3	355,78	2	411,42	1	430,99	1	422,58	1	401,55	1	370,67
CEN.PG016	PROP.PG065		300	4	378,61	3	422,66	2	447,11	2	443,60	1	428,08	1	399,76
CEN.PG017	PROP.PG066		400	5	382,84	4	425,61	2	446,22	2	452,30	2	438,65	1	413,52
CEN.PG018	PROP.PG067		500	6	358,59	4	434,51	3	452,09	2	448,88	2	442,09	2	420,88
CEN.PG019	PROP.PG068		600	7	395,48	5	438,07	3	455,44	3	450,18	2	439,56	2	420,41
CEN.PG020	PROP.PG069		700	8	398,06	6	439,60	4	457,82	3	454,71	2	439,27	2	421,11
CEN.PG114	PROP.PG172		800	9	400,35	7	440,28	4	459,62	4	457,68	3	441,80	2	417,51
CEN.PG115	PROP.PG173		900	10	404,71	7	444,78	4	462,19	4	461,57	3	445,76	2	419,97
CEN.PG116	PROP.PG174		1000	11	402,35	8	443,76	5	460,68	4	458,94	3	447,85	3	422,32
CEN.PG021	PROP.PG070	Composição IV FT- Cometa	200	5	282,86	4	368,74	2	438,72	2	469,45	1	482,22	1	477,38
CEN.PG022	PROP.PG071		300	7	306,19	5	391,65	3	488,10	3	471,22	2	487,60	2	495,24
CEN.PG023	PROP.PG072		400	9	316,96	7	396,94	4	459,00	3	479,16	2	488,45	2	488,99
CEN.PG024	PROP.PG073		500	11	319,12	8	403,64	5	461,31	4	484,67	3	496,80	2	494,48
CEN.PG025	PROP.PG074		600	13	321,75	10	405,16	6	464,61	5	488,34	3	500,01	3	501,75
CEN.PG117	PROP.PG175		700	15	328,25	11	410,28	7	466,96	5	491,96	4	500,77	3	505,40
CEN.PG118	PROP.PG176		800	17	329,08	13	410,16	8	469,17	6	490,69	4	503,56	3	504,57
CEN.PG119	PROP.PG177		900	19	330,52	14	413,98	8	470,10	7	495,61	5	504,54	4	506,70
CEN.PG120	PROP.PG178		1000	21	333,12	16	414,56	9	471,91	7	494,23	5	505,69	4	506,61

C – Tabela 1 (Continuação)

Cenário	Propriedade (Origem)	Descrição	Área (ha)	Prazo Calendário (dias)											
				10		15		20		25		30		35	
				n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹
CEN.PG026	PROP.PG075	Composição V FT Abyara	200	3	276,78	2	323,67	2	339,09	1	334,12	1	300,95	1	257,04
CEN.PG027	PROP.PG076		300	4	239,22	3	341,90	2	344,95	2	341,85	2	314,68	1	282,67
CEN.PG028	PROP.PG077		400	5	306,73	4	349,27	3	354,10	2	340,32	2	312,58	2	283,90
CEN.PG029	PROP.PG078		500	7	311,32	4	355,10	3	362,44	3	349,40	2	318,42	2	281,00
CEN.PG030	PROP.PG079		600	8	308,49	5	356,04	4	362,08	3	354,26	3	322,96	2	285,04
CEN.PG121	PROP.PG179		700	9	318,57	6	361,26	4	366,88	4	354,17	3	328,22	3	289,48
CEN.PG122	PROP.PG180		800	10	315,54	7	360,75	5	368,72	4	355,88	4	325,93	3	292,8
CEN.PG123	PROP.PG181		900	12	319,06	7	365,47	6	370,54	5	358,40	4	328,11	3	293,00
CEN.PG124	PROP.PG182	1000	13	324,00	8	364,64	6	369,52	5	357,57	4	331,27	4	293,15	

C - Tabela 2 Simulação CENÁRIO – Colhedora de 132,5 kW.

Tabela 2. Simulações CEN-PROP - Colmeirão de 152,5 kW.															
Propriedade			Área (ha)	Prazo Calendário (dias)											
Cenário	(Origem)	Descrição		10		15		20		25		30		35	
				n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹
CEN.PG031	PROP.PG080	Composição I FT Cometa + M-Soy 5942	200	2	457,45	1	500,90	1	513,79	1	510,18	1	496,25	1	473,65
CEN.PG032	PROP.PG081		300	3	462,96	2	511,82	1	545,76	1	544,63	1	532,41	1	513,10
CEN.PG033	PROP.PG082		400	4	471,64	3	523,06	2	555,91	2	560,67	1	552,08	1	531,64
CEN.PG034	PROP.PG083		500	4	491,04	3	527,93	2	557,32	2	571,24	2	564,14	1	542,77
CEN.PG035	PROP.PG084		600	5	485,84	4	528,87	3	554,34	2	565,78	2	561,5	1	548,62
CEN.PG036	PROP.PG085		700	6	483,48	5	520,2	3	559,57	2	563,86	2	564,98	2	553,46
CEN.PG037	PROP.PG086		800	7	491,76	5	537,02	3	564,08	3	565,36	2	562,86	2	551,79
CEN.PG038	PROP.PG087		900	8	487,16	6	533,54	4	562,92	3	569,67	2	562,26	2	549,95
CEN.PG039	PROP.PG088		1000	8	499,1	6	540,97	4	568,53	3	574,51	2	565,56	2	547,51
CEN.PG040	PROP.PG090	Composição II FT Abyara + BRS 133	200	2	412,06	1	423,76	1	396,41	1	344,52	1	275,29	-	-
CEN.PG041	PROP.PG091		300	3	418,26	2	444,80	1	433,10	1	380,52	1	309,81	-	-
CEN.PG042	PROP.PG092		400	4	422,50	2	457,63	2	444,44	2	396,17	2	327,06	-	-
CEN.PG043	PROP.PG093		500	4	441,76	2	457,85	2	450,3	2	401,81	2	337,40	-	-
CEN.PG044	PROP.PG094		600	5	439,14	3	454,88	2	444,06	2	395,41	2	329,44	-	-
CEN.PG045	PROP.PG095		700	6	440,59	3	461,44	2	448,95	2	398,2	2	331,78	-	-
CEN.PG046	PROP.PG096		800	7	443,40	4	466,94	3	450,28	3	400,28	2	331,76	-	-
CEN.PG125	PROP.PG183		900	8	440,96	4	465,53	3	454,44	3	402,94	3	335,41	-	-
CEN.PG126	PROP.PG184		1000	8	451,99	4	470,44	3	457,77	3	407,42	3	340,22	-	-

C – Tabela 2 (Continuação)

Propriedade			Área (ha)	Prazo Calendário (dias)											
Cenário	(Origem)	Descrição		10		15		20		25		30		35	
				n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹
CEN.PG047	PROP.PG097		200	2	371,08	1	408,75	1	411,52	1	398,31	1	375,47	1	344,57
CEN.PG048	PROP.PG098		300	3	372,66	2	418,10	2	438,83	1	431,20	1	411,62	1	384,02
CEN.PG049	PROP.PG099		400	4	382,90	3	426,25	2	450,14	2	446,46	1	430,87	1	402,56
CEN.PG050	PROP.PG100	Composição III	500	4	399,46	3	432,98	2	450,36	2	456,56	2	440,54	1	412,76
CEN.PG051	PROP.PG101	FT Cometa + FT Abyara	600	5	397,11	4	435,15	3	448,96	2	453,13	2	440,72	2	418,76
CEN.PG052	PROP.PG102		700	6	394,07	5	435,39	3	453,95	3	449,32	2	444,87	2	423,73
CEN.PG053	PROP.PG103		800	7	399,48	5	441,38	3	458,31	3	452,91	2	442,09	2	422,72
CEN.PG054	PROP.PG104		900	8	396,85	6	439,31	4	458,05	3	455,19	2	439,91	2	421,93
CEN.PG127	PROP.PG185		1000	8	408,94	6	446,96	4	462,51	3	460,77	3	442,89	2	419,40
CEN.PG055	PROP.PG105		200	4	277,49	3	370,61	2	434,38	2	455,94	1	460,30	1	452,46
CEN.PG056	PROP.PG106		300	5	299,17	4	375,03	3	436,18	2	469,56	2	480,19	1	487,11
CEN.PG057	PROP.PG107		400	7	311,65	5	396,01	3	451,74	3	474,53	2	490,59	2	497,96
CEN.PG058	PROP.PG108	Composição IV	500	8	326,96	6	408,03	4	459,85	3	481,92	2	490,10	2	491,20
CEN.PG059	PROP.PG109	FT Cometa	600	10	318,33	8	402,18	5	459,86	4	486,13	3	492,11	2	493,67
CEN.PG060	PROP.PG110		700	11	328,31	9	409,88	5	464,49	4	488,45	3	502,19	2	500,09
CEN.PG128	PROP.PG186		800	13	326,96	10	409,31	6	467,97	5	491,35	3	502,78	3	504,35
CEN.PG129	PROP.PG187		900	15	324,88	11	408,35	7	466,03	5	491,54	4	500,65	3	505,59
CEN.PG130	PROP.PG188		1000	16	332,15	12	415,9	7	471,90	6	494,01	4	505,45	3	505,63

C – Tabela 2 (Continuação)

Cenário	Propriedade (Origem)	Descrição	Área (ha)	Prazo Calendário (dias)											
				10		15		20		25		30		35	
				n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹
CEN.PG061	PROP.PG111	Composição V FT Abyara	200	2	257,27	2	332,83	1	337,83	1	314,52	1	276,59	1	232,67
CEN.PG062	PROP.PG112		300	3	287,32	2	324,31	2	343,29	2	340,14	1	312,35	1	270,06
CEN.PG063	PROP.PG113		400	4	297,71	3	345,54	2	348,29	2	344,80	1	317,44	1	285,26
CEN.PG064	PROP.PG114		500	5	312,27	3	360,14	3	357,79	2	341,98	2	315,80	2	285,05
CEN.PG065	PROP.PG115		600	6	306,24	4	352,11	3	361,06	3	347,09	2	317,01	2	280,99
CEN.PG066	PROP.PG116		700	7	318,35	5	359,60	3	368,68	3	354,08	3	320,57	2	285,42
CEN.PG131	PROP.PG189		800	8	312,74	5	359,44	4	365,12	3	356,99	3	325,57	2	287,58
CEN.PG132	PROP.PG190		900	9	316,59	6	360,35	4	366,46	4	354,08	3	328,42	3	289,80
CEN.PG133	PROP.PG191		1000	10	320,13	6	367,55	5	370,34	4	357,33	3	329,30	3	293,43

C - Tabela 3 Simulações CENÁRIO – Colhedora de 165,6 kW.

Tabela 3 Simulações CENÁRIOS Comedouro de 105,0 kW.															
Cenário	Propriedade		Área (ha)	Prazo Calendário (dias)											
	(Origem)	Descrição		10		15		20		25		30		35	
				n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹
CEN.PG067	PROP.PG117		200	2	456,85	1	481,56	1	493,43	1	487,17	1	473,02	1	450,24
CEN.PG068	PROP.PG118		300	2	493,33	2	528,73	1	538,88	1	534,65	1	520,66	1	498,05
CEN.PG069	PROP.PG119		400	3	495,74	2	536,78	1	560,19	1	556,98	1	544,48	1	521,96
CEN.PG070	PROP.PG120	Composição I	500	3	510,98	2	547,25	2	571,83	1	569,24	1	556,50	1	536,30
CEN.PG071	PROP.PG121	FT Cometa + M-Soy 5942	600	4	504,84	3	546,68	2	572,99	2	575,51	1	566,41	1	544,91
CEN.PG072	PROP.PG122		700	4	512,79	3	544,65	2	569,74	2	580,80	1	572,67	1	551,06
CEN.PG073	PROP.PG123		800	5	511,57	4	547,38	2	572,99	2	581,93	1	576,66	1	556,38
CEN.PG074	PROP.PG124		900	5	516,36	4	551,39	3	571,72	2	579,64	2	573,42	1	558,63
CEN.PG075	PROP.PG125		1000	6	518,47	4	554,04	3	575,24	2	578,36	2	576,53	1	562,13
CEN.PG076	PROP.PG131		200	2	403,4	1	400,86	1	375,91	1	321,55	1	252,23	-	-
CEN.PG077	PROP.PG132		300	2	443,88	1	449,03	1	421,41	1	369,01	1	299,78	-	-
CEN.PG078	PROP.PG133		400	3	448,61	2	464,64	1	442,74	1	389,91	1	322,14	-	-
CEN.PG079	PROP.PG134	Composição II	500	3	460,44	2	474,00	1	457,80	1	404,72	1	334,42	-	-
CEN.PG080	PROP.PG135	FT Abyara+ BRS 133	600	4	452,34	2	476,46	2	460,28	2	410,81	2	341,66	-	-
CEN.PG081	PROP.PG136		700	4	461,08	2	471,73	2	461,24	2	411,9	2	346,82	-	-
CEN.PG082	PROP.PG137		800	5	461,29	3	474,56	2	460,54	2	415,57	2	348,56	-	-
CEN.PG083	PROP.PG138		900	5	465,83	3	474,24	2	459,97	2	410,19	2	343,58	-	-
CEN.PG084	PROP.PG139		1000	6	468,92	3	479,66	2	465,21	2	413,85	2	347,00	-	-

C – Tabela 3 (Continuação)

Propriedade		Área (ha)	Prazo Calendário (dias)												
Cenário	(Origem)		Descrição	10		15		20		25		30		35	
				n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹
CEN.PG085	PROP.PG140		200	2	364,34	1	389,43	1	391,21	1	375,33	1	352,25	1	321,16
CEN.PG086	PROP.PG141		300	2	407,54	2	436,61	1	436,66	1	422,81	1	399,89	1	368,98
CEN.PG087	PROP.PG142		400	3	398,92	2	441,83	1	455,14	1	445,15	1	423,72	1	392,89
CEN.PG088	PROP.PG143	Composição III	500	3	422,50	2	453,99	2	466,21	1	456,26	1	435,73	1	407,23
CEN.PG089	PROP.PG144	FT Cometa + FT Abyara	600	4	417,13	3	450,78	2	467,93	2	461,78	1	444,69	1	415,84
CEN.PG090	PROP.PG145		700	4	422,34	3	450,10	2	463,46	2	466,53	2	449,46	1	421,18
CEN.PG091	PROP.PG146		800	5	420,49	4	453,14	2	467,22	2	468,67	2	453,05	1	426,61
CEN.PG092	PROP.PG147		900	5	428,65	4	458,00	3	466,97	2	467,17	2	452,65	2	428,93
CEN.PG093	PROP.PG148		1000	6	428,26	4	460,21	3	471,89	2	465,96	2	455,76	2	432,49
CEN.PG094	PROP.PG152		200	3	315,76	2	393,86	1	439,91	1	445,49	1	442,88	1	432,34
CEN.PG095	PROP.PG153		300	4	339,62	3	415,44	2	467,19	2	484,1	1	486,26	1	477,67
CEN.PG096	PROP.PG154		400	5	348,70	4	417,77	2	473,78	2	497,73	1	406,54	1	500,34
CEN.PG097	PROP.PG155	Composição IV	500	6	363,33	4	429,30	3	479,96	2	496,92	2	510,81	1	512,81
CEN.PG098	PROP.PG156	FT Cometa	600	7	367,34	5	436,06	3	479,38	3	496,34	2	507,98	2	510,76
CEN.PG099	PROP.PG157		700	8	366,11	6	436,06	4	480,6	3	499,15	2	506,76	2	509,27
CEN.PG100	PROP.PG158		800	9	369,5	7	437,47	4	489,26	3	505,51	2	510,78	2	508,85
CEN.PG101	PROP.PG159		900	10	372,76	8	441,06	5	487,85	4	508,57	3	512,04	2	511,05
CEN.PG102	PROP.PG160		1000	11	377,67	8	445,63	5	491,80	4	509,89	3	518,14	2	513,94

C – Tabela 3 (Continuação)

Cenário	Propriedade (Origem)	Descrição	Área (ha)	Prazo Calendário (dias)											
				10		15		20		25		30		35	
				n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹	n	US\$.ha ⁻¹
CEN.PG103	PROP.PG161	Composição V FT Abyara	200	2	312,21	1	336,53	1	320,41	1	294,76	1	254,12	1	209,95
CEN.PG104	PROP.PG162		300	2	306,57	2	364,74	1	364,64	1	340,01	1	301,33	1	257,25
CEN.PG105	PROP.PG163		400	3	326,26	2	360,51	2	368,44	1	358,4	1	232,53	1	279,49
CEN.PG106	PROP.PG164		500	4	341,45	2	368,1	2	372,95	2	363,79	1	334,59	1	293,96
CEN.PG107	PROP.PG165		600	4	339,38	3	375,05	2	369,35	2	357,94	2	334,42	2	296,06
CEN.PG108	PROP.PG166		700	5	341,12	3	380,03	3	374,86	2	358,61	2	331,05	2	298,37
CEN.PG109	PROP.PG167		800	6	347,36	4	380,23	3	380,41	2	363,34	2	331,34	2	297,96
CENP.G134	PROP.PG192		900	6	344,7	4	380,38	3	383,47	3	366,41	2	334,08	2	296,38
CEN.PG135	PROP.PG193		1000	7	353,83	4	385,58	3	387,05	3	370,56	2	336,85	2	298,51

D - Requisitos para instalar e executar o programa 1

Requisitos mínimos para executar o programa:

Hardware: Processador 233Mhz; 32MB de memória RAM; 24MB de memória livre disponível no HD.

Software:- Microsoft ® Windows 98 ou XP.

Procedimentos para instalar o programa:

1. Abrir o conteúdo do CD;
2. Entrar na pasta 'Setup' e executar o assistente de instalação 'Setup.exe';
3. O assistente vai auxiliar na instalação do programa que, por padrão, será instalado no diretório 'C:\Arquivos de programas\Jackson\UNICAMP\Tese de Doutorado\'. O assistente também permite a alteração desse diretório;
4. Aguardar enquanto os arquivos são copiados para o seu computador;
5. Finalizada a instalação, será possível executar o programa e/ou visualizar o seu código fonte.

E - Rotina para visualização dos códigos fonte do programa 1

É necessário a instalação prévia do Borland® Delphi 5.0 ou versão superior para visualizar corretamente os códigos fonte do programa. Também é necessário que o programa já esteja instalado no computador.

1. Abrir o diretório principal no qual o programa foi instalado. Por padrão, ele deverá ter sido instalado no diretório 'C:\Arquivos de programas\Jackson\UNICAMP\Tese de Doutorado\';
2. Acessar a pasta 'Códigos Fonte\' e procurar o arquivo 'PPrograma.dpr'. Clicar duas vezes sobre ele e aguarde o Borland® Delphi carregar.
3. Sendo programa é aberto, também é possível visualizar e compilar os códigos fonte abrindo os arquivos (extensão '.dpr' e '.pas') no bloco de notas.

F - Módulo 1: Módulo Principal (Arquivo PPrograma.dpr)

G - Código Fonte 1: PPrograma.dpr <<código>>

```
program PPrograma;

uses
  Forms,
  USimulacao in 'USimulacao.pas' {FrmSimul},
  UPropriedadeNovo in 'UPropriedadeNovo.pas' {FrmPropNovo},
  UPropriedadeAlterar in 'UPropriedadeAlterar.pas' {FrmPropAlterar},
  UCenarioNovo in 'UCenarioNovo.pas' {FrmCenNovo},
  UCenarioAlterar in 'UCenarioAlterar.pas' {FrmCenAlterar},
  URegiao in 'URegiao.pas' {FrmReg},
  URegiaoNovo in 'URegiaoNovo.pas' {FrmRegNovo},
  URegiaoAlterar in 'URegiaoAlterar.pas' {FrmRegAlterar},
  URegiaoExcluir in 'URegiaoExcluir.pas' {FrmRegExcluir},
  UMaquina in 'UMaquina.pas' {FrmMaq},
  UMaquinaNovo in 'UMaquinaNovo.pas' {FrmMaqNovo},
  UMaquinaAlterar in 'UMaquinaAlterar.pas' {FrmMaqAlterar},
  UMaquinaExcluir in 'UMaquinaExcluir.pas' {FrmMaqExcluir},
  UCultivar in 'UCultivar.pas' {FrmCultivar},
  USplash in 'USplash.pas' {FrmSplash},
  UIndicador in 'UIndicador.pas' {FrmIndicador},
  UCaixaDeEntrada in 'UCaixaDeEntrada.pas' {FrmCaixaDeEntrada},
  UExpExcel in 'UExpExcel.pas' {FrmExpExcel},
  UVisualizarImg in 'UVisualizarImg.pas' {FrmVisualizarImg};

{$R *.RES}

var atraso: Integer;

begin
  Application.Initialize;
  with TFrmSplash.Create (Application) do
    try
      Show;
      Update;
      Application.Title := 'Pontualidade na Seleção de Colhedoras de Soja (V 2.0)';
      with ProgressBarSplash do
        begin
          Application.CreateForm(TFrmSimul, FrmSimul);
          StepBy (5);
          Application.CreateForm(TFrmPropNovo, FrmPropNovo);
          StepBy (5);
          Application.CreateForm(TFrmPropAlterar, FrmPropAlterar);
          StepBy (5);
          Application.CreateForm(TFrmCenNovo, FrmCenNovo);
          StepBy (5);
          Application.CreateForm(TFrmCenAlterar, FrmCenAlterar);
          StepBy (5);
          Application.CreateForm(TFrmReg, FrmReg);
          StepBy (5);
          Application.CreateForm(TFrmRegNovo, FrmRegNovo);
```

```

        StepBy (5);
        Application.CreateForm(TFrmRegAlterar, FrmRegAlterar);
        StepBy (5);
        Application.CreateForm(TFrmRegExcluir, FrmRegExcluir);
        StepBy (5);
        Application.CreateForm(TFrmMaq, FrmMaq);
        StepBy (5);
        Application.CreateForm(TFrmMaqNovo, FrmMaqNovo);
        StepBy (5);
        Application.CreateForm(TFrmMaqAlterar, FrmMaqAlterar);
        StepBy (5);
        Application.CreateForm(TFrmMaqExcluir, FrmMaqExcluir);
        StepBy (5);
        Application.CreateForm(TFrmCultivar, FrmCultivar);
        StepBy (5);
    end;
    for atraso := 0 to 1000000000 do;
    finally
        Close;
        Free;
    end;
    Application.Run;
end.

```

F - Módulo 2: Módulo de Apresentação (Arquivo USplash.pas)

G - Código Fonte 2: USplash.pas <<código>>

```

unit USplash;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    ExtCtrls, ComCtrls;

type
    TFrmSplash = class(TForm)
        ImgSplash: TImage;
        ProgressBarSplash: TProgressBar;
        procedure FormCreate(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    FrmSplash: TFrmSplash;

implementation

{$R *.DFM}

```

```

procedure TFrmSplash.FormCreate(Sender: TObject);
begin
  ProgressBarSplash.Brush.Color := clSilver;
end;

end.

```

F - Módulo 3: Módulo de Simulação (Arquivo USimulação.pas)

G - Código Fonte 3: USimulacao.pas <<código>>

```

unit USimulacao;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Grids, DBGrids, ExtCtrls, StdCtrls, Buttons, DBCtrls, ComCtrls, Menus,
  Db, DBTables, Mask, ImgList;

type
  TFrmSimul = class(TForm)
    Lbl1: TLabel;
    GrpBoxPesquisa: TGroupBox;
    Lbl2: TLabel;
    Lbl3: TLabel;
    TxtCod: TEdit;
    DBLookupCbo1: TDBLookupComboBox;
    BtnPesquisar: TBitBtn;
    Pnl2: TPanel;
    Pnl1: TPanel;
    BtnNovo: TBitBtn;
    BtnAlterar: TBitBtn;
    BtnExcluir: TBitBtn;
    StatusBarSimul: TStatusBar;
    MainMenuSimul: TMainMenu;
    MenuPrograma: TMenuItem;
    OpSair: TMenuItem;
    MenuExibir: TMenuItem;
    OpRegiao: TMenuItem;
    OpColhedora: TMenuItem;
    OpCultivar: TMenuItem;
    BtnSair: TBitBtn;
    MenuOpcao: TMenuItem;
    OpCenario: TMenuItem;
    OpPropriedade: TMenuItem;
    DataSourcePropriedade: TDataSource;
    TblProp: TTable;
    QryProp: TQuery;
    QryPropCen: TQuery;
    DataSourceSimulacao: TDataSource;
    QrySimul: TQuery;
    QrySimulTal: TQuery;
    QrySimulCol: TQuery;
  end;

```

```

QryResultPropAno: TQuery;
QryResultMaqAno: TQuery;
QrySimulInd: TQuery;
DataSourceProp: TDataSource;
DataSourcePropCen: TDataSource;
QryPrec: TQuery;
TblSimul: TTable;
TblSimulCdSimulao: TStringField;
TblSimulDataDeCricao: TDateField;
TblSimulTurnoDeTrabalho: TFloatField;
TblSimulPrecipitaoCrtica: TFloatField;
TblSimulNmeroDeTalhes: TSmallintField;
DBGridSimul: TDBGrid;
Pnl3: TPanel;
Lbl4: TLabel;
DBTxt1: TDBEdit;
Lbl5: TLabel;
DBTxt2: TDBEdit;
Lbl6: TLabel;
DBTxt3: TDBEdit;
Lbl7: TLabel;
DBTxt4: TDBEdit;
DataSourceSimul: TDataSource;
DataSourceSimulCol: TDataSource;
DataSourceResultPropAno: TDataSource;
DataSourceResultMaqAno: TDataSource;
ImgListSimul: TImageList;
procedure FormShow(Sender: TObject);
procedure BtnPesquisarClick(Sender: TObject);
procedure BtnNovoClick(Sender: TObject);
procedure BtnAlterarClick(Sender: TObject);
procedure BtnExcluirClick(Sender: TObject);
procedure BtnSairClick(Sender: TObject);
procedure OpRegiaoClick(Sender: TObject);
procedure OpColhedoraClick(Sender: TObject);
procedure OpCultivarClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure OpCenarioClick(Sender: TObject);
procedure OpPropriedadeClick(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
function PesquisarPropriedade: Boolean;
function PesquisarCenario: Boolean;
procedure ExcluirSimulacao (codSimul: String);
end;

var
FrmSimul: TFrmSimul;
OpSimul: Integer;
PrecSaco, Juros, VidaUtil, Eficiencia, Salario, SegAloj, Diesel, Residual: Real;
const letra: array[1..26] of Char = ('A','B','C','D','E','F','G','H','I',
                                     'J','K','L','M','N','O','P','Q','R',
                                     'S','T','U','V','W','X','Y','Z');

```

```

const variavel: array[0..6] of String = ('Perda total (US$/ha)', 'Renda líquida (US$/ha)', 'Custo de
colheita (US$/ha)',
                                         'Custo médio (US$/sc)', 'Custo de serviço mecanizado (US$/ha)',
                                         'Tempo de utilização anual (h)', 'Número de colhedoras');
const atributo: array[0..4] of String = ('Variável', 'Menor valor', 'Maior valor', 'Média', 'Desvio padrão');

```

implementation

```

uses UCenarioNovo, UCenarioAlterar, UPropriedadeNovo, UPropriedadeAlterar, URegiao,
    UMaquina, UCultivar;

```

```

{$R *.DFM}

```

```

procedure TFrmSimul.FormShow(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    try
        TxtCod.Clear;
        case OpSimul of
        0: if (TblProp.RecordCount > 0) then
            begin
                with TblProp do
                begin
                    First;
                    Edit;
                    DBLookupCbo1.KeyValue := FieldValues['CódPropriedade'];
                end;
                PesquisarCenario;
            end;
        1: if (FrmReg.TblReg.RecordCount > 0) then
            begin
                with FrmReg.TblReg do
                begin
                    First;
                    Edit;
                    DBLookupCbo1.KeyValue := FieldValues['CódRegião'];
                end;
                PesquisarPropriedade;
            end;
        end;
    finally
        Screen.Cursor := crDefault;
    end;
end;

```

```

procedure TFrmSimul.BtnPesquisarClick(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    try
        case OpSimul of
        0: if (TblProp.RecordCount > 0) then
            if PesquisarCenario then
                with DBLookupCbo1 do
                begin
                    TblProp.Edit;
                    KeyValue := QryPropCen.FieldValues['CódPropriedade'];
                end;
            end;
        end;
    end;
end;

```

```

        end
    else
        MessageDlg ('Não há nenhum cenário que corresponda às descrições fornecidas.',
            MTInformation, [MbOK], 0);
1: if (FrmReg.TblReg.RecordCount > 0) then
    if PesquisarPropriedade then
        with DBLookupCbo1 do
            begin
                FrmReg.TblReg.Edit;
                KeyValue := QryProp.FieldValues['CódRegião'];
            end
        else
            MessageDlg ('Não há nenhuma propriedade que corresponda às descrições fornecidas.',
                MTInformation, [MbOK], 0)
        end;
    finally
        Screen.Cursor := crDefault;
    end;
end;

procedure TFrmSimul.BtnNovoClick(Sender: TObject);
var numReg: Integer;
    msg: String;
begin
    case OpSimul of
        0: begin
            numReg := TblProp.RecordCount;
            msg := 'Deve haver no mínimo uma propriedade simulada';
            end;
        1: begin
            numReg := FrmReg.TblReg.RecordCount;
            msg := 'Deve haver no mínimo uma região cadastrada';
            end;
    end;
    if Not (numReg > 0) then
        raise Exception.Create (msg)
    else if (FrmMaq.TblCol.RecordCount < 2) then
        raise Exception.Create ('Deve haver no mínimo duas colhedoras cadastradas')
    else if (OpSimul = 0) then
        FrmCenNovo.Show
    else
        FrmPropNovo.Show;
    end;

procedure TFrmSimul.BtnAlterarClick(Sender: TObject);
var numReg: Integer;
    obj: String;
begin
    case OpSimul of
        0: begin
            numReg := QryPropCen.RecordCount;
            obj := 'cenário';
            end;
        1: begin
            numReg := QryProp.RecordCount;
            obj := 'a propriedade';

```

```

    end;
end;
if Not (numReg > 0) then
    raise Exception.Create ('Não há nenhum' + obj + ' na lista')
else if (FrmMaq.TblCol.RecordCount < 2) then
    raise Exception.Create ('Deve haver no mínimo duas colhedoras cadastradas');
if (OpSimul = 0) then
    FrmCenAlterar.Show
else
    FrmPropAlterar.Show;
end;

procedure TFrmSimul.BtnExcluirClick(Sender: TObject);
var numReg, i: Integer;
    obj: String;
begin
    Screen.Cursor := crHourglass;
    try
        case OpSimul of
            0: begin
                numReg := QryPropCen.RecordCount;
                obj := 'o cenário';
                i := 2;
            end;
            1: begin
                numReg := QryProp.RecordCount;
                obj := 'a propriedade';
                i := 1;
            end;
        end;
        if Not (numReg > 0) then
            raise Exception.Create ('Não há nenhum' + Copy (obj, i, Length (obj) - (i - 1)) + ' na lista')
        else if (MessageDlg ('Deseja excluir ' + obj + '?', MTConfirmation, [MbYes,MbNo], 0) = mrYes) then
            begin
                case OpSimul of
                    0: begin
                        ExcluirSimulacao (QryPropCen.FieldValues['CódCenário']);
                        QryPropCen.Delete;
                    end;
                    1: begin
                        with QryPropCen do
                            begin
                                Close;
                                SQL.Clear;
                                SQL.Add ('select *');
                                SQL.Add ('from "PROPRIIDADE_Cenário"');
                                SQL.Add ('where CódPropriedade like "' + QryProp.FieldValues['CódPropriedade'] + '"');
                                Open;
                                FindFirst;
                            end;
                        if QryPropCen.Found then
                            repeat
                                ExcluirSimulacao (QryPropCen.FieldValues['CódCenário']);
                                QryPropCen.Delete;
                            until QryPropCen.Eof;
                        ExcluirSimulacao (QryProp.FieldValues['CódPropriedade']);
                    end;
                end;
            end;
        end;
    end;
end;

```

```

        QryProp.Delete;
    end;
end;
QryProp.Refresh;
QryPropCen.Refresh;
MessageDlg (UpperCase (Copy (obj, 1, 1)) + Copy (obj, 2, Length (obj) - 1) + ' foi excluíd' + Copy
(obj, 1, 1) + ' com êxito.',
            MTInformation, [MbOK], 0);
end;
finally
    Screen.Cursor := crDefault;
end;
end;

procedure TFrmSimul.BtnSairClick(Sender: TObject);
begin
    Application.Terminate;
end;

procedure TFrmSimul.OpRegiaoClick(Sender: TObject);
begin
    Hide;
    FrmReg.Show;
end;

procedure TFrmSimul.OpColhedoraClick(Sender: TObject);
begin
    Hide;
    FrmMaq.Show;
end;

procedure TFrmSimul.OpCultivarClick(Sender: TObject);
begin
    Hide;
    FrmCultivar.Show;
end;

procedure TFrmSimul.ExcluirSimulacao(codSimul: String);
begin
    with QrySimul do
    begin
        Close;
        SQL.Clear;
        SQL.Add ('select *');
        SQL.Add ('from "SIMULAÇÃO"');
        SQL.Add ('where CódSimulação like "' + codSimul + '"');
        Open;
    end;
    with QryResultMaqAno do
    begin
        Close;
        SQL.Clear;
        SQL.Add ('delete from "RESULTADO_MaqAno"');
        SQL.Add ('where CódSimulação like "' + codSimul + '"');
        ExecSQL;
        SQL.Clear;
    end;
end;

```

```

    SQL.Add ('select *');
    SQL.Add ('from "RESULTADO_MaqAno"');
    Open;
end;
with QryResultPropAno do
begin
    Close;
    SQL.Clear;
    SQL.Add ('delete from "RESULTADO_PropAno"');
    SQL.Add ('where CódSimulação like "' + codSimul + '"');
    ExecSQL;
    SQL.Clear;
    SQL.Add ('select *');
    SQL.Add ('from "RESULTADO_PropAno"');
    Open;
end;
with QrySimulCol do
begin
    Close;
    SQL.Clear;
    SQL.Add ('delete from "SIMULAÇÃO_Colhedora"');
    SQL.Add ('where CódSimulação like "' + codSimul + '"');
    ExecSQL;
    SQL.Clear;
    SQL.Add ('select *');
    SQL.Add ('from "SIMULAÇÃO_Colhedora"');
    Open;
end;
with QrySimulTal do
begin
    Close;
    SQL.Clear;
    SQL.Add ('delete from "SIMULAÇÃO_Talhão"');
    SQL.Add ('where CódSimulação like "' + codSimul + '"');
    ExecSQL;
    SQL.Clear;
    SQL.Add ('select *');
    SQL.Add ('from "SIMULAÇÃO_Talhão"');
    Open;
end;
QrySimulInd.Delete;
QrySimul.Delete;
end;

function TFrmSimul.PesquisarPropriedade: Boolean;
begin
    if (TxtCod.Text = '') then
        with QryProp do
            begin
                Close;
                SQL.Clear;
                SQL.Add ('select *');
                SQL.Add ('from "PROPRIIDADE"');
                SQL.Add ('where CódRegião = ' + IntToStr (DBLookupCbo1.KeyValue));
                Open;
            end
        end
    end
end

```

```

else
  with QryProp do
  begin
    Close;
    SQL.Clear;
    SQL.Add ('select *');
    SQL.Add ('from "PROPRIEDADE"');
    SQL.Add ('where CódPropriedade like "' + TxtCod.Text + '"');
    Open;
  end;
  QryProp.FindFirst;
  PesquisarPropriedade := QryProp.Found;
end;

function TFrmSimul.PesquisarCenario: Boolean;
begin
  if (TxtCod.Text = "") then
    with QryPropCen do
    begin
      Close;
      SQL.Clear;
      SQL.Add ('select *');
      SQL.Add ('from "PROPRIEDADE_Cenário"');
      SQL.Add ('where CódPropriedade like "' + DBLookupCbo1.KeyValue + '"');
      Open;
    end
  else
    with QryPropCen do
    begin
      Close;
      SQL.Clear;
      SQL.Add ('select *');
      SQL.Add ('from "PROPRIEDADE_Cenário"');
      SQL.Add ('where CódCenário like "' + TxtCod.Text + '"');
      Open;
    end;
    QryPropCen.FindFirst;
    PesquisarCenario := QryPropCen.Found;
  end;

procedure TFrmSimul.FormCreate(Sender: TObject);
begin
  OpSimul := 0;
end;

procedure TFrmSimul.OpCenarioClick(Sender: TObject);
begin
  Screen.Cursor := crHourglass;
  try
    if Not (OpCenario.Checked) then
      begin
        Hide;
        OpSimul := 0;
        Lbl2.Hint := 'Código do cenário (12 caracteres)';
        Lbl3.Hint := 'Propriedade à qual o cenário está vinculado';
        Lbl3.Caption := 'Propriedade:';
      end
    else
      begin
        Show;
        OpSimul := 1;
      end
  except
    ;
  end;
end;

```

```

Lbl3.Left := 331;
Lbl4.Hint := 'Data em que o cenário foi criado (dd/mm/aa)';
Lbl7.Hint := 'Número de talhões do cenário';
BtnPesquisar.Hint := 'Realizar a pesquisa de cenários';
BtnNovo.Hint := 'Criar um novo cenário';
BtnAlterar.Hint := 'Alterar o cenário';
BtnExcluir.Hint := 'Excluir o cenário do sistema';
TxtCod.Clear;
TxtCod.MaxLength := 12;
with DBGridSimul do
begin
  Hint := 'Lista de cenários';
  DataSource := DataSourcePropCen;
  Columns.Items[0].FieldName := 'CódCenário';
  Columns.Items[0].Title.Caption := 'Código do cenário';
end;
with TblSimul do
begin
  MasterFields := '';
  MasterSource := DataSourcePropCen;
  MasterFields := 'CódCenário';
end;
with DBLookupCbo1 do
begin
  Width := 120;
  Left := 401;
  KeyField := '';
  ListField := '';
  ListSource := DataSourcePropriedade;
  KeyField := 'CódPropriedade';
  ListField := 'CódPropriedade';
  if (TblProp.RecordCount > 0) then
  begin
    TblProp.First;
    TblProp.Edit;
    KeyValue := TblProp.FieldValues['CódPropriedade'];
    PesquisarCenario;
  end;
end;
OpCenario.Checked := true;
end;
finally
  Screen.Cursor := crDefault;
  Show;
end;
end;

procedure TFrmSimul.OpPropriedadeClick(Sender: TObject);
begin
  Screen.Cursor := crHourglass;
  try
    if Not (OpPropriedade.Checked) then
    begin
      Hide;
      OpSimul := 1;
      Lbl2.Hint := 'Código da propriedade (13 caracteres)';
    end;
  end;
end;

```

```

Lbl3.Hint := 'Região agroclimática';
Lbl3.Caption := 'Região: ';
Lbl3.Left := 399;
Lbl4.Hint := 'Data em que a propriedade foi criada (dd/mm/aa)';
Lbl7.Hint := 'Número de talhões da propriedade';
BtnPesquisar.Hint := 'Realizar a pesquisa de propriedades';
BtnNovo.Hint := 'Criar uma nova propriedade';
BtnAlterar.Hint := 'Alterar a propriedade';
BtnExcluir.Hint := 'Excluir a propriedade do sistema';
TxtCod.Clear;
TxtCod.MaxLength := 13;
with DBGridSimul do
begin
  Hint := 'Lista de propriedades';
  DataSource := DataSourceProp;
  Columns.Items[0].FieldName := 'CódPropriedade';
  Columns.Items[0].Title.Caption := 'Código da propriedade';
end;
with TblSimul do
begin
  MasterFields := '';
  MasterSource := DataSourceProp;
  MasterFields := 'CódPropriedade';
end;
with DBLookupCbo1 do
begin
  Width := 75;
  Left := 446;
  KeyField := '';
  ListField := '';
  ListSource := FrmReg.DataSourceRegiao;
  KeyField := 'CódRegião';
  ListField := 'Sigla';
  if (FrmReg.TblReg.RecordCount > 0) then
  begin
    FrmReg.TblReg.First;
    FrmReg.TblReg.Edit;
    KeyValue := FrmReg.TblReg.FieldValues['CódRegião'];
    PesquisarPropriedade;
  end;
end;
OpPropriedade.Checked := true;
end;
finally
  Screen.Cursor := crDefault;
  Show;
end;
end;
end.

```

F - Módulo 4: Módulo Cadastro de Máquinas (Arquivos UMaquina.pas, UMaquinaNovo.pas, UMaquinaAlterar.pas, UMaquinaExcluir.pas)

G - Código Fonte 4: UMaquina.pas <<código>>

```

unit UMaquina;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, DBCtrls, Grids, DBGrids, Db, DBTables, ComCtrls,
  ExtCtrls;

type
  TFrmMaq = class(TForm)
    Lbl1: TLabel;
    GrpBoxPesquisa: TGroupBox;
    Lbl2: TLabel;
    Lbl3: TLabel;
    TxtMarca: TEdit;
    TxtModelo: TEdit;
    BtnPesquisar: TBitBtn;
    DBGridCol: TDBGrid;
    DBImgCol: TDBImage;
    DataSourceCol: TDataSource;
    QryCol: TQuery;
    DataSourceColhedora: TDataSource;
    TblCol: TTable;
    Pnl1: TPanel;
    BtnNovo: TBitBtn;
    BtnAlterar: TBitBtn;
    BtnExcluir: TBitBtn;
    BtnFechar: TBitBtn;
    Pnl2: TPanel;
    StatusBarCol: TStatusBar;
    procedure BtnFecharClick(Sender: TObject);
    procedure BtnPesquisarClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure BtnNovoClick(Sender: TObject);
    procedure QryColAfterRefresh(DataSet: TDataSet);
    procedure BtnAlterarClick(Sender: TObject);
    procedure BtnExcluirClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure DBImgColClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FrmMaq: TFrmMaq;
  codigoMaq, opMaq: Integer;

implementation

uses USimulacao, UMaquinaNovo, UMaquinaAlterar, UMaquinaExcluir, UVisualizarImg;

```

```
{$R *.DFM}
```

```
procedure TFrmMaq.BtnFecharClick(Sender: TObject);  
begin  
    Close;  
end;
```

```
procedure TFrmMaq.BtnPesquisarClick(Sender: TObject);  
begin  
    Screen.Cursor := crHourglass;  
    try  
        if ((TxtMarca.Text = "") and (TxtModelo.Text = "")) then  
            with QryCol do  
                begin  
                    Close;  
                    SQL.Clear;  
                    SQL.Add ('select *');  
                    SQL.Add ('from "COLHEDORA"');  
                    Open;  
                end  
            else  
                if (TxtModelo.Text = "") then  
                    with QryCol do  
                        begin  
                            Close;  
                            SQL.Clear;  
                            SQL.Add ('select *');  
                            SQL.Add ('from "COLHEDORA"');  
                            SQL.Add ('where Marca like "' + TxtMarca.Text + '"');  
                            Open;  
                        end  
                    else if (TxtMarca.Text = "") then  
                        with QryCol do  
                            begin  
                                Close;  
                                SQL.Clear;  
                                SQL.Add ('select *');  
                                SQL.Add ('from "COLHEDORA"');  
                                SQL.Add ('where Modelo like "' + TxtModelo.Text + '"');  
                                Open;  
                            end  
                        else  
                            with QryCol do  
                                begin  
                                    Close;  
                                    SQL.Clear;  
                                    SQL.Add ('select *');  
                                    SQL.Add ('from "COLHEDORA"');  
                                    SQL.Add ('where Marca like "' + TxtMarca.Text + '"');  
                                    SQL.Add ('and Modelo like "' + TxtModelo.Text + '"');  
                                    Open;  
                                end;  
                            QryCol.FindFirst;  
                            if ((TblCol.RecordCount > 0) and Not (QryCol.Found)) then  
                                MessageDlg ('Não há nenhuma colhedora que corresponda às descrições fornecidas.',  
                                    MTInformation, [MbOK], 0);
```

```

finally
    Screen.Cursor := crDefault;
end;
end;

procedure TFrmMaq.FormShow(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    try
        TxtMarca.Clear;
        TxtModelo.Clear;
        opMaq := 0;
        with QryCol do
            begin
                Close;
                SQL.Clear;
                SQL.Add ('select *');
                SQL.Add ('from "COLHEDORA"');
                Open;
            end;
        finally
            Screen.Cursor := crDefault;
        end;
    end;

procedure TFrmMaq.FormCreate(Sender: TObject);
begin
    if (TblCol.RecordCount > 0) then
        begin
            TblCol.Last;
            codigoMaq := TblCol.FieldValues ['CódColhedora'] + 1;
        end
    else
        codigoMaq := 1;
    end;

procedure TFrmMaq.BtnNovoClick(Sender: TObject);
begin
    FrmMaqNovo.Show;
end;

procedure TFrmMaq.QryColAfterRefresh(DataSet: TDataSet);
begin
    case opMaq of
        0: begin
            codigoMaq := codigoMaq + 1;
            FrmMaqNovo.Close;
            MessageDlg ('A colhedora foi incluída com êxito.', MTInformation, [MbOK], 0);
            end;
        1: begin
            FrmMaqAlterar.Close;
            MessageDlg ('A colhedora foi alterada com êxito.', MTInformation, [MbOK], 0);
            end;
        2: begin
            FrmMaqExcluir.Close;
            MessageDlg ('A colhedora foi excluída com êxito.', MTInformation, [MbOK], 0);
            end;
    end;
end;

```

```

        end;
    end;
    TblCol.Refresh;
end;

procedure TFrmMaq.BtnAlterarClick(Sender: TObject);
begin
    if Not (QryCol.RecordCount > 0) then
        raise Exception.Create ('Não há nenhuma colhedora na lista');
    FrmMaqAlterar.Show;
end;

procedure TFrmMaq.BtnExcluirClick(Sender: TObject);
begin
    if Not (QryCol.RecordCount > 0) then
        raise Exception.Create ('Não há nenhuma colhedora na lista');
    FrmMaqExcluir.Show;
end;

procedure TFrmMaq.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    FrmSimul.Show;
end;

procedure TFrmMaq.DBImgColClick(Sender: TObject);
begin
    if Not (QryCol.RecordCount > 0) then
        raise Exception.Create ('Não há nenhuma colhedora na lista');
    try
        FrmVisualizarImg := TFrmVisualizarImg.Create (Application);
        with FrmVisualizarImg do
            begin
                HorzScrollBar.Range := DBImgCol.Picture.Width;
                VertScrollBar.Range := DBImgCol.Picture.Height;
                ImgVisualizar.Picture := DBImgCol.Picture;
                Caption := QryCol.FieldValues ['Marca'] + ' - ' + QryCol.FieldValues ['Modelo'];
                ShowModal;
            end;
        finally
            FrmVisualizarImg.Free;
        end;
    end;
end;

end.

```

G - Código Fonte 5: UMaquinaNovo.pas <<código>>

```

unit UMaquinaNovo;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    StdCtrls, ExtCtrls, Buttons, ExtDlgs, ComCtrls;

```

```

type
  TFrmMaqNovo = class(TForm)
    Lbl1: TLabel;
    GrpBoxMaq: TGroupBox;
    Lbl2: TLabel;
    Lbl3: TLabel;
    Lbl4: TLabel;
    Lbl5: TLabel;
    Lbl6: TLabel;
    TxtMarca: TEdit;
    TxtModelo: TEdit;
    TxtPlat: TEdit;
    TxtPot: TEdit;
    TxtPreco: TEdit;
    Pnl2: TPanel;
    Lbl7: TLabel;
    ImgMaq: TImage;
    BtnProcurar: TBitBtn;
    OpenPictureDialog1: TOpenPictureDialog;
    Pnl1: TPanel;
    BtnConfirmar: TBitBtn;
    BtnCancelar: TBitBtn;
    StatusBarMaqNovo: TStatusBar;
    procedure BtnProcurarClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure BtnCancelarClick(Sender: TObject);
    procedure BtnConfirmarClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FrmMaqNovo: TFrmMaqNovo;

implementation

uses UMaquina;

{$R *.DFM}

procedure TFrmMaqNovo.BtnProcurarClick(Sender: TObject);
begin
  if OpenPictureDialog1.Execute then
    ImgMaq.Picture.LoadFromFile (OpenPictureDialog1.FileName);
end;

procedure TFrmMaqNovo.FormShow(Sender: TObject);
begin
  TxtMarca.Clear;
  TxtModelo.Clear;
  TxtPot.Clear;
  TxtPlat.Clear;
  TxtPreco.Clear;
  ImgMaq.Picture := nil;

```

```

end;

procedure TFrmMaqNovo.BtnCancelarClick(Sender: TObject);
begin
    Close;
end;

procedure TFrmMaqNovo.BtnConfirmarClick(Sender: TObject);
var pot, larg, prec: Real;
begin
    Screen.Cursor := crHourglass;
    try try
        if (TxtMarca.Text = "") then
            raise Exception.Create ('Preencha o campo "Marca".')
        else if (TxtModelo.Text = "") then
            raise Exception.Create ('Preencha o campo "Modelo".')
        else if (TxtPot.Text = "") then
            raise Exception.Create ('Preencha o campo "Potência".')
        else if (TxtPlat.Text = "") then
            raise Exception.Create ('Preencha o campo "Plataforma".')
        else if (TxtPreco.Text = "") then
            raise Exception.Create ('Preencha o campo "Preço".')
        else if ImgMaq.Picture.Bitmap.Empty then
            raise Exception.Create ('Selecione a imagem');
        pot := StrToFloat (TxtPot.Text);
        larg := StrToFloat (TxtPlat.Text);
        prec := StrToFloat (TxtPreco.Text);
        FrmMaq.QryCol.InsertRecord ([codigoMaq, TxtMarca.Text, TxtModelo.Text, pot,
            larg, prec, ImgMaq.Picture]);

        opMaq := 0;
        FrmMaq.QryCol.Refresh;
    finally
        Screen.Cursor := crDefault;
    end;
    except
        on EConvertError do
            raise Exception.Create ('O valor do campo é inválido');
        end;
    end;
end;

end.

```

G - Código Fonte 6: UMaquinaAlterar.pas <<código>>

```

unit UMaquinaAlterar;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    ExtDlgs, Db, DBTables, ExtCtrls, DBCtrls, StdCtrls, Buttons, ComCtrls;

type
    TFrmMaqAlterar = class(TForm)
        Lbl1: TLabel;

```

```

GrpBoxMaq: TGroupBox;
Lbl2: TLabel;
Lbl3: TLabel;
Lbl4: TLabel;
Lbl5: TLabel;
Lbl6: TLabel;
TxtMarca: TEdit;
TxtModelo: TEdit;
TxtPlat: TEdit;
TxtPot: TEdit;
TxtPreco: TEdit;
Pnl2: TPanel;
Lbl7: TLabel;
ImgMaq: TImage;
BtnProcurar: TBitBtn;
OpenPictureDialog1: TOpenPictureDialog;
Pnl1: TPanel;
BtnConfirmar: TBitBtn;
BtnCancelar: TBitBtn;
StatusBarRegAlterar: TStatusBar;
procedure BtnProcurarClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure BtnCancelarClick(Sender: TObject);
procedure BtnConfirmarClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
{ Private declarations }
imagem: TDBImage;
public
{ Public declarations }
end;

var
  FrmMaqAlterar: TFrmMaqAlterar;

implementation

uses UMaquina;

{$R *.DFM}

procedure TFrmMaqAlterar.BtnProcurarClick(Sender: TObject);
begin
  if OpenPictureDialog1.Execute then
    ImgMaq.Picture.LoadFromFile (OpenPictureDialog1.FileName);
end;

procedure TFrmMaqAlterar.FormShow(Sender: TObject);
begin
  TxtMarca.Text := FrmMaq.QryCol.FieldValues['Marca'];
  TxtModelo.Text := FrmMaq.QryCol.FieldValues['Modelo'];
  TxtPot.Text := FrmMaq.QryCol.FieldValues['Potência'];
  TxtPlat.Text := FrmMaq.QryCol.FieldValues['LarguraDePlataforma'];
  TxtPreco.Text := FrmMaq.QryCol.FieldValues['PreçoDeAquisição'];
  ImgMaq.Picture := imagem.Picture;
end;

```

```

procedure TFrmMaqAlterar.BtnCancelarClick(Sender: TObject);
begin
  Close;
end;

procedure TFrmMaqAlterar.BtnConfirmarClick(Sender: TObject);
var pot, larg, prec: Real;
begin
  Screen.Cursor := crHourglass;
  try try
    if (TxtMarca.Text = "") then
      raise Exception.Create ('Preencha o campo "Marca".')
    else if (TxtModelo.Text = "") then
      raise Exception.Create ('Preencha o campo "Modelo".')
    else if (TxtPot.Text = "") then
      raise Exception.Create ('Preencha o campo "Potência".')
    else if (TxtPlat.Text = "") then
      raise Exception.Create ('Preencha o campo "Plataforma".')
    else if (TxtPreco.Text = "") then
      raise Exception.Create ('Preencha o campo "Preço".')
    else if ImgMaq.Picture.Bitmap.Empty then
      raise Exception.Create ('Selecione a imagem');
    pot := StrToFloat (TxtPot.Text);
    larg := StrToFloat (TxtPlat.Text);
    prec := StrToFloat (TxtPreco.Text);
    FrmMaq.QryCol.Edit;
  begin
    FrmMaq.QryCol.FieldValues['Marca'] := TxtMarca.Text;
    FrmMaq.QryCol.FieldValues['Modelo'] := TxtModelo.Text;
    FrmMaq.QryCol.FieldValues['Potência'] := pot;
    FrmMaq.QryCol.FieldValues['LarguraDePlataforma'] := larg;
    FrmMaq.QryCol.FieldValues['PreçoDeAquisição'] := prec;
    imagem.Picture := ImgMaq.Picture;
  end;
  opMaq := 1;
  FrmMaq.QryCol.Refresh;
  finally
    Screen.Cursor := crDefault;
  end;
  except
    on EConvertError do
      raise Exception.Create ('O valor do campo é inválido');
    end;
  end;
end;

procedure TFrmMaqAlterar.FormCreate(Sender: TObject);
begin
  imagem := TDBImage.Create (self);
  imagem.DataSource := FrmMaq.DataSourceCol;
  imagem.DataField := 'Imagem';
end;

end.

```

G - Código Fonte 7 : UMaquinaExcluir.pas <<código>>

```

unit UMaquinaExcluir;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtDlgs, Db, DBTables, ExtCtrls, DBCtrls, StdCtrls, Buttons, ComCtrls;

type
  TFrmMaqExcluir = class(TForm)
    Lbl1: TLabel;
    GrpBoxMaq: TGroupBox;
    Lbl2: TLabel;
    Lbl3: TLabel;
    Lbl4: TLabel;
    Lbl5: TLabel;
    Lbl6: TLabel;
    TxtMarca: TEdit;
    TxtModelo: TEdit;
    TxtPlat: TEdit;
    TxtPot: TEdit;
    TxtPreco: TEdit;
    Pnl2: TPanel;
    Lbl7: TLabel;
    ImgMaq: TImage;
    Pnl1: TPanel;
    BtnConfirmar: TBitBtn;
    BtnCancelar: TBitBtn;
    StatusBarMaqExcluir: TStatusBar;
    procedure FormShow(Sender: TObject);
    procedure BtnCancelarClick(Sender: TObject);
    procedure BtnConfirmarClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
    imagem: TDBImage;
  public
    { Public declarations }
  end;

var
  FrmMaqExcluir: TFrmMaqExcluir;

implementation

uses UMaquina, USimulacao;

{$R *.DFM}

procedure TFrmMaqExcluir.FormShow(Sender: TObject);
begin
  TxtMarca.Text := FrmMaq.QryCol.FieldValues['Marca'];
  TxtModelo.Text := FrmMaq.QryCol.FieldValues['Modelo'];
  TxtPot.Text := FrmMaq.QryCol.FieldValues['Potência'];
  TxtPlat.Text := FrmMaq.QryCol.FieldValues['LarguraDePlataforma'];

```

```

    TxtPreco.Text := FrmMq.QryCol.FieldValues['PreçoDeAquisição'];
    ImgMq.Picture := imagem.Picture;
end;

procedure TFrmMqExcluir.BtnCancelarClick(Sender: TObject);
begin
    Close;
end;

procedure TFrmMqExcluir.BtnConfirmarClick(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    try
        with FrmSimul.QrySimulCol do
            begin
                Close;
                SQL.Clear;
                SQL.Add ('select *');
                SQL.Add ('from "SIMULAÇÃO_Colhedora"');
                SQL.Add ('where CódColhedora = ' + IntToStr (FrmMq.QryCol.FieldValues['CódColhedora']));
                Open;
                FindFirst;
            end;
        if FrmSimul.QrySimulCol.Found then
            repeat
                with FrmSimul.QryResultMqAno do
                    begin
                        Close;
                        SQL.Clear;
                        SQL.Add ('delete from "RESULTADO_MqAno"');
                        SQL.Add ('where CódSimulação like "' + FrmSimul.QrySimulCol.FieldValues['CódSimulação'] +
                            '"');
                        SQL.Add ('and NumColhedora = ' + IntToStr
(FrmSimul.QrySimulCol.FieldValues['NumColhedora']));
                        ExecSQL;
                    end;
                FrmSimul.QrySimulCol.Delete;
            until FrmSimul.QrySimulCol.Eof;
            with FrmSimul.QryResultMqAno do
                begin
                    SQL.Clear;
                    SQL.Add ('select *');
                    SQL.Add ('from "RESULTADO_MqAno"');
                    Open;
                end;
            FrmMq.QryCol.Delete;
            opMq := 2;
            FrmMq.QryCol.Refresh;
        finally
            Screen.Cursor := crDefault;
        end;
    end;

procedure TFrmMqExcluir.FormCreate(Sender: TObject);
begin
    imagem := TDBImage.Create (self);

```

```

    imagem.DataSource := FrmMaq.DataSourceCol;
    imagem.DataField := 'Imagem';
end;

```

end.

F - Módulo 5: Módulo Cadastro de Regiões (Arquivos URegiao.pas, URegiaoNovo.pas, URegiaoAlterar.pas, URegiaoExcluir.pas)

G - Código Fonte 8: URegiao.pas <<código>>

unit URegiao;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, DBTables, Db, DBCtrls, Grids, DBGrids, StdCtrls, Buttons, ComCtrls, Mask, ExtCtrls;

type

```

TFrmReg = class(TForm)
    Lbl1: TLabel;
    DataSourceEstado: TDataSource;
    TblEstado: TTable;
    QryReg: TQuery;
    DataSourceReg: TDataSource;
    PgeCtrlReg: TPageControl;
    TabShtRegiao: TTabSheet;
    GrpBoxPesquisa: TGroupBox;
    Lbl3: TLabel;
    Lbl2: TLabel;
    TxtNome: TEdit;
    DBLookupCboEstado: TDBLookupComboBox;
    BtnPesquisar: TBitBtn;
    DBGridReg: TDBGrid;
    DBImgReg: TDBImage;
    TabShtSerieTemporal: TTabSheet;
    GrpBoxSerieTemporal: TGroupBox;
    Lbl4: TLabel;
    DBLookupCboReg: TDBLookupComboBox;
    Lbl5: TLabel;
    Lbl6: TLabel;
    MskTxtInicio: TMaskEdit;
    MskTxtFim: TMaskEdit;
    BtnConfirmar: TBitBtn;
    DataSourcePrec: TDataSource;
    QryPrec: TQuery;
    DBGridPrec: TDBGrid;
    Pnl5: TPanel;
    DBLblData: TDBText;
    Lbl7: TLabel;
    DBTxtPrec: TDBEdit;
    DBNavigatorPrec: TDBNavigator;

```

```

DataSourceRegiao: TDataSource;
TblReg: TTable;
Pnl1: TPanel;
BtnFechar: TBitBtn;
Pnl2: TPanel;
BtnNovo: TBitBtn;
BtnAlterar: TBitBtn;
BtnExcluir: TBitBtn;
Pnl3: TPanel;
StatusBarReg: TStatusBar;
Pnl4: TPanel;
procedure BtnPesquisarClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure BtnNovoClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure BtnAlterarClick(Sender: TObject);
procedure QryRegAfterRefresh(DataSet: TDataSet);
procedure BtnFecharClick(Sender: TObject);
procedure TabShtSerieTemporalShow(Sender: TObject);
procedure DBLookupCboRegClick(Sender: TObject);
procedure BtnConfirmarClick(Sender: TObject);
procedure BtnExcluirClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure DBImgRegClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FrmReg: TFrmReg;
  codigoReg, opReg: Integer;

implementation

uses USimulacao, URegiaoNovo, URegiaoAlterar, URegiaoExcluir, UVisualizarImg;

{$R *.DFM}

procedure TFrmReg.BtnPesquisarClick(Sender: TObject);
begin
  Screen.Cursor := crHourglass;
  try
    if (TxtNome.Text = '') then
      with QryReg do
        begin
          Close;
          SQL.Clear;
          SQL.Add ('select *');
          SQL.Add ('from "REGIÃO"');
          SQL.Add ('where CódEstado = ' + IntToStr (DBLookupCboEstado.KeyValue));
          Open;
        end
      else
        with QryReg do

```

```

begin
  Close;
  SQL.Clear;
  SQL.Add ('select *');
  SQL.Add ('from "REGIÃO"');
  SQL.Add ('where NomeRegião like ' + TxtNome.Text + '");
  SQL.Add ('and CódEstado = ' + IntToStr (DBLookupCboEstado.KeyValue));
  Open;
end;
QryReg.FindFirst;
if Not (QryReg.Found) then
  MessageDlg ('Não há nenhuma região que corresponda às descrições fornecidas.',
    MTInformation, [MbOK], 0);
finally
  Screen.Cursor := crDefault;
end;
end;

procedure TFrmReg.FormShow(Sender: TObject);
begin
  Screen.Cursor := crHourglass;
  try
    TxtNome.Clear;
    TblEstado.Edit;
    DBLookupCboEstado.KeyValue := 1;
    PgeCtrlReg.ActivePageIndex := 0;
    opReg := 0;
    with QryReg do
      begin
        Close;
        SQL.Clear;
        SQL.Add ('select * from "REGIÃO"');
        SQL.Add ('where CódEstado = ' + IntToStr (DBLookupCboEstado.KeyValue));
        Open;
      end
    finally
      Screen.Cursor := crDefault;
    end;
  end;

procedure TFrmReg.BtnNovoClick(Sender: TObject);
begin
  FrmRegNovo.Show;
end;

procedure TFrmReg.FormCreate(Sender: TObject);
begin
  if (TblReg.RecordCount > 0) then
    begin
      TblReg.Last;
      codigoReg := TblReg.FieldValues ['CódRegião'] + 1;
    end
  else
    codigoReg := 1;
  end;
end;

```

```

procedure TFrmReg.BtnAlterarClick(Sender: TObject);
begin
  if Not (QryReg.RecordCount > 0) then
    raise Exception.Create ('Não há nenhuma região na lista');
  FrmRegAlterar.Show;
end;

procedure TFrmReg.QryRegAfterRefresh(DataSet: TDataSet);
begin
  case opReg of
    0: begin
      codigoReg := codigoReg + 1;
      FrmRegNovo.Close;
      MessageDlg ('A região foi incluída com êxito.', MTInformation, [MbOK], 0);
    end;
    1: begin
      FrmRegAlterar.Close;
      MessageDlg ('A região foi alterada com êxito.', MTInformation, [MbOK], 0);
    end;
    2: begin
      FrmRegExcluir.Close;
      MessageDlg ('A região foi excluída com êxito.', MTInformation, [MbOK], 0);
    end;
  end;
  TblReg.Refresh;
end;

procedure TFrmReg.BtnFecharClick(Sender: TObject);
begin
  Close;
end;

procedure TFrmReg.TabShtSerieTemporalShow(Sender: TObject);
var cod: Integer;
begin
  MskTxtInicio.Clear;
  MskTxtFim.Clear;
  if (TblReg.RecordCount > 0) then
    begin
      if (QryReg.RecordCount > 0) then
        cod := QryReg.FieldValues['CódRegião']
      else
        begin
          TblReg.First;
          cod := TblReg.FieldValues['CódRegião'];
        end;
      TblReg.Edit;
      DBLookupCboReg.KeyValue := cod;
      DBLookupCboRegClick (self);
    end;
end;

procedure TFrmReg.DBLookupCboRegClick(Sender: TObject);
begin
  Screen.Cursor := crHourGlass;
  try

```

```

if (TblReg.RecordCount > 0) then
  with QryPrec do
    begin
      Close;
      SQL.Clear;
      SQL.Add ('select *');
      SQL.Add ('from "REGIÃO_Precipitação"');
      SQL.Add ('where CódRegião = ' + IntToStr (DBLookupCboReg.KeyValue));
      Open;
      Last;
    end;
  finally
    Screen.Cursor := crDefault;
  end;
end;

procedure TFrmReg.BtnConfirmarClick(Sender: TObject);
var prec: String;
    priData, ultData: TDateTime;
    continua: Boolean;
begin
  try try
    if Not (TblReg.RecordCount > 0) then
      raise Exception.Create ('Não há nenhuma região cadastrada');
    priData := StrToDate (MskTxtInicio.Text);
    ultData := StrToDate (MskTxtFim.Text);
  except
    on EConvertError do
      raise Exception.Create ('O valor do campo é inválido');
    end;
    if (priData > ultData) then
      raise Exception.Create ('O período de cadastramento é inválido');
    repeat
      prec := '0';
      continua := InputQuery ('Precipitação ocorrida', DateToStr (priData), prec);
      if continua then
        QryPrec.InsertRecord ([DBLookupCboReg.KeyValue, priData, StrToFloat (prec)])
      else
        Abort;
        priData := priData + 1;
      until (priData > ultData);
    except
      on EDatabaseError do
        raise Exception.Create ('Esta data já foi registrada');
      on EConvertError do
        raise Exception.Create ('O valor fornecido é inválido');
      end;
    end;
end;

procedure TFrmReg.BtnExcluirClick(Sender: TObject);
begin
  if Not (QryReg.RecordCount > 0) then
    raise Exception.Create ('Não há nenhuma região na lista');
  FrmRegExcluir.Show;
end;

```

```

procedure TFrmReg.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    FrmSimul.Show;
end;

procedure TFrmReg.DBImgRegClick(Sender: TObject);
begin
    if Not (QryReg.RecordCount > 0) then
        raise Exception.Create ('Não há nenhuma região na lista');
    try
        FrmVisualizarImg := TFrmVisualizarImg.Create (Application);
        with FrmVisualizarImg do
            begin
                HorzScrollBar.Range := DBImgReg.Picture.Width;
                VertScrollBar.Range := DBImgReg.Picture.Height;
                ImgVisualizar.Picture := DBImgReg.Picture;
                Caption := FrmReg.DBLookupCboEstado.Text + ' - ' +
FrmReg.QryReg.FieldValues['NomeRegião'];
                ShowModal;
            end;
        finally
            FrmVisualizarImg.Free;
        end;
    end;
end.

```

G - Código Fonte 9: URegiaoNovo.pas <<código>>

```

unit URegiaoNovo;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    StdCtrls, DBCtrls, ExtCtrls, Buttons, ComCtrls, Db, DBTables, ExtDlgs,
    Mask;

type
    TFrmRegNovo = class(TForm)
        Lbl1: TLabel;
        OpenPictureDialog1: TOpenPictureDialog;
        GrpBoxReg: TGroupBox;
        Lbl3: TLabel;
        Lbl2: TLabel;
        Lbl4: TLabel;
        Lbl6: TLabel;
        Lbl5: TLabel;
        TxtNome: TEdit;
        DBLookupCboEstado: TDBLookupComboBox;
        TxtSigla: TEdit;
        Pnl2: TPanel;
        ImgReg: TImage;
    end;

```

```

    BtnProcurar: TBitBtn;
    TxtTurno: TEdit;
    Pnl1: TPanel;
    BtnCancelar: TBitBtn;
    BtnConfirmar: TBitBtn;
    StatusBarRegNovo: TStatusBar;
    GrpBoxPer: TGroupBox;
    Lbl7: TLabel;
    Lbl8: TLabel;
    MskTxtPriData: TMaskEdit;
    MskTxtUltData: TMaskEdit;
    procedure BtnProcurarClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure BtnConfirmarClick(Sender: TObject);
    procedure BtnCancelarClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    FrmRegNovo: TFrmRegNovo;

implementation

uses URegiao;

{$R *.DFM}

procedure TFrmRegNovo.BtnProcurarClick(Sender: TObject);
begin
    if OpenPictureDialog1.Execute then
        ImgReg.Picture.LoadFromFile (OpenPictureDialog1.FileName);
end;

procedure TFrmRegNovo.FormShow(Sender: TObject);
begin
    FrmReg.TblEstado.Edit;
    DBLookupCboEstado.KeyValue := 1;
    TxtNome.Clear;
    TxtSigla.Clear;
    TxtTurno.Clear;
    MskTxtPriData.Clear;
    MskTxtUltData.Clear;
    ImgReg.Picture := nil;
end;

procedure TFrmRegNovo.BtnConfirmarClick(Sender: TObject);
var turno: Real;
    priData, ultData: TDateTime;
begin
    Screen.Cursor := crHourglass;
    try try
        if (TxtNome.Text = "") then
            raise Exception.Create ('Preencha o campo "Nome".')

```

```

else if (TxtSigla.Text = "") then
    raise Exception.Create ('Preencha o campo "Sigla".')
else if (TxtTurno.Text = "") then
    raise Exception.Create ('Preencha o campo "Turno".')
else if (MskTxtPriData.Text = ' / ') then
    raise Exception.Create ('Preencha o campo "Primeira data".')
else if (MskTxtUltData.Text = ' / ') then
    raise Exception.Create ('Preencha o campo "Última data".')
else if ImgReg.Picture.Bitmap.Empty then
    raise Exception.Create ('Selecione a imagem');
turno := StrToFloat (TxtTurno.Text);
priData := StrToDate (MskTxtPriData.Text);
ultData := StrToDate (MskTxtUltData.Text);
if (priData > ultData) then
    raise Exception.Create ('O período de semeadura é inválido');
FrmReg.QryReg.InsertRecord ([codigoReg, DBLookupCboEstado.KeyValue, TxtNome.Text,
    TxtSigla.Text, turno, priData, ultData, ImgReg.Picture]);
opReg := 0;
FrmReg.QryReg.Refresh;
finally
    Screen.Cursor := crDefault;
end;
except
    on EConvertError do
        raise Exception.Create ('O valor do campo é inválido');
    end;
end;

procedure TFrmRegNovo.BtnCancelarClick(Sender: TObject);
begin
    Close;
end;

end.

```

G - Código Fonte 10: URegiaoAlterar.pas <<código>>

```

unit URegiaoAlterar;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    ExtDlgs, Db, DBTables, ExtCtrls, DBCtrls, StdCtrls, Buttons, ComCtrls,
    Mask;

type
    TFrmRegAlterar = class(TForm)
        Lbl1: TLabel;
        GrpBoxReg: TGroupBox;
        Lbl3: TLabel;
        Lbl2: TLabel;
        Lbl4: TLabel;
        Lbl6: TLabel;

```

```

Lbl5: TLabel;
TxtNome: TEdit;
DBLookupCboEstado: TDBLookupComboBox;
TxtSigla: TEdit;
Pnl2: TPanel;
ImgReg: TImage;
TxtTurno: TEdit;
OpenPictureDialog1: TOpenPictureDialog;
BntProcurar: TBitBtn;
Pnl1: TPanel;
BntConfirmar: TBitBtn;
BntCancelar: TBitBtn;
StatusBarRegAlterar: TStatusBar;
GrpBoxPer: TGroupBox;
Lbl7: TLabel;
Lbl8: TLabel;
MskTxtPriData: TMaskEdit;
MskTxtUltData: TMaskEdit;
procedure FormShow(Sender: TObject);
procedure BntProcurarClick(Sender: TObject);
procedure BntConfirmarClick(Sender: TObject);
procedure BntCancelarClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
{ Private declarations }
imagem: TDBImage;
public
{ Public declarations }
end;

var
FrmRegAlterar: TFrmRegAlterar;

implementation

uses URegiao;

{$R *.DFM}

procedure TFrmRegAlterar.FormShow(Sender: TObject);
begin
FrmReg.TblEstado.Edit;
DBLookupCboEstado.KeyValue := FrmReg.QryReg.FieldValues['CódEstado'];
TxtNome.Text := FrmReg.QryReg.FieldValues['NomeRegião'];
TxtSigla.Text := FrmReg.QryReg.FieldValues['Sigla'];
TxtTurno.Text := FrmReg.QryReg.FieldValues['TurnoDeTrabalho'];
MskTxtPriData.Text := FormatDateTime ('dd/mm',
FrmReg.QryReg.FieldValues['PriDataDeSemeadura']);
MskTxtUltData.Text := FormatDateTime ('dd/mm',
FrmReg.QryReg.FieldValues['UltDataDeSemeadura']);
ImgReg.Picture := imagem.Picture;
end;

procedure TFrmRegAlterar.BntProcurarClick(Sender: TObject);
begin
if OpenPictureDialog1.Execute then

```

```

    ImgReg.Picture.LoadFromFile (OpenPictureDialog1.FileName);
end;

procedure TFrmRegAlterar.BtnConfirmarClick(Sender: TObject);
var turno: Real;
    priData, ultData: TDateTime;
begin
    Screen.Cursor := crHourglass;
    try try
        if (TxtNome.Text = "") then
            raise Exception.Create ('Preencha o campo "Nome".')
        else if (TxtSigla.Text = "") then
            raise Exception.Create ('Preencha o campo "Sigla".')
        else if (TxtTurno.Text = "") then
            raise Exception.Create ('Preencha o campo "Turno".')
        else if (MskTxtPriData.Text = ' / ') then
            raise Exception.Create ('Preencha o campo "Primeira data".')
        else if (MskTxtUltData.Text = ' / ') then
            raise Exception.Create ('Preencha o campo "Última data".')
        else if ImgReg.Picture.Bitmap.Empty then
            raise Exception.Create ('Selecione a imagem');
        turno := StrToFloat (TxtTurno.Text);
        priData := StrToDate (MskTxtPriData.Text);
        ultData := StrToDate (MskTxtUltData.Text);
        if (priData > ultData) then
            raise Exception.Create ('O período de semeadura é inválido');
        FrmReg.QryReg.Edit;
        begin
            FrmReg.QryReg.FieldValues['CódEstado'] := DBLookupCboEstado.KeyValue;
            FrmReg.QryReg.FieldValues['NomeRegião'] := TxtNome.Text;
            FrmReg.QryReg.FieldValues['Sigla'] := TxtSigla.Text;
            FrmReg.QryReg.FieldValues['TurnoDeTrabalho'] := turno;
            FrmReg.QryReg.FieldValues['PriDataDeSemeadura'] := priData;
            FrmReg.QryReg.FieldValues['UltDataDeSemeadura'] := ultData;
            imagem.Picture := ImgReg.Picture;
        end;
        opReg := 1;
        FrmReg.QryReg.Refresh;
    finally
        Screen.Cursor := crDefault;
    end;
except
    on EConvertError do
        raise Exception.Create ('O valor do campo é inválido');
    end;
end;

procedure TFrmRegAlterar.BtnCancelarClick(Sender: TObject);
begin
    Close;
end;

procedure TFrmRegAlterar.FormCreate(Sender: TObject);
begin
    imagem := TDBImage.Create (self);
    imagem.DataSource := FrmReg.DataSourceReg;

```

```

    imagem.DataField := 'Imagem';
end;

end.

```

G - Código Fonte 11: URegiaoExcluir.pas <<código>>

```

unit URegiaoExcluir;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Mask, DBCtrls, Buttons, ExtCtrls, ComCtrls;

type
  TFrmRegExcluir = class(TForm)
    Lbl1: TLabel;
    GrpBoxReg: TGroupBox;
    Lbl2: TLabel;
    Lbl3: TLabel;
    Lbl4: TLabel;
    Lbl5: TLabel;
    TxtNome: TEdit;
    TxtSigla: TEdit;
    TxtTurno: TEdit;
    Pnl2: TPanel;
    ImgReg: TImage;
    Pnl1: TPanel;
    BtnConfirmar: TBitBtn;
    BtnCancelar: TBitBtn;
    Lbl6: TLabel;
    StatusBarRegExcluir: TStatusBar;
    GrpBoxPer: TGroupBox;
    Lbl7: TLabel;
    MskTxtPriData: TMaskEdit;
    Lbl8: TLabel;
    MskTxtUltData: TMaskEdit;
    TxtEstado: TEdit;
    procedure FormShow(Sender: TObject);
    procedure BtnCancelarClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure BtnConfirmarClick(Sender: TObject);
  private
    { Private declarations }
    imagem: TDBImage;
  public
    { Public declarations }
  end;

var
  FrmRegExcluir: TFrmRegExcluir;

implementation

```

```

uses URegiao, UCultivar, USimulacao;

{$R *.DFM}

procedure TFrmRegExcluir.FormShow(Sender: TObject);
begin
    FrmReg.TblEstado.FindKey ([FrmReg.QryReg.FieldValues['CódEstado']]);
    TxtEstado.Text := FrmReg.TblEstado.FieldValues['Sigla'];
    TxtNome.Text := FrmReg.QryReg.FieldValues['NomeRegião'];
    TxtSigla.Text := FrmReg.QryReg.FieldValues['Sigla'];
    TxtTurno.Text := FrmReg.QryReg.FieldValues['TurnoDeTrabalho'];
    MskTxtPriData.Text := FormatDateTime ('dd/mm',
FrmReg.QryReg.FieldValues['PriDataDeSemeadura']);
    MskTxtUltData.Text := FormatDateTime ('dd/mm',
FrmReg.QryReg.FieldValues['UltDataDeSemeadura']);
    ImgReg.Picture := imagem.Picture;
end;

procedure TFrmRegExcluir.BtnCancelarClick(Sender: TObject);
begin
    Close;
end;

procedure TFrmRegExcluir.FormCreate(Sender: TObject);
begin
    imagem := TDBImage.Create (self);
    imagem.DataSource := FrmReg.DataSourceReg;
    imagem.DataField := 'Imagem';
end;

procedure TFrmRegExcluir.BtnConfirmarClick(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    try
        with FrmReg.QryPrec do
            begin
                Close;
                SQL.Clear;
                SQL.Add ('delete from "REGIÃO_Precipitação");
                SQL.Add ('where CódRegião = ' + IntToStr (FrmReg.QryReg.FieldValues['CódRegião']));
                ExecSQL;
            end;
        with FrmCultivar.QryRegCul do
            begin
                Close;
                SQL.Clear;
                SQL.Add ('delete from "REGIÃO_CULTIVAR");
                SQL.Add ('where CódRegião = ' + IntToStr (FrmReg.QryReg.FieldValues['CódRegião']));
                ExecSQL;
                SQL.Clear;
                SQL.Add ('select *');
                SQL.Add ('from "REGIÃO_CULTIVAR");
                Open;
            end;
        with FrmSimul.QryProp do

```

```

begin
  Close;
  SQL.Clear;
  SQL.Add ('select *');
  SQL.Add ('from "PROPRIIDADE"');
  SQL.Add ('where CódRegião = ' + IntToStr (FrmReg.QryReg.FieldValues['CódRegião']));
  Open;
  FindFirst;
end;
if FrmSimul.QryProp.Found then
  repeat
    with FrmSimul.QryPropCen do
      begin
        Close;
        SQL.Clear;
        SQL.Add ('select *');
        SQL.Add ('from "PROPRIIDADE_Cenário"');
        SQL.Add ('where CódPropriedade like "' + FrmSimul.QryProp.FieldValues['CódPropriedade'] +
        ""');
        Open;
        FindFirst;
      end;
      if FrmSimul.QryPropCen.Found then
        repeat
          FrmSimul.ExcluirSimulacao (FrmSimul.QryPropCen.FieldValues['CódCenário']);
          FrmSimul.QryPropCen.Delete;
        until FrmSimul.QryPropCen.Eof;
        FrmSimul.ExcluirSimulacao (FrmSimul.QryProp.FieldValues['CódPropriedade']);
        FrmSimul.QryProp.Delete;
      until FrmSimul.QryProp.Eof;
      FrmReg.QryReg.Delete;
      opReg := 2;
      FrmReg.QryReg.Refresh;
    finally
      Screen.Cursor := crDefault;
    end;
  end;
end.
end.

```

F - Módulo 6: Módulo Cadastro de Cultivares (Arquivo UCultivar.pas)

G - Código Fonte 12: UCultivar.pas <<código>>

```

unit UCultivar;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  DBTables, Db, StdCtrls, DBCtrls, Buttons, Mask, ExtCtrls, ComCtrls;

type

```

```

TFrmCultivar = class(TForm)
  GrpBoxPesquisa: TGroupBox;
  Lbl1: TLabel;
  Lbl2: TLabel;
  Lbl3: TLabel;
  DataSourceRegCul: TDataSource;
  QryRegCul: TQuery;
  DataSourceCultivar: TDataSource;
  TblCul: TTable;
  DBLookupCboReg: TDBLookupComboBox;
  DBLookupCboCul: TDBLookupComboBox;
  BtnPesquisar: TBitBtn;
  Pnl3: TPanel;
  Lbl4: TLabel;
  Lbl5: TLabel;
  Lbl6: TLabel;
  Lbl7: TLabel;
  Lbl8: TLabel;
  TxtPerdA: TEdit;
  TxtPerdB: TEdit;
  TxtPerdC: TEdit;
  TxtProdC: TEdit;
  Pnl1: TPanel;
  BtnNovo: TBitBtn;
  BtnAlterar: TBitBtn;
  BtnExcluir: TBitBtn;
  BtnFechar: TBitBtn;
  Pnl2: TPanel;
  StatusBarCul: TStatusBar;
  TxtCiclo: TEdit;
  procedure FormShow(Sender: TObject);
  procedure BtnFecharClick(Sender: TObject);
  procedure BtnPesquisarClick(Sender: TObject);
  procedure BtnNovoClick(Sender: TObject);
  procedure BtnAlterarClick(Sender: TObject);
  procedure QryRegCulAfterRefresh(DataSet: TDataSet);
  procedure BtnExcluirClick(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure DBLookupCboCulClick(Sender: TObject);
  procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
  procedure AtualizarCampos (Sender: TObject);
end;

var
  FrmCultivar: TFrmCultivar;
  opCul: Integer;

implementation

uses USimulacao, URegiao;

{$R *.DFM}

```

```

procedure TFrmCultivar.AtualizarCampos(Sender: TObject);
begin
  if (QryRegCul.Found and Not (opCul = 2)) then
  begin
    TxtCiclo.Text := QryRegCul.FieldValues['CicloDeMaturação'];
    TxtPerdA.Text := QryRegCul.FieldValues['CoeficienteDePerdA'];
    TxtPerdB.Text := QryRegCul.FieldValues['CoeficienteDePerdB'];
    TxtPerdC.Text := QryRegCul.FieldValues['CoeficienteDePerdC'];
    TxtProdC.Text := QryRegCul.FieldValues['CoeficienteDeProdC'];
  end
  else
  begin
    TxtCiclo.Clear;
    TxtPerdA.Clear;
    TxtPerdB.Clear;
    TxtPerdC.Clear;
    TxtProdC.Clear;
  end;
end;

procedure TFrmCultivar.FormShow(Sender: TObject);
begin
  Screen.Cursor := crHourGlass;
  try
    opCul := 0;
    TblCul.Edit;
    DBLookupCboCul.KeyValue := 1;
    if (FrmReg.TblReg.RecordCount > 0) then
    begin
      with FrmReg.TblReg do
      begin
        First;
        Edit;
        DBLookupCboReg.KeyValue := FieldValues['CódRegião'];
      end;
      with QryRegCul do
      begin
        Close;
        SQL.Clear;
        SQL.Add ('select *');
        SQL.Add ('from "REGIÃO_CULTIVAR"');
        SQL.Add ('where CódRegião = ' + IntToStr (DBLookupCboReg.KeyValue));
        SQL.Add ('and CódCultivar = ' + IntToStr (DBLookupCboCul.KeyValue));
        Open;
      end;
    end;
    QryRegCul.FindFirst;
    BtnNovo.Enabled := Not (QryRegCul.Found);
    BtnAlterar.Enabled := QryRegCul.Found;
    BtnExcluir.Enabled := QryRegCul.Found;
    AtualizarCampos (self);
  finally
    Screen.Cursor := crDefault;
  end;
end;

```

```

procedure TFrmCultivar.BtnFecharClick(Sender: TObject);
begin
    Close;
end;

procedure TFrmCultivar.BtnPesquisarClick(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    try
        if (FrmReg.TblReg.RecordCount > 0) then
            begin
                with QryRegCul do
                    begin
                        Close;
                        SQL.Clear;
                        SQL.Add ('select *');
                        SQL.Add ('from "REGIÃO_CULTIVAR"');
                        SQL.Add ('where CódRegião = ' + IntToStr (DBLookupCboReg.KeyValue));
                        SQL.Add ('and CódCultivar = ' + IntToStr (DBLookupCboCul.KeyValue));
                        Open;
                        FindFirst;
                    end;
                BtnNovo.Enabled := Not (QryRegCul.Found);
                BtnAlterar.Enabled := QryRegCul.Found;
                BtnExcluir.Enabled := QryRegCul.Found;
                AtualizarCampos (self);
                if Not (QryRegCul.Found) then
                    MessageDlg ('Não há função de perdas que corresponda às descrições fornecidas.',
                        MTInformation, [MbOK], 0);
            end;
        finally
            Screen.Cursor := crDefault;
        end;
    end;

procedure TFrmCultivar.BtnNovoClick(Sender: TObject);
var ciclo: Integer;
    perdA, perdB, perdC, prodC: Real;
begin
    Screen.Cursor := crHourglass;
    try try
        if Not (FrmReg.TblReg.RecordCount > 0) then
            raise Exception.Create ('Não há nenhuma região cadastrada')
        else if (TxtCiclo.Text = '') then
            raise Exception.Create ('Preencha o campo "Ciclo de maturação".')
        else if (TxtPerdA.Text = '') then
            raise Exception.Create ('Preencha o campo "Coeficiente de Perda A".')
        else if (TxtPerdB.Text = '') then
            raise Exception.Create ('Preencha o campo "Coeficiente de Perda B".')
        else if (TxtPerdC.Text = '') then
            raise Exception.Create ('Preencha o campo "Coeficiente de Perda C".')
        else if (TxtProdC.Text = '') then
            raise Exception.Create ('Preencha o campo "Coeficiente de Prod. C".');
        ciclo := StrToInt (TxtCiclo.Text);
        perdA := StrToFloat (TxtPerdA.Text);
    end;
end;

```

```

perdB := StrToFloat (TxtPerdB.Text);
perdC := StrToFloat (TxtPerdC.Text);
prodC := StrToFloat (TxtProdC.Text);
QryRegCul.InsertRecord ([DBLookupCboReg.KeyValue, DBLookupCboCul.KeyValue,
                        ciclo, perdA, perdB, perdC, prodC]);
opCul := 0;
QryRegCul.Refresh;
finally
    Screen.Cursor := crDefault;
end;
except
    on EConvertError do
        raise Exception.Create ('O valor do campo é inválido');
    end;
end;

procedure TFrmCultivar.BtnAlterarClick(Sender: TObject);
var ciclo: Integer;
    perdA, perdB, perdC, prodC: Real;
begin
    Screen.Cursor := crHourglass;
    try try
        if (TxtCiclo.Text = '') then
            raise Exception.Create ('Preencha o campo "Ciclo de maturação".')
        else if (TxtPerdA.Text = '') then
            raise Exception.Create ('Preencha o campo "Coeficiente de Perda A".')
        else if (TxtPerdB.Text = '') then
            raise Exception.Create ('Preencha o campo "Coeficiente de Perda B".')
        else if (TxtPerdC.Text = '') then
            raise Exception.Create ('Preencha o campo "Coeficiente de Perda C".')
        else if (TxtProdC.Text = '') then
            raise Exception.Create ('Preencha o campo "Coeficiente de Prod. C".');
        ciclo := StrToInt (TxtCiclo.Text);
        perdA := StrToFloat (TxtPerdA.Text);
        perdB := StrToFloat (TxtPerdB.Text);
        perdC := StrToFloat (TxtPerdC.Text);
        prodC := StrToFloat (TxtProdC.Text);
        QryRegCul.Edit;
        begin
            QryRegCul.FieldValues['CicloDeMaturação'] := ciclo;
            QryRegCul.FieldValues['CoeficienteDePerdA'] := perdA;
            QryRegCul.FieldValues['CoeficienteDePerdB'] := perdB;
            QryRegCul.FieldValues['CoeficienteDePerdC'] := perdC;
            QryRegCul.FieldValues['CoeficienteDeProdC'] := prodC;
        end;
        opCul := 1;
        QryRegCul.Refresh;
    finally
        Screen.Cursor := crDefault;
    end;
except
    on EConvertError do
        raise Exception.Create ('O valor do campo é inválido');
    end;
end;

```

```

procedure TFrmCultivar.QryRegCulAfterRefresh(DataSet: TDataSet);
begin
  case opCul of
    0: begin
      BtnNovo.Enabled := Not (BtnNovo.Enabled);
      BtnAlterar.Enabled := Not (BtnAlterar.Enabled);
      BtnExcluir.Enabled := Not (BtnExcluir.Enabled);
      MessageDlg ('A função de perdas foi incluída com êxito.', MTInformation,
        [MbOK], 0);
    end;
    1: MessageDlg ('A função de perdas foi alterada com êxito.', MTInformation,
      [MbOK], 0);
    2: begin
      BtnNovo.Enabled := Not (BtnNovo.Enabled);
      BtnAlterar.Enabled := Not (BtnAlterar.Enabled);
      BtnExcluir.Enabled := Not (BtnExcluir.Enabled);
      AtualizarCampos (self);
      MessageDlg ('A função de perdas foi excluída com êxito.', MTInformation,
        [MbOK], 0);
    end;
  end;
end;

procedure TFrmCultivar.BtnExcluirClick(Sender: TObject);
begin
  if (MessageDlg ('Deseja excluir a função de perdas?', MTConfirmation, [MbYes,MbNo], 0) = mrYes)
  then
    begin
      QryRegCul.Delete;
      opCul := 2;
      QryRegCul.Refresh;
    end;
  end;

procedure TFrmCultivar.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  FrmSimul.Show;
end;

procedure TFrmCultivar.DBLookupCboCulClick(Sender: TObject);
begin
  DBLookupCboCul.Hint := 'Grupo de Maturação ' + TblCul.FieldValues['GrupoDeMaturação'];
end;

procedure TFrmCultivar.FormCreate(Sender: TObject);
begin
  DBLookupCboCulClick (self);
end;

end.

```

F - Módulo 7: Módulo Visualização de Imagens (Arquivo UVisualizarImg.pas)

G - Código Fonte 13: UVisualizarImg.pas <<código>>

```
unit UVisualizarImg;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls;

type
  TFrmVisualizarImg = class(TForm)
    ImgVisualizar: TImage;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FrmVisualizarImg: TFrmVisualizarImg;

implementation

{$R *.DFM}

end.
```

F - Módulo 8: Módulo Geração de Cenários (Arquivos UCenarioNovo.pas, UCenarioAlterar.pas)

G - Código Fonte 14: UCenarioNovo.pas <<código>>

```
unit UCenarioNovo;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Menus, StdCtrls, ComCtrls, ToolWin, Buttons, ExtCtrls, DBCtrls, CheckLst,
  Grids, Mask, DB;

type
  TFrmCenNovo = class(TForm)
    Lbl1: TLabel;
    MainMenuCenNovo: TMainMenu;
    MenuCenario: TMenuItem;
    MenuEditar: TMenuItem;
    OpNovo: TMenuItem;
    OpSalvar: TMenuItem;
    OpIndicador: TMenuItem;
    ToolBarCenNovo: TToolBar;
  end;
```

ToolBtnNovo: TToolButton;
 ToolBtnSalvar: TToolButton;
 Pnl1: TPanel;
 Pnl3: TPanel;
 Pnl2: TPanel;
 BtnConfirmar: TBitBtn;
 BtnCancelar: TBitBtn;
 LblCod: TLabel;
 PgeCtrlCenNovo: TPageControl;
 TabShtCen: TTabSheet;
 TabShtCustoOpt: TTabSheet;
 TabShtCustoMaq: TTabSheet;
 StatusBarCenNovo: TStatusBar;
 Pnl4: TPanel;
 GrpBoxCenCol: TGroupBox;
 Pnl5: TPanel;
 GrpBoxCen: TGroupBox;
 GrpBoxPer: TGroupBox;
 Pnl6: TPanel;
 Lbl2: TLabel;
 Lbl3: TLabel;
 Lbl4: TLabel;
 Lbl5: TLabel;
 Lbl6: TLabel;
 TxtTurno: TEdit;
 TxtPrec: TEdit;
 Pnl7: TPanel;
 ChkListBoxCenCol: TCheckListBox;
 GrpBoxCenTal: TGroupBox;
 Pnl8: TPanel;
 StgCenTal: TStringGrid;
 PopupMenuCenTal: TPopupMenu;
 OpIncluirTal: TMenuItem;
 OpExcluirTal: TMenuItem;
 MenuN1: TMenuItem;
 OpLimparListaTal: TMenuItem;
 DBLookupCboCul: TDBLookupComboBox;
 TxtCell: TMaskEdit;
 StgCustoOpt: TStringGrid;
 StgCustoMaq: TStringGrid;
 PopupMenuGrad1: TPopupMenu;
 OpLocalizarAno1: TMenuItem;
 PopupMenuGrad2: TPopupMenu;
 OpLocalizarAno2: TMenuItem;
 ToolBarCustoOpt: TToolBar;
 ToolBtnLocalizarAno1: TToolButton;
 ToolBarCustoMaq: TToolBar;
 ToolBtnLocalizarAno2: TToolButton;
 ToolBarCenTal: TToolBar;
 ToolBtnIncluirTal: TToolButton;
 ToolBtnExcluirTal: TToolButton;
 ToolN1: TToolButton;
 ToolBtnLimparListaTal: TToolButton;
 DBLookupCboProp: TDBLookupComboBox;
 TxtPriAno: TEdit;
 TxtUltAno: TEdit;

```

TabShtResultFinal: TTabSheet;
StgResultFinalEst: TStringGrid;
PopupMenuGradResultFinal: TPopupMenu;
OpExportar: TMenuItem;
ToolN2: TToolButton;
ToolBtnIndicador: TToolButton;
procedure ChkListBoxCenColClickCheck(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure OpIncluirTalClick(Sender: TObject);
procedure OpExcluirTalClick(Sender: TObject);
procedure OpLimparListaTalClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure DBLookupCboCulClick(Sender: TObject);
procedure DBLookupCboCulExit(Sender: TObject);
procedure StgCenTalSelectCell(Sender: TObject; ACol, ARow: Integer;
    var CanSelect: Boolean);
procedure DBLookupCboPropClick(Sender: TObject);
procedure BtnConfirmarClick(Sender: TObject);
procedure StgCenTalGetEditText(Sender: TObject; ACol, ARow: Integer;
    var Value: String);
procedure TxtCellExit(Sender: TObject);
procedure DBLookupCboCulKeyPress(Sender: TObject; var Key: Char);
procedure TxtCellKeyPress(Sender: TObject; var Key: Char);
procedure OpSalvarClick(Sender: TObject);
procedure OpIndicadorClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure BtnCancelarClick(Sender: TObject);
procedure OpLocalizarAno1Click(Sender: TObject);
procedure OpLocalizarAno2Click(Sender: TObject);
procedure ChkListBoxCenColMouseDown(Sender: TObject;
    Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
procedure ChkListBoxCenColKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
procedure OpNovoClick(Sender: TObject);
procedure OpExportarClick(Sender: TObject);
private
{ Private declarations }
priAno, ultAno, numCol, indCol: Integer;
turnoTrab, precCrit, areaTotalProp, areaTotalCen: Real;
codColhedora: array of array[0..4] of Real;
modColhedora: array of array[0..1] of String;
indColhedora: array of Integer;
talCultivar: array[1..26,0..5] of Real;
linAno: array of array[0..1] of Integer;
detAno: array of array[0..6] of Real;
colAno: array of array of Real;
resultPropAno: array of array of array[0..5] of Real;
resultMaqAno: array of array of array[0..4] of Real;
resultFinalVar: array[0..6] of array[0..3] of Real;
resultFinalTal: array of array[0..1] of array[0..3] of Real;
categoria: array[0..0] of array of array[0..3] of Real;
valCell, codProp, codCen: String;
cenGerado, cenSalvado: Boolean;
public
{ Public declarations }
function PesquisarFuncPerdas (codReg, codCul: Integer): Boolean;

```

```

function GerarCodigo: Boolean;
function SalvarCenario: Boolean;
function LocalizarAno (anoAT, indGrad: Integer): Boolean;
procedure ListarColhedoras (Sender: TObject);
procedure LimparCenario (opCab: Integer);
procedure RecuperarPropriedade (Sender: TObject);
procedure RecuperarPropTal (codReg, codCul: Integer);
procedure IncluirTal (Sender: TObject);
procedure ExcluirTal (Sender: TObject);
procedure LimparListaTal (Sender: TObject);
procedure FuncPerdas (Sender: TObject);
procedure ServMecanizado (Sender: TObject);
procedure ResultadoFinal (numVar: Integer);
procedure DistribuirValores (indVar: Integer; valor: array of Real);
procedure CalcularMedias (opResult, indVar: Integer; valor: array of Real);
end;

var
  FrmCenNovo: TFrmCenNovo;

implementation

uses USimulacao, UIndicador, URegiao, UMaquina, UCultivar, UExpExcel;

{$R *.DFM}

procedure TFrmCenNovo.ListarColhedoras(Sender: TObject);
var cs, c: Integer;
    strModelo: String;
begin
  ChkListBoxCenCol.Items.Clear;
  SetLength (codColhedora, FrmMaq.TblCol.RecordCount);
  SetLength (modColhedora, FrmMaq.TblCol.RecordCount);
  SetLength (indColhedora, numCol);
  FrmMaq.TblCol.First;
  cs := 0;
  c := 0;
  repeat
    with FrmMaq.TblCol do
      begin
        codColhedora[cs,0] := FieldValues['CódColhedora'];
        codColhedora[cs,1] := 0;
        codColhedora[cs,2] := FieldValues['Potência'];
        codColhedora[cs,3] := FieldValues['LarguraDePlataforma'];
        codColhedora[cs,4] := FieldValues['PreçoDeAquisição'];
        modColhedora[cs,0] := FieldValues['Marca'];
        modColhedora[cs,1] := FieldValues['Modelo'];
      end;
    strModelo := modColhedora[cs,1];
    strModelo := strModelo + ' (plataforma de '
    strModelo := strModelo + FloatToStr (codColhedora[cs,3]);
    strModelo := strModelo + ' metros)';
    ChkListBoxCenCol.Items.Add (strModelo);
    if ((numCol > 0) and (c < numCol)) then
      with FrmSimul.QrySimulCol do
        if (codColhedora[cs,0] = FieldValues['CódColhedora']) then

```

```

begin
    codColhedora[cs,1] := FieldValues['VelocidadeDeOperação'];
    indColhedora[c] := cs;
    ChkListBoxCenCol.Checked[cs] := true;
    c := c + 1;
    Next;
end;
cs := cs + 1;
FrmMaq.TblCol.Next;
until FrmMaq.TblCol.Eof;
end;

function TFrmCenNovo.PesquisarFuncPerdas(codReg, codCul: Integer): Boolean;
var ACol, t: Integer;
begin
    PesquisarFuncPerdas := true;
    with FrmCultivar.QryRegCul do
    begin
        Close;
        SQL.Clear;
        SQL.Add ('select *');
        SQL.Add ('from "REGIÃO_CULTIVAR"');
        SQL.Add ('where CódRegião = ' + IntToStr (codReg));
        SQL.Add ('and CódCultivar = ' + IntToStr (codCul));
        Open;
        FindFirst;
    end;
    if Not (FrmCultivar.QryRegCul.Found) then
    begin
        PesquisarFuncPerdas := false;
        for ACol := 4 to 10 do
            StgCenTal.Cells[ACol,StgCenTal.Row] := 'Não encontrado';
        MessageDlg ('Não há nenhuma função de perdas que corresponda às descrições fornecidas.',
            MTInformation, [MbOK], 0);
    end
    else
        with StgCenTal do
        begin
            Cells[4,Row] := FrmCultivar.QryRegCul.FieldValues['CicloDeMaturação'];
            if (Cells[5,Row] <> "") then
                if (Cells[5,Row] = 'Não encontrado') then
                    for ACol := 5 to 10 do
                        Cells[ACol,Row] := "
                    else
                        begin
                            Cells[6,Row] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[5,Row]) + StrToInt
(Cells[4,Row]));
                            Cells[7,Row] := Cells[6,Row];
                            Cells[8,Row] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[7,Row]) + 24);
                            Cells[9,Row] := '0';
                            Cells[10,Row] := '24';
                            t := -1;
                            if ((Row > 1) and (Cells[5,Row - 1] <> "") and (StrToDate (Cells[7,Row]) <= StrToDate
(Cells[8,Row - 1]))) then
                                t := Row - 1

```

```

        else if (Row < (RowCount - 1)) and (Cells[5,Row + 1] <> "") and (StrToDate (Cells[7,Row + 1]) <=
StrToDate (Cells[8,Row])) then
            t := Row + 1;
            if (t >= 0) then
                MessageDlg ('O período de colheita do talhão ' + letra[Row] + ' coincide com o período do
talhão ' + letra[t] + ',
                    MTWarning, [MbOK], 0);
            end;
        end;
    end;
end;

procedure TFrmCenNovo.ChkListBoxCenColClickCheck(Sender: TObject);
var v: Real;
    strVeloc: String;
    valido, continua: Boolean;
begin
    try
        if (indCol = ChkListBoxCenCol.ItemIndex) then
            if ChkListBoxCenCol.Checked[indCol] then
                begin
                    numCol := numCol + 1;
                    strVeloc := FloatToStr (codColhedora[indCol,1]);
                    valido := false;
                    repeat
                        continua := InputQuery (ChkListBoxCenCol.Items.Strings[indCol], 'Velocidade de operação
(km/h)', strVeloc);
                        if continua then
                            begin
                                v := StrToFloat (strVeloc);
                                if ((v < 3) or (v > 7)) then
                                    MessageDlg ('A velocidade de operação é inválida.', MError, [MbOK], 0)
                                else
                                    begin
                                        valido := true;
                                        codColhedora[indCol,1] := v;
                                    end;
                                end;
                            until (valido or Not (continua));
                            if Not (continua) then
                                begin
                                    ChkListBoxCenCol.Checked[indCol] := false;
                                    numCol := numCol - 1;
                                end;
                            end
                        else
                            numCol := numCol - 1
                        else
                            ChkListBoxCenCol.Checked[indCol] := Not (ChkListBoxCenCol.Checked[indCol]);
                    except
                        on EConvertError do
                            begin
                                MessageDlg ('O valor fornecido é inválido.', MError, [MbOK], 0);
                                indCol := ChkListBoxCenCol.ItemIndex;
                                if (codColhedora[indCol,1] = 0) then
                                    begin
                                        ChkListBoxCenCol.Checked[indCol] := false;

```

```

        numCol := numCol - 1;
    end;
end;
end;
end;

procedure TFrmCenNovo.FormShow(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    try
        with FrmSimul.TblProp do
            begin
                First;
                Edit;
                DBLookupCboProp.KeyValue := FieldValues['CódPropriedade'];
            end;
            RecuperarPropriedade (self);
        finally
            Screen.Cursor := crDefault;
        end;
    end;
end;

procedure TFrmCenNovo.IncluirTal(Sender: TObject);
begin
    with StgCenTal do
        begin
            RowCount := RowCount + 1;
            Rows[RowCount - 1].Clear;
            Cells[0, RowCount - 1] := letra[RowCount - 1];
            talCultivar[RowCount - 1, 0] := 0;
            talCultivar[RowCount - 1, 1] := 0;
        end;
    end;
end;

procedure TFrmCenNovo.ExcluirTal(Sender: TObject);
var ARow: Integer;
begin
    with StgCenTal do
        begin
            if (Row < (RowCount - 1)) then
                for ARow := Row to (RowCount - 2) do
                    begin
                        Rows[ARow] := Rows[ARow + 1];
                        Cells[0, ARow] := letra[ARow];
                        talCultivar[ARow] := talCultivar[ARow + 1];
                    end;
                if (RowCount > 2) then
                    RowCount := RowCount - 1
                else
                    begin
                        Rows[1].Clear;
                        Cells[0, 1] := letra[1];
                        talCultivar[1, 0] := 0;
                        talCultivar[1, 1] := 0;
                    end;
            end;
        end;
    end;
end;

```

```

end;

procedure TFrmCenNovo.LimparListaTal(Sender: TObject);
var ARow: Integer;
begin
  with StgCenTal do
    begin
      for ARow := (RowCount - 1) downto 1 do
        Rows[ARow].Clear;
        RowCount := 2;
        Cells[0,1] := letra[1];
        talCultivar[1,0] := 0;
        talCultivar[1,1] := 0;
      end;
    end;
end;

procedure TFrmCenNovo.OpIncluirTalClick(Sender: TObject);
begin
  if (Not (cenGerado) and (StgCenTal.RowCount < 27) and (talCultivar[StgCenTal.RowCount - 1,1] <>
0)) then
    IncluirTal (self);
  end;
end;

procedure TFrmCenNovo.OpExcluirTalClick(Sender: TObject);
begin
  if Not (cenGerado) then
    ExcluirTal (self);
  end;
end;

procedure TFrmCenNovo.OpLimparListaTalClick(Sender: TObject);
begin
  if Not (cenGerado) then
    LimparListaTal (self);
  end;
end;

procedure TFrmCenNovo.FormCreate(Sender: TObject);
begin
  with StgCenTal do
    begin
      Cells[0,0] := 'Talhão';ColWidths[0] := 40;
      Cells[1,0] := 'Cultivar';ColWidths[1] := 127;
      Cells[2,0] := 'Grupo de maturação';ColWidths[2] := 104;
      Cells[3,0] := 'Área de lavoura (ha)';ColWidths[3] := 103;
      Cells[4,0] := 'Ciclo de maturação (dias)';ColWidths[4] := 126;
      Cells[5,0] := 'Data de semeadura';ColWidths[5] := 100;
      Cells[6,0] := 'Data provável de colheita';ColWidths[6] := 129;
      Cells[7,0] := 'Início da colheita';ColWidths[7] := 89;
      Cells[8,0] := 'Término previsto';ColWidths[8] := 85;
      Cells[9,0] := 'Atraso inicial (dias)';ColWidths[9] := 94;
      Cells[10,0] := 'Atraso previsto (dias)';ColWidths[10] := 105;
      DefaultRowHeight := DBLookupCboCul.Height;
    end;
  StgCustoOpt.DefaultRowHeight := StgCenTal.DefaultRowHeight;
  StgCustoMaq.DefaultRowHeight := StgCenTal.DefaultRowHeight;
  with StgResultFinalEst do
    begin

```

```

ColWidths[3] := 72;
ColWidths[4] := 81; {era 76}
DefaultRowHeight := StgCenTal.DefaultRowHeight;
end;
TxtCell.BorderStyle := bsNone;
end;

procedure TFrmCenNovo.LimparCenario(opCab: Integer);
var ARow: Integer;
begin
cenGerado := false;
cenSalvado := false;
LblCod.Caption := "";
for ARow := (StgCustoOpt.RowCount - 1) downto 0 do
    StgCustoOpt.Rows[ARow].Clear;
StgCustoOpt.RowCount := 2;
for ARow := (StgCustoMaq.RowCount - 1) downto 0 do
    StgCustoMaq.Rows[ARow].Clear;
StgCustoMaq.RowCount := 2;
for ARow := (StgResultFinalEst.RowCount - 1) downto 0 do
    StgResultFinalEst.Rows[ARow].Clear;
StgResultFinalEst.RowCount := 5;
StgResultFinalEst.RowCount := 2;
case opCab of
0: begin
    with StgCustoOpt do
    begin
        ColCount := 7;
        Cells[0,0] := 'Ano/Talhão';ColWidths[0] := 148;
        Cells[1,0] := 'Início (dd/mm/aa)';ColWidths[1] := 93;
        Cells[2,0] := 'Término (dd/mm/aa)';ColWidths[2] := 104;
        Cells[3,0] := 'Atraso (dias)';ColWidths[3] := 65;
        Cells[4,0] := 'Nº de dias úteis';ColWidths[4] := 81;
        Cells[5,0] := 'Perda total (sc)';ColWidths[5] := 78;
        Cells[6,0] := 'Renda total (sc)';ColWidths[6] := 82;
    end;
    with StgCustoMaq do
    begin
        ColCount := 6;
        Cells[0,0] := 'Ano/Colhedora/Talhão';ColWidths[0] := 177;
        Cells[1,0] := 'Área colhida (ha)';ColWidths[1] := 87;
        Cells[2,0] := 'Perda total (US$)';ColWidths[2] := 110;
        Cells[3,0] := 'Renda total (US$)';ColWidths[3] := 92;
        Cells[4,0] := 'Custo de serviço mecanizado (US$)';ColWidths[4] := 176;
        Cells[5,0] := 'Custo unitário (US$/sc)';ColWidths[5] := 117;
    end;
    StgResultFinalEst.ColWidths[0] := 136;
end;
1: begin
    with StgCustoOpt do
    begin
        ColCount := 9;
        Cells[0,0] := 'Data (dd/mm/aa)';ColWidths[0] := 148;
        Cells[1,0] := 'Precipitação (mm)';ColWidths[1] := 120;
        Cells[2,0] := 'Situação';ColWidths[2] := 49;
        Cells[3,0] := 'Atraso (dias)';ColWidths[3] := 65;

```

```

Cells[4,0] := 'Perda física diária (sc)';ColWidths[4] := 112;
Cells[5,0] := 'Perda física acumulada (sc)';ColWidths[5] := 139;
Cells[6,0] := 'Renda física diária (sc)';ColWidths[6] := 116;
Cells[7,0] := 'Renda física acumulada (sc)';ColWidths[7] := 143;
Cells[8,0] := 'Área colhida (ha)';ColWidths[8] := 87;
end;
with StgCustoMaq do
begin
ColCount := 7;
Cells[0,0] := 'Ano/Colhedora/Talhão';ColWidths[0] := 177;
Cells[1,0] := 'F.U. produtivo';ColWidths[1] := 87;
Cells[2,0] := 'F.U. improdutivo';ColWidths[2] := 110;
Cells[3,0] := 'Custo produtivo (US$)';ColWidths[3] := 111;
Cells[4,0] := 'Custo improdutivo (US$)';ColWidths[4] := 121;
Cells[5,0] := 'Custo de serviço mecanizado (US$)';ColWidths[5] := 176;
Cells[6,0] := 'Custo unitário (US$/sc)';ColWidths[6] := 117;
end;
StgResultFinalEst.ColWidths[0] := 193;
end;
end;
end;

procedure TFrmCenNovo.RecuperarPropriedade(Sender: TObject);
var t, a, c, numTal, numAnos, aln: Integer;
    capInstal, custoDeSM: Real;
    strData: String[8];
    capColhedora: array of Real;
begin
codProp := DBLookupCboProp.KeyValue;
FrmReg.TblReg.FindKey ([FrmSimul.QryProp.FieldValues['CódRegião']]);
with FrmSimul.QrySimul do
begin
Close;
SQL.Clear;
SQL.Add ('select *');
SQL.Add ('from "SIMULAÇÃO"');
SQL.Add ('where CódSimulação like "' + codProp + '"');
Open;
turnoTrab := FieldValues['TurnoDeTrabalho'];
precCrit := FieldValues['PrecipitaçãoCrítica'];
TxtTurno.Text := FieldValues['TurnoDeTrabalho'];
TxtPrec.Text := FieldValues['PrecipitaçãoCrítica'];
end;
with FrmSimul.QrySimulInd do
begin
PrecSaco := FieldValues['PreçoDoSaco'];
Juros := FieldValues['TaxaDeJuros'];
VidaUtil := FieldValues['VidaÚtil'];
Eficiencia := FieldValues['Eficiência'];
Salario := FieldValues['Salário'];
SegAloj := FieldValues['TaxaDeSeguroAlojamento'];
Diesel := FieldValues['PreçoDoDiesel'];
Residual := FieldValues['ValorResidual'];
end;
with FrmSimul.QrySimulCol do
begin

```

```

Close;
SQL.Clear;
SQL.Add ('select *');
SQL.Add ('from "SIMULAÇÃO_Colhedora"');
SQL.Add ('where CódSimulação like "' + codProp + '"');
Open;
First;
numCol := RecordCount;
end;
ListarColhedoras (self);
with FrmSimul.QrySimulTal do
begin
Close;
SQL.Clear;
SQL.Add ('select *');
SQL.Add ('from "SIMULAÇÃO_Talhão"');
SQL.Add ('where CódSimulação like "' + codProp + '"');
Open;
First;
numTal := RecordCount;
areaTotalProp := 0;
end;
LimparListaTal (self);
with FrmSimul.QryResultPropAno do
begin
Close;
SQL.Clear;
SQL.Add ('select *');
SQL.Add ('from "RESULTADO_PropAno"');
SQL.Add ('where CódSimulação like "' + codProp + '"');
Open;
First;
numAnos := RecordCount div numTal;
end;
LimparCenario (0);
with FrmSimul.QryResultMaqAno do
begin
Close;
SQL.Clear;
SQL.Add ('select *');
SQL.Add ('from "RESULTADO_MaqAno"');
SQL.Add ('where CódSimulação like "' + codProp + '"');
Open;
First;
end;
SetLength (linAno, numAnos);
SetLength (detAno, numAnos);
SetLength (colAno, numAnos);
SetLength (resultPropAno, numTal);
SetLength (resultMaqAno, numTal);
SetLength (resultFinalTal, numTal);
SetLength (capColhedora, numCol);
capInstal := 0;
for t := 1 to numTal do
begin
SetLength (resultPropAno[t - 1], numAnos);

```

```

SetLength (resultMaqAno[t - 1], numAnos);
RecuperarPropTal (FrmReg.TblReg.FieldValues['CódRegião'],
FrmSimul.QrySimulTal.FieldValues['CódCultivar']);
for a := 0 to (numAnos - 1) do
begin
SetLength (resultMaqAno[t - 1,a], numCol);
with FrmSimul.QryResultPropAno do
begin
if ((t - 1) = 0) then
begin
SetLength (colAno[a], numCol);
detAno[a,0] := FieldValues['Ano'];
detAno[a,1] := 1;
detAno[a,2] := 0;
detAno[a,3] := 0;
detAno[a,4] := 0;
detAno[a,5] := 0;
detAno[a,6] := 0;
end;
resultPropAno[t - 1,a,0] := FieldValues['PerdaAcumulada'];
resultPropAno[t - 1,a,1] := FieldValues['ColheitaAcumulada'];
if (talCultivar[t,1] <> 0) then
begin
strData := StgCenTal.Cells[7,t];
System.Insert (Copy (FloatToStr (detAno[a,0]), 3, 4), strData, 7);
resultPropAno[t - 1,a,2] := StrToDate (FormatDateTime ('dd/mm/' + Copy (FloatToStr
(detAno[a,0]), 3, 4), FieldValues['DataDeInícioDaColheita'])) - StrToDate (strData);
aln := Trunc ((StrToDate (StgCenTal.Cells[7,t]) + resultPropAno[t - 1,a,2]) - (StrToDate
(StgCenTal.Cells[6,t]) + 3));
if (aln < 0) then
aln := 0;
end;
resultPropAno[t - 1,a,3] := aln;
resultPropAno[t - 1,a,4] := FieldValues['PrazoDeColheita'];
resultPropAno[t - 1,a,5] := FieldValues['NúmeroDeDiasÚteis'];
Next;
end;
for c := 0 to (numCol - 1) do
with FrmSimul.QryResultMaqAno do
begin
resultMaqAno[t - 1,a,c,0] := FieldValues['NúmeroDeColhedoras'];
resultMaqAno[t - 1,a,c,1] := FieldValues['CustoProdutivo'];
resultMaqAno[t - 1,a,c,2] := FieldValues['CustoUnitário'];
resultMaqAno[t - 1,a,c,3] := FieldValues['ÁreaColhida'];
if ((t - 1) = 0) then
begin
colAno[a,c] := 0;
if (a = 0) then
begin
capColhedora[c] := (codColhedora[indColhedora[c],1] * resultMaqAno[0,a,c,0] *
codColhedora[indColhedora[c],3] * (Eficiencia / 1000)) * turnoTrab;
capInstal := capInstal + capColhedora[c];
end;
end;
Next;
end;
end;

```

```

end;
FrmSimul.QrySimulTal.Next;
end;
StgCenTal.RowCount := StgCenTal.RowCount - 1;
for a := 0 to (numAnos - 1) do
begin
with StgCustoOpt do
begin
linAno[a,0] := RowCount - 1;
Cells[0,RowCount - 1] := 'ANO - ' + FloatToStr (detAno[a,0]);
RowCount := RowCount + 2;
for t := 1 to numTal do
begin
Cells[0,RowCount - 1] := 'Talhão ' + letra[t];
if (talCultivar[t,1] <> 0) then
begin
Cells[1,RowCount - 1] := FormatDateTime ('dd/mm/' + Copy (FloatToStr (detAno[a,0]), 3, 2),
StrToDate (StgCenTal.Cells[7,t]) + resultPropAno[t - 1,a,2]);
Cells[2,RowCount - 1] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[1,RowCount - 1]) +
resultPropAno[t - 1,a,4]);
end
else
begin
Cells[1,RowCount - 1] := 'Não encontrado';
Cells[2,RowCount - 1] := 'Não encontrado';
end;
Cells[3,RowCount - 1] := FloatToStr (resultPropAno[t - 1,a,3] + resultPropAno[t - 1,a,4]);
Cells[4,RowCount - 1] := FloatToStr (resultPropAno[t - 1,a,5]);
Cells[5,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,a,0]]);
Cells[6,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,a,1]]);
detAno[a,2] := detAno[a,2] + resultPropAno[t - 1,a,5] * turnoTrab;
detAno[a,3] := detAno[a,3] + resultPropAno[t - 1,a,0];
detAno[a,4] := detAno[a,4] + resultPropAno[t - 1,a,1];
RowCount := RowCount + 1;
end;
RowCount := RowCount + 5;
Cells[0,RowCount - 6] := 'RESULTADOS';
Cells[0,RowCount - 5] := 'Perda total (US$):';Cells[1,RowCount - 5] := Format ('%n', [detAno[a,3] *
PrecSaco]);
Cells[0,RowCount - 4] := 'Renda total (US$):';Cells[1,RowCount - 4] := Format ('%n', [detAno[a,4] *
PrecSaco]);
Cells[0,RowCount - 3] := 'Tempo de utilização anual (h):';Cells[1,RowCount - 3] := Format ('%n',
[detAno[a,2]]);
end;
with StgCustoMaq do
begin
linAno[a,1] := RowCount - 1;
Cells[0,RowCount - 1] := 'ANO - ' + FloatToStr (detAno[a,0]);
RowCount := RowCount + 2;
for c := 0 to (numCol - 1) do
begin
RowCount := RowCount + 6;
Cells[0,RowCount - 7] := 'Marca: ' + modColhedora[indColhedora[c],0];
Cells[0,RowCount - 6] := 'Modelo: ' + modColhedora[indColhedora[c],1];
Cells[0,RowCount - 5] := 'Plataforma: ' + Format ('%n m', [codColhedora[indColhedora[c],3]]);
Cells[0,RowCount - 4] := 'Velocidade: ' + Format ('%n km/h', [codColhedora[indColhedora[c],1]]);

```

```

Cells[0,RowCount - 3] := 'Número de colhedoras: ' + FloatToStr (resultMaqAno[0,a,c,0]);
Cells[0,RowCount - 2] := 'Capacidade: ' + Format ('%n ha/dia', [capColhedora[c]]);
Cells[0,RowCount - 1] := 'RESULTADOS';
custoDeSM := 0;
for t := 1 to numTal do
begin
    RowCount := RowCount + 1;
    Cells[0,RowCount - 1] := 'Talhão ' + letra[t];
    Cells[1,RowCount - 1] := Format ('%n', [resultMaqAno[t - 1,a,c,3]]);
    Cells[2,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,a,0] * PrecSaco * (capColhedora[c] /
capInstal)]);
    Cells[3,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,a,1] * PrecSaco * (capColhedora[c] /
capInstal)]);
    Cells[4,RowCount - 1] := Format ('%n', [resultMaqAno[t - 1,a,c,1]]);
    Cells[5,RowCount - 1] := Format ('%n', [resultMaqAno[t - 1,a,c,2]]);
    custoDeSM := custoDeSM + resultMaqAno[t - 1,a,c,1];
    colAno[a,c] := colAno[a,c] + resultMaqAno[t - 1,a,c,2];
end;
RowCount := RowCount + 5;
detAno[a,5] := detAno[a,5] + custoDeSM;
colAno[a,c] := colAno[a,c] / numTal;
detAno[a,6] := detAno[a,6] + colAno[a,c];
Cells[0,RowCount - 5] := 'SERVIÇO MECANIZADO (US$)';Cells[1,RowCount - 5] := Format
('%n', [custoDeSM]);
Cells[0,RowCount - 4] := 'CUSTO M. COLHEDORA (US$/sc)';Cells[1,RowCount - 4] := Format
('%n', [colAno[a,c]]);
end;
detAno[a,5] := PrecSaco * (detAno[a,4] - detAno[a,3]) - detAno[a,5];
detAno[a,6] := detAno[a,6] / numCol;
Cells[0,RowCount - 2] := 'RENTA LÍQUIDA (US$)';Cells[1,RowCount - 2] := Format ('%n',
[detAno[a,5]]);
Cells[0,RowCount - 1] := 'CUSTO M. FROTA (US$)';Cells[1,RowCount - 1] := Format ('%n',
[detAno[a,6]]);
RowCount := RowCount + 2;
end;
end;
StgCustoOpt.RowCount := StgCustoOpt.RowCount - 2;
StgCustoMaq.RowCount := StgCustoMaq.RowCount - 2;
areaTotalCen := areaTotalProp;
priAno := Trunc (detAno[0,0]);
ultAno := Trunc (detAno[numAnos - 1,0]);
TxtPriAno.Text := IntToStr (priAno);
TxtUltAno.Text := IntToStr (ultAno);
PgCtrlCenNovo.ActivePageIndex := 0;
ResultadoFinal (4);
end;

procedure TFrmCenNovo.RecuperarPropTal(codReg, codCul: Integer);
var ACol, aln: Integer;
begin
    FrmCultivar.TblCul.FindKey ([codCul]);
    with FrmCultivar.QryRegCul do
    begin
        Close;
        SQL.Clear;
        SQL.Add ('select *');

```

```

SQL.Add ('from "REGIÃO_CULTIVAR"');
SQL.Add ('where CódRegião = ' + IntToStr (codReg));
SQL.Add ('and CódCultivar = ' + IntToStr (codCul));
Open;
FindFirst;
end;
with StgCenTal do
begin
Cells[0,RowCount - 1] := letra[RowCount - 1];
Cells[1,RowCount - 1] := FrmCultivar.TblCul.FieldValues['NomeCultivar'];
Cells[2,RowCount - 1] := FrmCultivar.TblCul.FieldValues['GrupoDeMaturação'];
Cells[3,RowCount - 1] := FrmSimul.QrySimulTal.FieldValues['ÁreaDeLavoura'];
areaTotalProp := areaTotalProp + FrmSimul.QrySimulTal.FieldValues['ÁreaDeLavoura'];
talCultivar[RowCount - 1,0] := 0;
if Not (FrmCultivar.QryRegCul.Found) then
begin
for ACol := 4 to 10 do
Cells[ACol,RowCount - 1] := 'Não encontrado';
MessageDlg ('A função de perdas à qual o talhão ' + letra[RowCount - 1] + ' está vinculado foi
removida.',
MTWarning, [MbOK], 0);
talCultivar[RowCount - 1,1] := 0;
end
else
begin
Cells[4,RowCount - 1] := FrmCultivar.QryRegCul.FieldValues['CicloDeMaturação'];
Cells[5,RowCount - 1] := FormatDateTime ('dd/mm/' + Copy (IntToStr (StrToInt (FormatDateTime
('yyyy', Date)) - 1), 3, 2),
FrmSimul.QrySimulTal.FieldValues['DataDeSemeadura']);
Cells[6,RowCount - 1] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[5,RowCount - 1]) +
StrToInt (Cells[4,RowCount - 1]));
Cells[7,RowCount - 1] := FormatDateTime ('dd/mm/' + FormatDateTime ('yy', Date),
FrmSimul.QrySimulTal.FieldValues['DataDeInícioDaColheita']);
aln := Trunc (StrToDate (Cells[7,RowCount - 1]) - (StrToDate (Cells[6,RowCount - 1]) + 3));
if (aln < 1) then
aln := 0;
Cells[8,RowCount - 1] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[7,RowCount - 1]) + (24 -
aln));
Cells[9,RowCount - 1] := IntToStr (aln);
Cells[10,RowCount - 1] := '24';
talCultivar[RowCount - 1,1] := codCul;
talCultivar[RowCount - 1,2] := FrmCultivar.QryRegCul.FieldValues['CoeficienteDePerdA'];
talCultivar[RowCount - 1,3] := FrmCultivar.QryRegCul.FieldValues['CoeficienteDePerdB'];
talCultivar[RowCount - 1,4] := FrmCultivar.QryRegCul.FieldValues['CoeficienteDePerdC'];
talCultivar[RowCount - 1,5] := FrmCultivar.QryRegCul.FieldValues['CoeficienteDeProdC'];
end;
RowCount := RowCount + 1;
end;
end;

procedure TFrmCenNovo.DBLookupCboCulClick(Sender: TObject);
begin
StgCenTal.Cells[2,StgCenTal.Row] := FrmCultivar.TblCul.FieldValues['GrupoDeMaturação'];
talCultivar[StgCenTal.Row,0] := 1;
if PesquisarFuncPerdas (FrmReg.TblReg.FieldValues['CódRegião'], DBLookupCboCul.KeyValue)
then

```

```

with FrmCultivar.QryRegCul do
begin
    talCultivar[StgCenTal.Row,1] := DBLookupCboCul.KeyValue;
    talCultivar[StgCenTal.Row,2] := FieldValues['CoeficienteDePerdA'];
    talCultivar[StgCenTal.Row,3] := FieldValues['CoeficienteDePerdB'];
    talCultivar[StgCenTal.Row,4] := FieldValues['CoeficienteDePerdC'];
    talCultivar[StgCenTal.Row,5] := FieldValues['CoeficienteDeProdC'];
end
else
    talCultivar[StgCenTal.Row,1] := 0;
end;

procedure TFrmCenNovo.DBLookupCboCulExit(Sender: TObject);
var resp: Word;
    continua: Boolean;
begin
    continua := false;
    if (talCultivar[StgCenTal.Row,0] = 1) then
    begin
        if (talCultivar[StgCenTal.Row,1] <> 0) then
            StgCenTal.Cells[1,StgCenTal.Row] := DBLookupCboCul.Text
        else
            begin
                resp := MessageDlg ('O cultivar é inválido. Pressione OK para selecionar outro ou Cancel para
excluir o talhão.',
                    MTErrror, [MbOK,MbCancel], 0);
                if (resp = mrOK) then
                begin
                    continua := true;
                    DBLookupCboCul.SetFocus;
                end
                else
                    ExcluirTal (self);
                end;
            end;
        if Not (continua) then
        begin
            DBLookupCboCul.Visible := false;
            StgCenTal.SetFocus;
        end;
    end;
end;

procedure TFrmCenNovo.StgCenTalSelectCell(Sender: TObject; ACol,
    ARow: Integer; var CanSelect: Boolean);
var cod: Real;
    R: TRect;
begin
    if (ACol = 1) then
    begin
        R := StgCenTal.CellRect (ACol,ARow);
        R.Left := R.Left + StgCenTal.Left;
        R.Right := R.Right + StgCenTal.Left;
        R.Top := R.Top + StgCenTal.Top;
        R.Bottom := R.Bottom + StgCenTal.Top;
        with DBLookupCboCul do
            begin

```

```

Left := R.Left + 1;
Top := R.Top + 1;
Width := (R.Right + 1) - R.Left;
Height := (R.Bottom + 1) - R.Top;
if (talCultivar[ARow,1] <> 0) then
    cod := talCultivar[ARow,1]
else
    cod := 1;
FrmCultivar.TblCul.Edit;
KeyValue := cod;
Visible := true;
SetFocus;
end;
end;
end;

```

```

procedure TFrmCenNovo.DBLookupCboPropClick(Sender: TObject);
var resp: Word;
begin
    Screen.Cursor := crHourglass;
    try
        if (DBLookupCboProp.KeyValue <> codProp) then
            begin
                if (cenGerado and Not (cenSalvado)) then
                    begin
                        resp := MessageDlg ('Deseja salvar o novo cenário?', MTWarning, [MbYes,MbNo,MbCancel], 0);
                        if (resp = mrYes) then
                            cenSalvado := SalvarCenario;
                        end;
                    if (resp <> mrCancel) then
                        RecuperarPropriedade (self)
                    else
                        begin
                            FrmSimul.TblProp.Edit;
                            DBLookupCboProp.KeyValue := codProp;
                        end;
                    end;
            finally
                Screen.Cursor := crDefault;
            end;
        end;
    end;
end;

```

```

procedure TFrmCenNovo.BtnConfirmarClick(Sender: TObject);
var t, anoAT, c, cs: Integer;
begin
    Screen.Cursor := crHourglass;
    try try
        if cenGerado then
            raise Exception.Create ('O cenário já foi gerado');
        if (TxtTurno.Text = "") then
            raise Exception.Create ('Preencha o campo "Turno de trabalho".');
        turnoTrab := StrToFloat (TxtTurno.Text);
        if (turnoTrab > FrmReg.TblReg.FieldValues['TurnoDeTrabalho']) then
            raise Exception.Create ('O turno de trabalho é inválido')
        else if ((numCol = 0) or (numCol > 1)) then
            raise Exception.Create ('Deve haver uma colhedora selecionada');
    end;
    end;
end;

```

```

areaTotalCen := 0;
for t := 1 to (StgCenTal.RowCount - 1) do
begin
  if Not (talCultivar[t,1] <> 0) then
    raise Exception.Create ('Nenhum cultivar foi atribuído ao talhão ' + letra[t])
  else if (StgCenTal.Cells[3,t] = "") then
    raise Exception.Create ('Forneça a área de lavoura do talhão ' + letra[t])
  else if (StgCenTal.Cells[5,t] = "") then
    raise Exception.Create ('Forneça a data de semeadura do talhão ' + letra[t])
  else if ((t > 1) and (StrToDate (StgCenTal.Cells[7,t]) <= StrToDate (StgCenTal.Cells[8,t - 1]))) then
    raise Exception.Create ('O período de colheita do talhão ' + letra[t] + ' coincide com o do talhão ' +
letra[t - 1] + '.');
    areaTotalCen := areaTotalCen + StrToFloat (StgCenTal.Cells[3,t]);
  end;
  if (areaTotalCen > areaTotalProp) then
    raise Exception.Create ('A área total do cenário é maior que a área da propriedade da qual se
deseja gerá-lo.');
```

```

  SetLength (indColhedora, numCol);
  SetLength (linAno, (ultAno + 1) - priAno);
  SetLength (detAno, (ultAno + 1) - priAno);
  SetLength (colAno, (ultAno + 1) - priAno);
  SetLength (resultPropAno, StgCenTal.RowCount - 1);
  SetLength (resultMaqAno, StgCenTal.RowCount - 1);
  SetLength (resultFinalTal, StgCenTal.RowCount - 1);
  c := 0;
  for cs := 0 to (ChkListBoxCenCol.Items.Count - 1) do
    if ChkListBoxCenCol.Checked[cs] then
      begin
        indColhedora[c] := cs;
        c := c + 1;
      end;
  for t := 1 to (StgCenTal.RowCount - 1) do
    begin
      SetLength (resultPropAno[t - 1], (ultAno + 1) - priAno);
      SetLength (resultMaqAno[t - 1], (ultAno + 1) - priAno);
      for anoAT := priAno to ultAno do
        begin
          SetLength (resultMaqAno[t - 1, (anoAT - priAno)], numCol);
          if ((t - 1) = 0) then
            begin
              SetLength (colAno[anoAT - priAno], numCol);
              detAno[anoAT - priAno, 0] := anoAT;
              detAno[anoAT - priAno, 1] := 1;
              detAno[anoAT - priAno, 2] := 0;
              detAno[anoAT - priAno, 3] := 0;
              detAno[anoAT - priAno, 4] := 0;
              detAno[anoAT - priAno, 5] := 0;
              for c := 0 to (numCol - 1) do
                colAno[anoAT - priAno, c] := 0;
              end;
            end;
        end;
      end;
    end;
  LimparCenario (1);
  FuncPerdas (self);
  ServMecanizado (self);
  ResultadoFinal (7);

```

```

    cenGerado := GerarCodigo;
finally
    Screen.Cursor := crDefault;
end;
except
    on EConvertError do
        raise Exception.Create ('O valor do campo é inválido');
    end;
end;

procedure TFrmCenNovo.StgCenTalGetEditText(Sender: TObject; ACol,
    ARow: Integer; var Value: String);
var R: TRect;
begin
    valCell := StgCenTal.Cells[ACol,ARow];
    R := StgCenTal.CellRect (ACol,ARow);
    R.Left := R.Left + StgCenTal.Left;
    R.Right := R.Right + StgCenTal.Left;
    R.Top := R.Top + StgCenTal.Top;
    R.Bottom := R.Bottom + StgCenTal.Top;
    with TxtCell do
    begin
        case ACol of
            0, 1, 2, 3, 4, 9, 10: EditMask := '';
            5, 6, 7, 8: EditMask := '!99/99/99;1;_';
        end;
        Left := R.Left + 2;
        Top := R.Top + 2;
        Width := R.Right - R.Left;
        Height := R.Bottom - R.Top;
        Text := valCell;
        Visible := true;
        SetFocus;
    end;
end;

procedure TFrmCenNovo.TxtCellExit(Sender: TObject);
var t, atraso: Integer;
    valido: Boolean;
    data: TDateTime;
begin
    try try
        case StgCenTal.Col of
            0, 1, 2, 4, 6, 9, 10: begin
                StgCenTal.Cells[StgCenTal.Col,StgCenTal.Row] := valCell;
                if (valCell <> TxtCell.Text) then
                    MessageDlg ('O valor não pode ser alterado porque o campo "" +
StgCenTal.Cells[StgCenTal.Col,0] + "" está aberto somente para leitura.',
                        MTErrors, [MbOK], 0);
                end;
            3: StgCenTal.Cells[3,StgCenTal.Row] := FloatToStr (StrToFloat (TxtCell.Text));
            5: with StgCenTal do
                begin
                    valido := true;
                    data := StrToDate (TxtCell.Text);

```

```

        if (data < StrToDate (FormatDateTime ('dd/mm/' + Copy (IntToStr (StrToInt (FormatDateTime
('yyyy', Date)) - 1), 3, 2), FrmReg.TblReg.FieldValues['PriDataDeSemeadura'])))
        or (data > StrToDate (FormatDateTime ('dd/mm/' + Copy (IntToStr (StrToInt (FormatDateTime
('yyyy', Date)) - 1), 3, 2), FrmReg.TblReg.FieldValues['UltDataDeSemeadura']))) then
            valido := false
        else
            for t := 1 to (RowCount - 1) do
                if ((t <> Row) and (talCultivar[t,1] = talCultivar[Row,1]) and (Cells[5,t] = TxtCell.Text)) then
                    valido := false;
            if valido then
                begin
                    Cells[5,Row] := TxtCell.Text;
                    Cells[6,Row] := FormatDateTime ('dd/mm/yy', data + StrToInt (Cells[4,Row]));
                    Cells[7,Row] := Cells[6,Row];
                    Cells[8,Row] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[7,Row]) + 24);
                    Cells[9,Row] := '0';
                    Cells[10,Row] := '24';
                    t := -1;
                    if ((Row > 1) and (Cells[5,Row - 1] <> '') and (StrToDate (Cells[7,Row]) <= StrToDate
(Cells[8,Row - 1]))) then
                        t := Row - 1
                    else if ((Row < (RowCount - 1)) and (Cells[5,Row + 1] <> '') and (StrToDate (Cells[7,Row + 1])
<= StrToDate (Cells[8,Row]))) then
                        t := Row + 1;
                    if (t >= 0) then
                        MessageDlg ('O período de colheita do talhão ' + letra[Row] + ' coincide com o do talhão ' +
letra[t] + '.',
                                MTWarning, [MbOK], 0);
                end
            else
                begin
                    Cells[5,Row] := valCell;
                    MessageDlg ('A data de semeadura é inválida.', MTErrror, [MbOK], 0);
                end
            end;
7: with StgCenTal do
    begin
        data := StrToDate (TxtCell.Text);
        if (data < (StrToDate (Cells[6,Row]) - 3))
        or (data > (StrToDate (Cells[6,Row]) + 27))
        or (data > StrToDate (Cells[8,Row]))
        or ((StrToDate (Cells[8,Row]) - data) > 44) then
            begin
                Cells[7,Row] := valCell;
                MessageDlg ('A data de início da colheita é inválida.', MTErrror, [MbOK], 0);
            end
        else
            begin
                atraso := 0;
                if (data > (StrToDate (Cells[6,Row]) + 3)) then
                    begin
                        atraso := Trunc (data - (StrToDate (Cells[6,Row]) + 3));
                        MessageDlg ('A data ' + TxtCell.Text + ' excede em ' + FloatToStr (atraso) + ' dia(s) o limite
máximo para o início provável da colheita.',
                                MTWarning, [MbOK], 0);
                    end
                end;
    end;

```

```

Cells[7,Row] := TxtCell.Text;
Cells[9,Row] := FloatToStr (atraso);
Cells[10,Row] := FloatToStr ((StrToDate (Cells[8,Row]) + atraso) - data);
if ((Row > 1) and (Cells[5,Row - 1] <> '') and (data <= StrToDate (Cells[8,Row - 1]))) then
    MessageDlg ('O período de colheita do talhão ' + letra[Row] + ' coincide com o do talhão ' +
letra[Row - 1] + '.',
                MTWarning, [MbOK], 0);
end;
end;
8: with StgCenTal do
begin
    data := StrToDate (TxtCell.Text);
    atraso := Trunc ((data + StrToInt (Cells[9,Row])) - StrToDate (Cells[7,Row]));
    if ((data < StrToDate (Cells[7,Row])) or (atraso > 44)) then
    begin
        Cells[8,Row] := valCell;
        MessageDlg ('A data de término da colheita é inválida.', MError, [MbOK], 0);
    end
    else
    begin
        Cells[8,Row] := TxtCell.Text;
        Cells[10,Row] := FloatToStr (atraso);
        if ((Row < (RowCount - 1)) and (Cells[5,Row + 1] <> '') and (StrToDate (Cells[7,Row + 1]) <=
data)) then
            MessageDlg ('O período de colheita do talhão ' + letra[Row] + ' coincide com o do talhão ' +
letra[Row + 1] + '.',
                        MTWarning, [MbOK], 0);
        end;
    end;
end;
finally
    TxtCell.Visible := false;
    StgCenTal.SetFocus;
end;
except
    on EConvertError do
    begin
        StgCenTal.Cells[StgCenTal.Col,StgCenTal.Row] := valCell;
        MessageDlg ('O valor do campo é inválido.', MError, [MbOK], 0);
    end;
end;
end;

procedure TFrmCenNovo.DBLookupCboCulKeyPress(Sender: TObject;
var Key: Char);
begin
    if (Key = #13) then
    begin
        Key := #0;
        Perform (CM_DialogKey, VK_TAB, 0);
    end;
end;

procedure TFrmCenNovo.TxtCellKeyPress(Sender: TObject; var Key: Char);
begin
    if (Key = #13) then

```

```

begin
  Key := #0;
  Perform (CM_DialogKey, VK_TAB, 0);
end;
end;

procedure TFrmCenNovo.FuncPerdas(Sender: TObject);
var t, anoAT, aln, aPrev, aFi, atrasoStart, atraso, diasUT, ARow: Integer;
    pAN, pAT, prodMax, ritmoTrab, r, perdTotal, prodTotal, areaC: Real;
    sitData: array of Integer;
begin
  for anoAT := priAno to ultAno do
    begin
      linAno[anoAT - priAno, 0] := StgCustoOpt.RowCount - 1;
      StgCustoOpt.Cells[0, StgCustoOpt.RowCount - 1] := 'ANO - ' + IntToStr (anoAT);
      StgCustoOpt.RowCount := StgCustoOpt.RowCount + 2;
      for t := 1 to (StgCenTal.RowCount - 1) do
        begin
          StgCustoOpt.Cells[0, StgCustoOpt.RowCount - 1] := 'TALHÃO ' + letra[t];
          aln := StrToInt (StgCenTal.Cells[9, t]);
          aPrev := StrToInt (StgCenTal.Cells[10, t]);
          pAT := talCultivar[t, 4];
          atrasoStart := -1;
          repeat
            atrasoStart := atrasoStart + 1;
            pAN := pAT;
            pAT := talCultivar[t, 2] * Sqr (atrasoStart) + talCultivar[t, 3] * atrasoStart + talCultivar[t, 4];
          until (pAT > pAN);
          atrasoStart := atrasoStart - 1;
          prodMax := (talCultivar[t, 2] * -1) * Sqr(atrasoStart) + (talCultivar[t, 3] * -1) * atrasoStart +
            talCultivar[t, 5];
          with FrmSimul.QryPrec do
            begin
              Close;
              SQL.Clear;
              SQL.Add ('select *');
              SQL.Add ('from "REGIÃO_Precipitação"');
              SQL.Add ('where CódRegião = ' + IntToStr (FrmReg.TblReg.FieldValues['CódRegião']));
              SQL.Add ('and Data between ' + FormatDateTime ('mm/dd/' + IntToStr (anoAT), StrToDate
                (StgCenTal.Cells[7, t])) + ' and ' + FormatDateTime ('mm/dd/' + IntToStr (anoAT), StrToDate
                (StgCenTal.Cells[8, t])) + '"));
              SQL.Add ('and ' + FormatDateTime ('mm/dd/' + IntToStr (anoAT), StrToDate
                (StgCenTal.Cells[8, t])) + '"));
              Open;
              First;
            end;
            aFi := 0;
            diasUT := 0;
            pAT := 0;
            SetLength (sitData, (aPrev + 1) - aln);
            for atraso := 0 to (aPrev - aln) do
              begin
                pAN := pAT;
                pAT := FrmSimul.QryPrec.FieldValues['Precipitação'];
                if ((pAT + pAN) <= precCrit) then
                  begin
                    sitData[atraso] := 1;

```

```

    aFi := atraso;
    diasUT := diasUT + 1;
end
else
    sitData[atraso] := 0;
    StgCustoOpt.RowCount := StgCustoOpt.RowCount + 1;
    StgCustoOpt.Cells[0, StgCustoOpt.RowCount - 1] := FormatDateTime ('dd/mm/yy',
FrmSimul.QryPrec.FieldValues["Data"]);
    StgCustoOpt.Cells[1, StgCustoOpt.RowCount - 1] := Format ('%n', [pAT]);
    FrmSimul.QryPrec.Next;
end;
if (diasUT > 0) then
begin
    detAno[anoAT - priAno, 2] := detAno[anoAT - priAno, 2] + diasUT * turnoTrab;
    ritmoTrab := StrToFloat (StgCenTal.Cells[3, t]) / diasUT;
end
else
begin
    detAno[anoAT - priAno, 1] := 0;
    ritmoTrab := 0;
end;
perdTotal := 0;
prodTotal := 0;
areaC := 0;
for atraso := aln to aPrev do
begin
    ARow := StgCustoOpt.RowCount - ((aPrev + 1) - atraso);
    if (sitData[atraso - aln] = 1) then
begin
        r := ritmoTrab;
        StgCustoOpt.Cells[2, ARow] := 'Útil';
end
else
begin
        r := 0;
        StgCustoOpt.Cells[2, ARow] := 'Inútil';
end;
if (atraso > atrasoStart) then
    pAT := talCultivar[t, 2] * Sqr(atraso) + talCultivar[t, 3] * atraso + talCultivar[t, 4]
else
    pAT := 0;
perdTotal := perdTotal + pAT * r;
prodTotal := prodTotal + (prodMax - pAT) * r;
areaC := areaC + r;
StgCustoOpt.Cells[3, ARow] := IntToStr (atraso);
StgCustoOpt.Cells[4, ARow] := Format ('%n', [pAT * r]);
StgCustoOpt.Cells[5, ARow] := Format ('%n', [perdTotal]);
StgCustoOpt.Cells[6, ARow] := Format ('%n', [(prodMax - pAT) * r]);
StgCustoOpt.Cells[7, ARow] := Format ('%n', [prodTotal]);
StgCustoOpt.Cells[8, ARow] := Format ('%n', [areaC]);
end;
detAno[anoAT - priAno, 3] := detAno[anoAT - priAno, 3] + perdTotal;
detAno[anoAT - priAno, 4] := detAno[anoAT - priAno, 4] + prodTotal;
resultPropAno[t - 1, (anoAT - priAno), 0] := perdTotal;
resultPropAno[t - 1, (anoAT - priAno), 1] := prodTotal;
resultPropAno[t - 1, (anoAT - priAno), 2] := aFi;

```

```

resultPropAno[t - 1,(anoAT - priAno),3] := diasUT;
StgCustoOpt.RowCount := StgCustoOpt.RowCount + 9;
StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 9] := 'RESULTADOS (talhão ' + letra[t] + ')';
StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 8] := 'Início da colheita:.';
StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 8] := FormatDateTime ('dd/mm/' + Copy (IntToStr
(anoAT), 3, 2), StrToDate (StgCenTal.Cells[7,t]));
StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 7] := 'Término da colheita:.';
StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 6] := 'Nº de dias úteis:.';
StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 5] := 'Ritmo de trabalho (ha/dia):.';
StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 4] := 'Perda total (US$):.';
StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 3] := 'Renda total (US$):.';
if (ritmoTrab > 0) then
begin
StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 7] := FormatDateTime ('dd/mm/yy', StrToDate
(StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 8]) + aFi);
StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 6] := IntToStr (diasUT);
StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 5] := Format ('%n', [ritmoTrab]);
StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 4] := Format ('%n', [perdTotal * PrecSaco]);
StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 3] := Format ('%n', [prodTotal * PrecSaco]);
end
else
begin
StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 7] := 'Colheita não realizada';
StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 6] := 'Nenhum dia foi constatado';
StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 5] := 'Colheita não realizada';
StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 4] := 'Colheita não realizada';
StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 3] := 'Colheita não realizada';
end;
end;
with StgCustoOpt do
begin
RowCount := StgCustoOpt.RowCount + 6;
Cells[0,RowCount - 7] := 'RESULTADOS (ano de ' + IntToStr (anoAT) + ')';
Cells[0,RowCount - 6] := 'Perda total (US$):.';Cells[1,RowCount - 6] := Format ('%n', [detAno[anoAT
- priAno,3] * PrecSaco]);
Cells[0,RowCount - 5] := 'Renda total (US$):.';Cells[1,RowCount - 5] := Format ('%n', [detAno[anoAt
- priAno,4] * PrecSaco]);
Cells[0,RowCount - 4] := 'Tempo de utilização anual (h):.';Cells[1,RowCount - 4] := Format ('%n',
[detAno[anoAT - priAno,2]]);
Cells[0,RowCount - 3] := 'Situação do ano:.';
if (detAno[anoAT - priAno,1] = 1) then
Cells[1,RowCount - 3] := 'Válido para a estatística'
else
Cells[1,RowCount - 3] := 'Inválido para a estatística';
end;
end;
StgCustoOpt.RowCount := StgCustoOpt.RowCount - 2;
end;

procedure TFrmCenNovo.ServMecanizado(Sender: TObject);
var t, anoAT, c: Integer;
numColAN, numColAT, custoDePR, custoDeUT: Real;
custoPT, custoImPT, custoDeSM, custoDeSMTTotal, custoUnitario: Real;
capColhedora: Real;
begin
for anoAT := priAno to ultAno do

```

```

begin
  linAno[anoAT - priAno,1] := StgCustoMaq.RowCount - 1;
  StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 1] := 'ANO - ' + IntToStr (anoAT);
  StgCustoMaq.RowCount := StgCustoMaq.RowCount + 2;
  if (detAno[anoAT - priAno,1] = 1) then
    begin
      StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 3] :=
        StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 3] + ' (válido)';
      for c := 0 to (numCol - 1) do
        begin
          capColhedora := codColhedora[indColhedora[c],1] * codColhedora[indColhedora[c],3] * (Eficiencia
            / 1000) * turnoTrab;
          numColAN := 0;
          for t := 1 to (StgCenTal.RowCount - 1) do
            begin
              numColAT := (StrToFloat (StgCenTal.Cells[3,t]) / resultPropAno[t - 1,(anoAT - priAno),3]) /
                capColhedora;
              if (numColAT > numColAN) then
                numColAN := numColAT;
            end;
            numColAT := Int (numColAN);
            if ((numColAN - numColAT) > 0) then
              numColAT := numColAT + 1;
          with StgCustoMaq do
            begin
              RowCount := RowCount + 6;
              Cells[0,RowCount - 7] := 'Marca: ' + modColhedora[indColhedora[c],0];
              Cells[0,RowCount - 6] := 'Modelo: ' + modColhedora[indColhedora[c],1];
              Cells[0,RowCount - 5] := 'Plataforma: ' + Format ('%n m', [codColhedora[indColhedora[c],3]]);
              Cells[0,RowCount - 4] := 'Velocidade: ' + Format ('%n km/h', [codColhedora[indColhedora[c],1]]);
              Cells[0,RowCount - 3] := 'Número de colhedoras: ' + FloatToStr (numColAT);
              Cells[0,RowCount - 2] := 'Capacidade: ' + Format ('%n ha/dia', [capColhedora * numColAT]);
              Cells[0,RowCount - 1] := 'RESULTADOS';
              custoDeSMTtotal := 0;
              for t := 1 to (StgCenTal.RowCount - 1) do
                begin
                  numColAN := (StrToFloat (StgCenTal.Cells[3,t]) / resultPropAno[t - 1,(anoAT - priAno),3]) /
                    capColhedora;
                  custoDePR := (codColhedora[indColhedora[c],4] * (1 - Residual / 100) * ((exp (ln (1 + (Juros /
                    100)) * VidaUtil) * (Juros / 100)) / (exp (ln (1 + (Juros / 100)) * VidaUtil) - 1))) +
                    ((codColhedora[indColhedora[c],4] * (Residual / 100)) * (Juros / 100)) +
                    (codColhedora[indColhedora[c],4] * (SegAloj / 100));
                  custoDeUT := (((Salario / 220) * 2.1) + (0.184 * codColhedora[indColhedora[c],2]) * Diesel +
                    (((0.00059 * codColhedora[indColhedora[c],2]) + 0.02169) * (Diesel * 4.5)) + (0.4 *
                    codColhedora[indColhedora[c],4]) / (VidaUtil * detAno[anoAT - priAno,2])) * detAno[anoAT - priAno,2];
                  custoPT := ((custoDePR + custoDeUT) * numColAN) * (StrToFloat (StgCenTal.Cells[3,t]) /
                    areaTotalCen);
                  custolmPT := ((custoDePR + (((Salario / 220) * 2.1) * detAno[anoAT - priAno,2])) * (numColAT -
                    numColAN)) * (StrToFloat (StgCenTal.Cells[3,t]) / areaTotalCen);
                  custoDeSM := custoPT + custolmPT;
                  custoUnitario := (custoDeSM + (resultPropAno[t - 1,(anoAT - priAno),0] * PrecSaco)) /
                    resultPropAno[t - 1,(anoAT - priAno),1];
                  RowCount := RowCount + 1;
                  Cells[0,RowCount - 1] := 'Talhão ' + letra[t];
                  Cells[1,RowCount - 1] := Format ('%n', [numColAN / numColAT]);
                  Cells[2,RowCount - 1] := Format ('%n', [1 - numColAN / numColAT]);
                end;
              RowCount := RowCount + 1;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

Cells[3,RowCount - 1] := Format ('%n', [custoPT]);
Cells[4,RowCount - 1] := Format ('%n', [custoImPT]);
Cells[5,RowCount - 1] := Format ('%n', [custoDeSM]);
Cells[6,RowCount - 1] := Format ('%n', [custoUnitario]);
custoDeSMTTotal := custoDeSMTTotal + custoDeSM;
colAno[anoAT - priAno,c] := colAno[anoAT - priAno,c] + custoUnitario;
resultMaqAno[t - 1,(anoAT - priAno),c,0] := numColAT;
resultMaqAno[t - 1,(anoAT - priAno),c,1] := numColAN / numColAT;
resultMaqAno[t - 1,(anoAT - priAno),c,2] := custoPT;
resultMaqAno[t - 1,(anoAT - priAno),c,3] := custoImPT;
resultMaqAno[t - 1,(anoAT - priAno),c,4] := custoUnitario;
end;
RowCount := RowCount + 4;
colAno[anoAT - priAno,c] := colAno[anoAT - priAno,c] / (StgGenTal.RowCount - 1);
Cells[0,RowCount - 4] := 'SERVIÇO MECANIZADO (US$)';Cells[1,RowCount - 4] := Format
('%n', [custoDeSMTTotal]);
Cells[0,RowCount - 3] := 'CUSTO M. COLHEDORA (US$/sc)';Cells[1,RowCount - 3] := Format
('%n', [colAno[anoAT - priAno,c]]);
end;
detAno[anoAT - priAno,5] := PrecSaco * (detAno[anoAT - priAno,4] - detAno[anoAT - priAno,3]) -
custoDeSMTTotal;
StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 1] := 'RENDA LÍQUIDA
(US$)';StgCustoMaq.Cells[1,StgCustoMaq.RowCount - 1] := Format ('%n', [detAno[anoAT -
priAno,5]]);
StgCustoMaq.RowCount := StgCustoMaq.RowCount + 2;
end;
else
StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 3] :=
StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 3] + ' (inválido)';
end;
StgCustoMaq.RowCount := StgCustoMaq.RowCount - 2;
end;

procedure TFrmCenNovo.ResultadoFinal(numVar: Integer);
var a, i, ind, x, ACol: Integer;
    valoresVar: array[0..6] of array of Real;
    valoresTal: array of array[0..1] of array of Real;
    valoresCol: array of array of Real;
begin
    SetLength (valoresTal, StgCenTal.RowCount - 1);
    ind := 0;
    x := 0;
    for a := 0 to High (detAno) do
        if (detAno[a,1] = 1) then
            begin
                for i := 0 to (numVar - 1) do
                    SetLength (valoresVar[i], ind + 1);
                    valoresVar[0,ind] := (detAno[a,3] * PrecSaco) / areaTotalCen;
                    valoresVar[1,ind] := detAno[a,5] / areaTotalCen;
                    valoresVar[2,ind] := (detAno[a,4] * PrecSaco - detAno[a,5]) / areaTotalCen;
                    if (numVar = 7) then
                        begin
                            valoresVar[3,ind] := colAno[a,0];
                            valoresVar[4,ind] := valoresVar[2,ind] - (detAno[a,3] * PrecSaco) / areaTotalCen;
                            valoresVar[5,ind] := detAno[a,2];

```

```

    valoresVar[6,ind] := resultMaqAno[0,a,0,0];
    x := 2;
end
else
    valoresVar[3,ind] := detAno[a,6];
for i := 1 to (StgCenTal.RowCount - 1) do
begin
    SetLength (valoresTal[i - 1,0], ind + 1);
    SetLength (valoresTal[i - 1,1], ind + 1);
    valoresTal[i - 1,0,ind] := resultPropAno[i - 1,a,4 - x] + 1;
    valoresTal[i - 1,1,ind] := resultPropAno[i - 1,a,5 - x];
end;
ind := ind + 1;
end;
with StgResultFinalEst do
begin
    Cells[0,0] := 'Nº de anos da série:'; Cells[1,0] := IntToStr ((ultAno - priAno) + 1);
    Cells[0,1] := 'Nº de anos válidos:'; Cells[1,1] := IntToStr (ind);
    if (ind > 0) then
    begin
        RowCount := RowCount + 3;
        Cells[0,RowCount - 2] := 'ESTATÍSTICA';
        for ACol := 0 to 4 do
            Cells[ACol,RowCount - 1] := atributo[ACol];
        for i := 0 to (numVar - 1) do
        begin
            RowCount := RowCount + 1;
            CalcularMedias (0, i, valoresVar[i]);
            Cells[0,RowCount - 1] := variavel[i];
            for ACol := 1 to 4 do
                Cells[ACol,RowCount - 1] := Format ('%n', [resultFinalVar[i,ACol - 1]]);
            end;
            RowCount := RowCount + 3;
            Cells[0,RowCount - 2] := 'DETALHES DO TALHÃO';
            Cells[0,RowCount - 1] := 'Prazo (dias)';
            for i := 1 to 4 do
            begin
                RowCount := RowCount + 1;
                Cells[0,RowCount - 1] := atributo[i];
            end;
            ind := RowCount;
            RowCount := RowCount + 1;
            Cells[0,RowCount - 1] := 'Nº de dias úteis';
            for i := 1 to 4 do
            begin
                RowCount := RowCount + 1;
                Cells[0,RowCount - 1] := atributo[i];
            end;
            for ACol := 1 to (StgCenTal.RowCount - 1) do
            begin
                if (ACol > (ColCount - 1)) then
                    ColCount := ColCount + 1;
                Cells[ACol,ind - 6] := 'Talhão ' + letra[ACol];
                CalcularMedias (1, ACol - 1, valoresTal[ACol - 1,0]);
                CalcularMedias (2, ACol - 1, valoresTal[ACol - 1,1]);
                for i := 0 to 3 do

```

```

begin
  Cells[ACol,ind - (4 - i)] := Format ('%n', [resultFinalTal[ACol - 1,0,i]]);
  Cells[ACol,RowCount - (4 - i)] := Format ('%n', [resultFinalTal[ACol - 1,1,i]]);
end;
end;
if (x = 2) then
begin
  RowCount := RowCount + 3;
  Cells[0,RowCount - 2] := 'DISTRIBUIÇÃO DE COLHEDORA';
  Cells[0,RowCount - 1] := 'Nº de colhedoras';
  Cells[1,RowCount - 1] := 'Freq. obs.';
  Cells[2,RowCount - 1] := 'Freq. acum.';
  Cells[3,RowCount - 1] := 'Prob. obs. (%)';
  Cells[4,RowCount - 2] := 'F.U. PROD. (%)';
  for ACol := 1 to (StgCenTal.RowCount - 1) do
  begin
    if ((ACol + 4) > ColCount) then
      ColCount := ColCount + 1;
      Cells[ACol + 3,RowCount - 1] := 'Talhão ' + letra[ACol];
    end;
    DistribuirValores (0, valoresVar[6]);
    SetLength (valoresCol, High (categoria[0]) + 1);
    for ind := High (categoria[0]) downto 0 do
    begin
      SetLength (valoresCol[ind], StgCenTal.RowCount - 1);
      for i := 1 to (StgCenTal.RowCount - 1) do
        valoresCol[ind,i - 1] := 0;
      for a := 0 to High (detAno) do
        if (detAno[a,1] = 1) then
          if (resultMaqAno[0,a,0,0] = categoria[0,ind,0]) then
            for i := 1 to (StgCenTal.RowCount - 1) do
              valoresCol[ind,i - 1] := valoresCol[ind,i - 1] + (resultMaqAno[i - 1,a,0,1] * 100) /
categoria[0,ind,1];
            RowCount := RowCount + 1;
            for ACol := 0 to 3 do
              Cells[ACol,RowCount - 1] := Format ('%n', [categoria[0,ind,ACol]]);
            for ACol := 1 to (StgCenTal.RowCount - 1) do
              Cells[ACol + 3,RowCount - 1] := Format ('%n', [valoresCol[ind,ACol - 1]]);
            end;
          end;
        end;
      end;
    else
      raise Exception.Create ('Não foi possível gerar o cenário porque não há nenhum ano na série
temporal');
    end;
  end;
end;

function TFrmCenNovo.SalvarCenario: Boolean;
var t, anoAT, c: Integer;
begin
  FrmSimul.QryPropCen.InsertRecord ([codCen,codProp]);
  FrmSimul.QrySimul.InsertRecord ([codCen,Date,turnoTrab,precCrit,StgCenTal.RowCount - 1]);
  FrmSimul.QrySimulInd.InsertRecord ([codCen,PrecSaco,Juros,VidaUtil,Eficiencia,Salario,
SegAloj,Diesel,Residual]);
  for c := 0 to (numCol - 1) do

```

```

    FrmSimul.QrySimulCol.InsertRecord
    ([codCen,c,codColhedora[indColhedora[c],0],codColhedora[indColhedora[c],1]]);
    for t := 1 to (StgCenTal.RowCount - 1) do
    begin
        FrmSimul.QrySimulTal.InsertRecord
        ([codCen,t,talCultivar[t,1],StgCenTal.Cells[5,t],StgCenTal.Cells[7,t],StgCenTal.Cells[10,t],StgCenTal.C
        ells[3,t]]);
        for anoAT := priAno to ultAno do
            if (detAno[anoAT - priAno,1] = 1) then
            begin
                FrmSimul.QryResultPropAno.InsertRecord ([codCen,t,anoAT,Null,StrToDate
                (StgCenTal.Cells[7,t]) + resultPropAno[t - 1,(anoAT - priAno),2],
                resultPropAno[t - 1,(anoAT - priAno),0],resultPropAno[t - 1,(anoAT -
                priAno),1],
                resultPropAno[t - 1,(anoAT - priAno),2],resultPropAno[t - 1,(anoAT -
                priAno),3]]);
                for c := 0 to (numCol - 1) do
                    FrmSimul.QryResultMaqAno.InsertRecord ([codCen,t,anoAT,c,resultMaqAno[t - 1,(anoAT -
                    priAno),c,0],resultMaqAno[t - 1,(anoAT - priAno),c,1],
                    resultMaqAno[t - 1,(anoAT - priAno),c,2],resultMaqAno[t - 1,(anoAT -
                    priAno),c,3],resultMaqAno[t - 1,(anoAT - priAno),c,4]]);
                end;
            end;
        SalvarCenario := true;
        FrmSimul.QryPropCen.Refresh;
        MessageDlg ('O novo cenário foi salvo com êxito.', MTInformation, [MbOK], 0);
    end;
end;

```

```

procedure TFrmCenNovo.OpSalvarClick(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    try
        if Not (cenGerado) then
            raise Exception.Create ('Não há nenhum cenário gerado')
        else if (cenSalvado) then
            raise Exception.Create ('O novo cenário já foi salvo')
        else
            cenSalvado := SalvarCenario;
        finally
            Screen.Cursor := crDefault;
        end;
    end;
end;

```

```

procedure TFrmCenNovo.OpIndicadorClick(Sender: TObject);
begin
    try
        FrmIndicador := TFrmIndicador.Create (Application);
        FrmIndicador.ShowModal;
    finally
        FrmIndicador.Free;
    end;
end;

```

```

procedure TFrmCenNovo.FormClose(Sender: TObject; var Action: TCloseAction);
var resp: Word;
begin

```

```

Screen.Cursor := crHourglass;
try
  if (cenGerado and Not (cenSalvado)) then
    begin
      resp := MessageDlg ('Deseja salvar o novo cenário?', MTWarning, [MbYes,MbNo,MbCancel], 0);
      if (resp = mrYes) then
        cenSalvado := SalvarCenario
      else if (resp = mrCancel) then
        Action := caNone;
      end;
    finally
      Screen.Cursor := crDefault;
    end;
  end;

procedure TFrmCenNovo.BtnCancelarClick(Sender: TObject);
begin
  Close;
end;

function TFrmCenNovo.LocalizarAno(anoAT, indGrad: Integer): Boolean;
var a, ARow: Integer;
    valido, achou: Boolean;
begin
  valido := true;
  a := anoAT - priAno;
  if ((anoAT < priAno) or (anoAT > ultAno)) then
    valido := false
  else if Not (anoAT = detAno[a,0]) then
    begin
      a := 0;
      achou := false;
      repeat
        if (anoAT = detAno[a,0]) then
          achou := true
        else
          a := a + 1;
      until (achou or (a > High (detAno)));
      valido := achou;
    end;
  if valido then
    case indGrad of
      0: with StgCustoOpt do
        begin
          ARow := RowCount - (linAno[a,0] + 1);
          if (ARow >= 20) then
            ARow := 20;
          Col := 0;
          Row := linAno[a,0] + ARow;
          Row := Row - ARow;
        end;
      1: with StgCustoMaq do
        begin
          ARow := RowCount - (linAno[a,1] + 1);
          if (ARow >= 20) then
            ARow := 20;

```

```

        Col := 0;
        Row := linAno[a,1] + ARow;
        Row := Row - ARow;
    end;
end;
LocalizarAno := valido;
end;

procedure TFrmCenNovo.OpLocalizarAno1Click(Sender: TObject);
var strAno: String;
begin
    try
        strAno := '';
        if InputQuery ('Série temporal - localizar ano', 'Ano (aaaa)', strAno) then
            if Not (LocalizarAno (StrToInt (strAno), 0)) then
                MessageDlg ('O ano especificado não existe.', MTInformation, [MbOK], 0);
            except
                on EConvertError do
                    raise Exception.Create ('O valor fornecido é inválido');
            end;
        end;
    end;

procedure TFrmCenNovo.OpLocalizarAno2Click(Sender: TObject);
var strAno: String;
begin
    try
        strAno := '';
        if InputQuery ('Série temporal - localizar ano', 'Ano (aaaa)', strAno) then
            if Not (LocalizarAno (StrToInt (strAno), 1)) then
                MessageDlg ('O ano especificado não existe.', MTInformation, [MbOK], 0);
            except
                on EConvertError do
                    raise Exception.Create ('O valor fornecido é inválido');
            end;
        end;
    end;

procedure TFrmCenNovo.ChkListBoxCenColMouseDown(Sender: TObject;
    Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var APoint: TPoint;
begin
    APoint.X := X;
    APoint.Y := Y;
    indCol := ChkListBoxCenCol.ItemAtPos (APoint, true);
end;

procedure TFrmCenNovo.ChkListBoxCenColKeyDown(Sender: TObject;
    var Key: Word; Shift: TShiftState);
begin
    indCol := ChkListBoxCenCol.ItemIndex;
    if (indCol < 0) then
        indCol := 0;
    end;

procedure TFrmCenNovo.OpNovoClick(Sender: TObject);
var resp: Word;
    continua: Boolean;

```

```

begin
  Screen.Cursor := crHourglass;
  try
    continua := true;
    if (cenGerado and Not (cenSalvado)) then
      begin
        resp := MessageDlg ('Deseja salvar o novo cenário?', MTWarning, [MbYes,MbNo,MbCancel], 0);
        if (resp = mrYes) then
          cenSalvado := SalvarCenario
        else if (resp = mrCancel) then
          continua := false;
        end;
      if continua then
        with FrmSimul.TblProp do
          begin
            First;
            Edit;
            DBLookupCboProp.KeyValue := FieldValues['CódPropriedade'];
            RecuperarPropriedade (self);
          end;
        finally
          Screen.Cursor := crDefault;
        end;
      end;
end;

function TFrmCenNovo.GerarCodigo: Boolean;
var ind, e: Integer;
    strInd: String[3];
begin
  with FrmSimul.QrySimul do
    begin
      Close;
      SQL.Clear;
      SQL.Add ('select *');
      SQL.Add ('from "SIMULAÇÃO"');
      SQL.Add ('where CódSimulação like "CEN.' + FrmReg.TblReg.FieldValues['Sigla'] + '%"');
      SQL.Add ('and extract (year from DataDeCriação) = ' + FormatDateTime ('yyyy', Date));
      Open;
      FindLast;
      if Found then
        Val (Copy (FieldValues['CódSimulação'], 7, 3), ind, e)
      else
        ind := 0;
        ind := ind + 1;
        if (ind < 100) then
          begin
            strInd := '0';
            if (ind < 10) then
              strInd := strInd + '0';
            end;
            strInd := strInd + IntToStr (ind);
            codCen := 'CEN.' + FrmReg.TblReg.FieldValues['Sigla'] + strInd + '-' + FormatDateTime ('yy', Date);
          end;
        GerarCodigo := true;
        LblCod.Caption := codCen;
        MessageDlg ('O código do novo cenário é ' + codCen + '.', MTInformation, [MbOK], 0);

```

end;

procedure TFrmCenNovo.DistribuirValores(indVar: Integer; valor: array of Real);

var i, ind, freq: Integer;

 vAT: Real;

 ordenou: Boolean;

begin

 ind := 0;

 repeat

 ind := ind + 1;

 ordenou := true;

 for i := High (valor) downto ind do

 if (valor[i] > valor[i - 1]) then

 begin

 vAT := valor[i - 1];

 valor[i - 1] := valor[i];

 valor[i] := vAT;

 ordenou := false;

 end;

 until ordenou or (ind = High (valor));

 ind := -1;

 for i := 0 to High (valor) do

 if ((i > 0) and (valor[i] = valor[i - 1])) then

 categoria[indVar, ind, 1] := categoria[indVar, ind, 1] + 1

 else

 begin

 ind := ind + 1;

 SetLength (categoria[indVar], ind + 1);

 categoria[indVar, ind, 0] := valor[i];

 categoria[indVar, ind, 1] := 1;

 end;

 freq := 0;

 for ind := High (categoria[indVar]) downto 0 do

 begin

 freq := freq + Trunc (categoria[indVar, ind, 1]);

 categoria[indVar, ind, 2] := freq;

 categoria[indVar, ind, 3] := (freq / (High (valor) + 1)) * 100;

 end;

end;

procedure TFrmCenNovo.CalcularMedias(opResult, indVar: Integer; valor: array of Real);

var i: Integer;

 min, max, media, desvio: Real;

begin

 min := valor[0];

 max := valor[0];

 media := 0;

 for i := 0 to High (valor) do

 begin

 if (min > valor[i]) then

 min := valor[i]

 else if (max < valor[i]) then

 max := valor[i];

 media := media + valor[i];

 end;

 media := media / (High (valor) + 1);

```

desvio := 0;
for i := 0 to High (valor) do
  desvio := desvio + Sqr (valor[i] - media);
desvio := Sqrt (desvio / High (valor));
case opResult of
  0: begin
    resultFinalVar[indVar,0] := min;
    resultFinalVar[indVar,1] := max;
    resultFinalVar[indVar,2] := media;
    resultFinalVar[indVar,3] := desvio;
  end;
  1, 2: begin
    resultFinalTal[indVar,opResult - 1,0] := min;
    resultFinalTal[indVar,opResult - 1,1] := max;
    resultFinalTal[indVar,opResult - 1,2] := media;
    resultFinalTal[indVar,opResult - 1,3] := desvio;
  end;
end;
end;

procedure TFrmCenNovo.OpExportarClick(Sender: TObject);
var t, ind, i: Integer;
    achou: Boolean;
begin
  if Not (StgResultFinalEst.RowCount > 2) then
    raise Exception.Create ('Não há nenhuma informação para exportar');
  try
    FrmExpExcel := TFrmExpExcel.Create (Application);
    with FrmExpExcel do
      begin
        SetLength (indCultivar, 1);
        SetLength (indColhedora, numCol);
        if Not (cenGerado) then
          begin
            opSimul := 0;
            codSimul := codProp;
          end
        else
          begin
            opSimul := 1;
            codSimul := codCen;
          end;
        areaTotalSimul := areaTotalCen;
        stgVirtual := StgResultFinalEst;
        ind := 0;
        for t := 1 to (StgCenTal.RowCount - 1) do
          if (t > 1) then
            begin
              achou := false;
              i := 0;
              repeat
                if (talCultivar[t,1] = indCultivar[i,0]) then
                  achou := true
                else
                  i := i + 1;
              until (achou or (i > ind));
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

    if achou then
    begin
        indCultivar[i,1] := indCultivar[i,1] + StrToFloat (StgCenTal.Cells[3,t]);
        indCultivar[i,2] := indCultivar[i,2] + 1;
    end
    else
    begin
        ind := ind + 1;
        SetLength (indCultivar, ind + 1);
        indCultivar[ind,0] := talCultivar[t,1];
        indCultivar[ind,1] := StrToFloat (StgCenTal.Cells[3,t]);
        indCultivar[ind,2] := 1;
    end;
end
else
begin
    indCultivar[0,0] := talCultivar[1,1];
    indCultivar[0,1] := StrToFloat (StgCenTal.Cells[3,1]);
    indCultivar[0,2] := 1;
end;
capInstal := 0;
for i := 0 to High (indColhedora) do
begin
    indColhedora[i,0] := codColhedora[FrmCenNovo.indColhedora[i],3];
    indColhedora[i,1] := codColhedora[FrmCenNovo.indColhedora[i],2];
    if (opSimul = 0) then
    begin
        indColhedora[i,2] := resultMaqAno[0,0,i,0];
        capInstal := capInstal + (codColhedora[FrmCenNovo.indColhedora[i],1] * indColhedora[i,2] *
codColhedora[FrmCenNovo.indColhedora[i],3] * (Eficiencia / 1000)) * turnoTrab;
    end;
end;
ShowModal;
end;
finally
    FrmExpExcel.Free;
end;
end;

end.

```

G - Código Fonte 15: UCenarioAlterar.pas <<código>>

```

unit UCenarioAlterar;

```

```

interface

```

```

uses

```

```

    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    Menus, StdCtrls, ComCtrls, ToolWin, Buttons, ExtCtrls, DBCtrls, CheckLst,
    Grids, Mask, DB;

```

```

type

```

```

    TFrmCenAlterar = class(TForm)

```

Lbl1: TLabel;
 MainMenuCenAlterar: TMainMenu;
 MenuCenario: TMenuItem;
 MenuEditar: TMenuItem;
 OpSalvar: TMenuItem;
 OpIndicador: TMenuItem;
 ToolBarCenAlterar: TToolBar;
 ToolBtnSalvar: TToolButton;
 Pnl1: TPanel;
 Pnl3: TPanel;
 Pnl2: TPanel;
 BtnConfirmar: TBitBtn;
 BtnCancelar: TBitBtn;
 LblCod: TLabel;
 PgeCtrlCenAlterar: TPageControl;
 TabShtCen: TTabSheet;
 TabShtCustoOpt: TTabSheet;
 TabShtCustoMaq: TTabSheet;
 StatusBarCenAlterar: TStatusBar;
 Pnl4: TPanel;
 GrpBoxCenCol: TGroupBox;
 Pnl5: TPanel;
 GrpBoxCen: TGroupBox;
 GrpBoxPer: TGroupBox;
 Pnl7: TPanel;
 Lbl2: TLabel;
 Lbl3: TLabel;
 Lbl4: TLabel;
 Lbl5: TLabel;
 Lbl6: TLabel;
 TxtTurno: TEdit;
 TxtPrec: TEdit;
 Pnl8: TPanel;
 ChkListBoxCenCol: TCheckListBox;
 GrpBoxCenTal: TGroupBox;
 Pnl9: TPanel;
 StgCenTal: TStringGrid;
 PopupMenuCenTal: TPopupMenu;
 OpIncluirTal: TMenuItem;
 OpExcluirTal: TMenuItem;
 MenuN1: TMenuItem;
 OpLimparListaTal: TMenuItem;
 DBLookupCboCul: TDBLookupComboBox;
 TxtCell: TMaskEdit;
 StgCustoOpt: TStringGrid;
 StgCustoMaq: TStringGrid;
 PopupMenuGrad1: TPopupMenu;
 OpLocalizarAno1: TMenuItem;
 PopupMenuGrad2: TPopupMenu;
 OpLocalizarAno2: TMenuItem;
 ToolBarCustoOpt: TToolBar;
 ToolBtnLocalizarAno1: TToolButton;
 ToolBarCustoMaq: TToolBar;
 ToolBtnLocalizarAno2: TToolButton;
 ToolBarCenTal: TToolBar;
 ToolBtnIncluirTal: TToolButton;

```

ToolBtnExcluirTal: TToolButton;
ToolN1: TToolButton;
ToolBtnLimparListaTal: TToolButton;
TxtPriAno: TEdit;
TxtUltAno: TEdit;
TxtProp: TEdit;
TabShtResultFinal: TTabSheet;
StgResultFinalEst: TStringGrid;
Pnl6: TPanel;
PopupMenuGradResultFinal: TPopupMenu;
OpExportar: TMenuItem;
ToolN2: TToolButton;
ToolBtnIndicador: TToolButton;
procedure ChkListBoxCenColClickCheck(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure OpIncluirTalClick(Sender: TObject);
procedure OpExcluirTalClick(Sender: TObject);
procedure OpLimparListaTalClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure DBLookupCboCulClick(Sender: TObject);
procedure DBLookupCboCulExit(Sender: TObject);
procedure StgCenTalSelectCell(Sender: TObject; ACol, ARow: Integer;
    var CanSelect: Boolean);
procedure BtnConfirmarClick(Sender: TObject);
procedure StgCenTalGetEditText(Sender: TObject; ACol, ARow: Integer;
    var Value: String);
procedure TxtCellExit(Sender: TObject);
procedure DBLookupCboCulKeyPress(Sender: TObject; var Key: Char);
procedure TxtCellKeyPress(Sender: TObject; var Key: Char);
procedure OpSalvarClick(Sender: TObject);
procedure OpIndicadorClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure BtnCancelarClick(Sender: TObject);
procedure OpLocalizarAno1Click(Sender: TObject);
procedure OpLocalizarAno2Click(Sender: TObject);
procedure ChkListBoxCenColMouseDown(Sender: TObject;
    Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
procedure ChkListBoxCenColKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
procedure OpExportarClick(Sender: TObject);
private
{ Private declarations }
priAno, ultAno, numCol, indCol: Integer;
turnoTrab, precCrit, areaTotalProp, areaTotalCen: Real;
codColhedora: array of array[0..4] of Real;
modColhedora: array of array[0..1] of String;
indColhedora: array of Integer;
talCultivar: array[1..26,0..5] of Real;
linAno: array of array[0..1] of Integer;
detAno: array of array[0..5] of Real;
colAno: array of array of Real;
resultPropAno: array of array of array[0..3] of Real;
resultMaqAno: array of array of array of array[0..4] of Real;
resultFinalVar: array[0..6] of array[0..3] of Real;
resultFinalTal: array of array[0..1] of array[0..3] of Real;
categoria: array[0..0] of array of array[0..3] of Real;

```

```

    valCell, codProp, codCen: String;
    cenAlterado, cenSalvado: Boolean;
public
{ Public declarations }
function PesquisarFuncPerdas (codReg, codCul: Integer): Boolean;
function SalvarCenario: Boolean;
function LocalizarAno (anoAT, indGrad: Integer): Boolean;
procedure ListarColhedoras (Sender: TObject);
procedure LimparCenario (opCab: Integer);
procedure RecuperarCenario (Sender: TObject);
procedure RecuperarCenTal (codReg, codCul: Integer);
procedure IncluirTal (Sender: TObject);
procedure ExcluirTal (Sender: TObject);
procedure LimparListaTal (Sender: TObject);
procedure FuncPerdas (Sender: TObject);
procedure ServMecanizado (Sender: TObject);
procedure ResultadoFinal (Sender: TObject);
procedure DistribuirValores (indVar: Integer; valor: array of Real);
procedure CalcularMedias (opResult, indVar: Integer; valor: array of Real);
end;

var
    FrmCenAlterar: TFrmCenAlterar;

implementation

uses USimulacao, UIndicador, URegiao, UMaquina, UCultivar, UExpExcel;

{$R *.DFM}

procedure TFrmCenAlterar.ListarColhedoras(Sender: TObject);
var cs, c: Integer;
    strModelo: String;
begin
    ChkListBoxCenCol.Items.Clear;
    SetLength (codColhedora, FrmMaq.TblCol.RecordCount);
    SetLength (modColhedora, FrmMaq.TblCol.RecordCount);
    SetLength (indColhedora, numCol);
    FrmMaq.TblCol.First;
    cs := 0;
    c := 0;
    repeat
        with FrmMaq.TblCol do
            begin
                codColhedora[cs,0] := FieldValues['CódColhedora'];
                codColhedora[cs,1] := 0;
                codColhedora[cs,2] := FieldValues['Potência'];
                codColhedora[cs,3] := FieldValues['LarguraDePlataforma'];
                codColhedora[cs,4] := FieldValues['PreçoDeAquisição'];
                modColhedora[cs,0] := FieldValues['Marca'];
                modColhedora[cs,1] := FieldValues['Modelo'];
            end;
            strModelo := modColhedora[cs,1];
            strModelo := strModelo + ' (plataforma de ';
            strModelo := strModelo + FloatToStr (codColhedora[cs,3]);
            strModelo := strModelo + ' metros)';
        end;
    until c = numCol - 1;
    c := c + 1;
end;

```

```

ChkListBoxCenCol.Items.Add (strModelo);
if ((numCol > 0) and (c < numCol)) then
with FrmSimul.QrySimulCol do
  if (codColhedora[cs,0] = FieldValues['CódColhedora']) then
    begin
      codColhedora[cs,1] := FieldValues['VelocidadeDeOperação'];
      indColhedora[c] := cs;
      ChkListBoxCenCol.Checked[cs] := true;
      c := c + 1;
      Next;
    end;
  cs := cs + 1;
  FrmMaq.TblCol.Next;
until FrmMaq.TblCol.Eof;
end;

function TFrmCenAlterar.PesquisarFuncPerdas(codReg, codCul: Integer): Boolean;
var ACol, t: Integer;
begin
  PesquisarFuncPerdas := true;
  with FrmCultivar.QryRegCul do
    begin
      Close;
      SQL.Clear;
      SQL.Add ('select *');
      SQL.Add ('from "REGIÃO_CULTIVAR"');
      SQL.Add ('where CódRegião = ' + IntToStr (codReg));
      SQL.Add ('and CódCultivar = ' + IntToStr (codCul));
      Open;
      FindFirst;
    end;
  if Not (FrmCultivar.QryRegCul.Found) then
    begin
      PesquisarFuncPerdas := false;
      for ACol := 4 to 10 do
        StgCenTal.Cells[ACol,StgCenTal.Row] := 'Não encontrado';
      MessageDlg ('Não há nenhuma função de perdas que corresponda às descrições fornecidas.',
        MTInformation, [MbOK], 0);
    end
  else
    with StgCenTal do
      begin
        Cells[4,Row] := FrmCultivar.QryRegCul.FieldValues['CicloDeMaturação'];
        if (Cells[5,Row] <> '') then
          if (Cells[5,Row] = 'Não encontrado') then
            for ACol := 5 to 10 do
              Cells[ACol,Row] := ''
            else
              begin
                Cells[6,Row] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[5,Row]) + StrToInt
(Cells[4,Row]));
                Cells[7,Row] := Cells[6,Row];
                Cells[8,Row] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[7,Row]) + 24);
                Cells[9,Row] := '0';
                Cells[10,Row] := '24';
                t := -1;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

        if ((Row > 1) and (Cells[5,Row - 1] <> "") and (StrToDate (Cells[7,Row]) <= StrToDate
(Cells[8,Row - 1]))) then
            t := Row - 1
            else if (Row < (RowCount - 1)) and (Cells[5,Row + 1] <> "") and (StrToDate (Cells[7,Row + 1]) <=
StrToDate (Cells[8,Row])) then
                t := Row + 1;
            if (t >= 0) then
                MessageDlg ('O período de colheita do talhão ' + letra[Row] + ' coincide com o período do
talhão ' + letra[t] + ',
                    MTWarning, [MbOK], 0);
            end;
        end;
    end;
end;

```

```

procedure TFrmCenAlterar.ChkListBoxCenColClickCheck(Sender: TObject);
var v: Real;
    strVeloc: String;
    valido, continua: Boolean;
begin
    try
        if (indCol = ChkListBoxCenCol.ItemIndex) then
            if ChkListBoxCenCol.Checked[indCol] then
                begin
                    numCol := numCol + 1;
                    strVeloc := FloatToStr (codColhedora[indCol,1]);
                    valido := false;
                    repeat
                        continua := InputQuery (ChkListBoxCenCol.Items.Strings[indCol], 'Velocidade de operação
(km/h)', strVeloc);
                        if continua then
                            begin
                                v := StrToFloat (strVeloc);
                                if ((v < 3) or (v > 7)) then
                                    MessageDlg ('A velocidade de operação é inválida.', MTErrror, [MbOK], 0)
                                else
                                    begin
                                        valido := true;
                                        codColhedora[indCol,1] := v;
                                    end;
                                end;
                            until (valido or Not (continua));
                            if Not (continua) then
                                begin
                                    ChkListBoxCenCol.Checked[indCol] := false;
                                    numCol := numCol - 1;
                                end;
                            end
                        else
                            numCol := numCol - 1
                        else
                            ChkListBoxCenCol.Checked[indCol] := Not (ChkListBoxCenCol.Checked[indCol]);
                    except
                        on EConvertError do
                            begin
                                MessageDlg ('O valor fornecido é inválido.', MTErrror, [MbOK], 0);
                                indCol := ChkListBoxCenCol.ItemIndex;

```

```

        if (codColhedora[indCol,1] = 0) then
        begin
            ChkListBoxCenCol.Checked[indCol] := false;
            numCol := numCol - 1;
        end;
    end;
end;
end;

procedure TFrmCenAlterar.FormShow(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    try
        RecuperarCenario (self);
    finally
        Screen.Cursor := crDefault;
    end;
end;

procedure TFrmCenAlterar.IncluirTal(Sender: TObject);
begin
    with StgCenTal do
    begin
        RowCount := RowCount + 1;
        Rows[RowCount - 1].Clear;
        Cells[0,RowCount - 1] := letra[RowCount - 1];
        talCultivar[RowCount - 1,0] := 0;
        talCultivar[RowCount - 1,1] := 0;
    end;
end;

procedure TFrmCenAlterar.ExcluirTal(Sender: TObject);
var ARow: Integer;
begin
    with StgCenTal do
    begin
        if (Row < (RowCount - 1)) then
            for ARow := Row to (RowCount - 2) do
            begin
                Rows[ARow] := Rows[ARow + 1];
                Cells[0,ARow] := letra[ARow];
                talCultivar[ARow] := talCultivar[ARow + 1];
            end;
        if (RowCount > 2) then
            RowCount := RowCount - 1
        else
            begin
                Rows[1].Clear;
                Cells[0,1] := letra[1];
                talCultivar[1,0] := 0;
                talCultivar[1,1] := 0;
            end;
        end;
    end;
end;

procedure TFrmCenAlterar.LimparListaTal(Sender: TObject);

```

```

var ARow: Integer;
begin
  with StgCenTal do
    begin
      for ARow := (RowCount - 1) downto 1 do
        Rows[ARow].Clear;
        RowCount := 2;
        Cells[0,1] := letra[1];
        talCultivar[1,0] := 0;
        talCultivar[1,1] := 0;
      end;
    end;

    procedure TFrmCenAlterar.OpIncluirTalClick(Sender: TObject);
    begin
      if ((StgCenTal.RowCount < 27) and (talCultivar[StgCenTal.RowCount - 1,1] <> 0)) then
        IncluirTal (self);
      end;

      procedure TFrmCenAlterar.OpExcluirTalClick(Sender: TObject);
      begin
        ExcluirTal (self);
      end;

      procedure TFrmCenAlterar.OpLimparListaTalClick(Sender: TObject);
      begin
        LimparListaTal (self);
      end;

      procedure TFrmCenAlterar.FormCreate(Sender: TObject);
      begin
        with StgCenTal do
          begin
            Cells[0,0] := 'Talhão';ColWidths[0] := 40;
            Cells[1,0] := 'Cultivar';ColWidths[1] := 127;
            Cells[2,0] := 'Grupo de maturação';ColWidths[2] := 104;
            Cells[3,0] := 'Área de lavoura (ha)';ColWidths[3] := 103;
            Cells[4,0] := 'Ciclo de maturação (dias)';ColWidths[4] := 126;
            Cells[5,0] := 'Data de semeadura';ColWidths[5] := 100;
            Cells[6,0] := 'Data provável de colheita';ColWidths[6] := 129;
            Cells[7,0] := 'Início da colheita';ColWidths[7] := 89;
            Cells[8,0] := 'Término previsto';ColWidths[8] := 85;
            Cells[9,0] := 'Atraso inicial (dias)';ColWidths[9] := 94;
            Cells[10,0] := 'Atraso previsto (dias)';ColWidths[10] := 105;
            DefaultRowHeight := DBLookupCboCul.Height;
          end;
        StgCustoOpt.DefaultRowHeight := StgCenTal.DefaultRowHeight;
        with StgCustoMaq do
          begin
            Cells[0,0] := 'Ano/Colhedora/Talhão';ColWidths[0] := 177;
            Cells[1,0] := 'F.U. produtivo';ColWidths[1] := 87;
            Cells[2,0] := 'F.U. improdutivo';ColWidths[2] := 110;
            Cells[3,0] := 'Custo produtivo (US$)';ColWidths[3] := 111;
            Cells[4,0] := 'Custo improdutivo (US$)';ColWidths[4] := 121;
            Cells[5,0] := 'Custo de serviço mecanizado (US$)';ColWidths[5] := 176;
            Cells[6,0] := 'Custo unitário (US$/sc)';ColWidths[6] := 117;
          end;
        end;
      end;
    end;
  end;
end;

```

```

    DefaultRowHeight := StgCenTal.DefaultRowHeight;
end;
with StgResultFinalEst do
begin
    ColWidths[0] := 193;
    ColWidths[3] := 72;
    ColWidths[4] := 81; {era 76}
    DefaultRowHeight := StgCenTal.DefaultRowHeight;
end;
TxtCell.BorderStyle := bsNone;
end;

procedure TFrmCenAlterar.LimparCenario(opCab: Integer);
var ARow: Integer;
begin
    cenAlterado := false;
    cenSalvado := false;
    for ARow := (StgCustoOpt.RowCount - 1) downto 0 do
        StgCustoOpt.Rows[ARow].Clear;
    StgCustoOpt.RowCount := 2;
    for ARow := (StgCustoMaq.RowCount - 1) downto 1 do
        StgCustoMaq.Rows[ARow].Clear;
    StgCustoMaq.RowCount := 2;
    for ARow := (StgResultFinalEst.RowCount - 1) downto 0 do
        StgResultFinalEst.Rows[ARow].Clear;
    StgResultFinalEst.ColCount := 5;
    StgResultFinalEst.RowCount := 2;
    with StgCustoOpt do
        case opCab of
            0: begin
                ColCount := 7;
                Cells[0,0] := 'Ano/Talhão';ColWidths[0] := 148;
                Cells[1,0] := 'Início (dd/mm/aa)';ColWidths[1] := 93;
                Cells[2,0] := 'Termino (dd/mm/aa)';ColWidths[2] := 104;
                Cells[3,0] := 'Atraso (dias)';ColWidths[3] := 65;
                Cells[4,0] := 'Nº de dias úteis';ColWidths[4] := 81;
                Cells[5,0] := 'Perda total (sc)';ColWidths[5] := 78;
                Cells[6,0] := 'Renda total (sc)';ColWidths[6] := 82;
            end;
            1: begin
                ColCount := 9;
                Cells[0,0] := 'Data (dd/mm/aa)';ColWidths[0] := 148;
                Cells[1,0] := 'Precipitação (mm)';ColWidths[1] := 120;
                Cells[2,0] := 'Situação';ColWidths[2] := 49;
                Cells[3,0] := 'Atraso (dias)';ColWidths[3] := 65;
                Cells[4,0] := 'Perda física diária (sc)';ColWidths[4] := 112;
                Cells[5,0] := 'Perda física acumulada (sc)';ColWidths[5] := 139;
                Cells[6,0] := 'Renda física diária (sc)';ColWidths[6] := 116;
                Cells[7,0] := 'Renda física acumulada (sc)';ColWidths[7] := 143;
                Cells[8,0] := 'Área colhida (ha)';ColWidths[8] := 87;
            end;
        end;
    end;
end;

procedure TFrmCenAlterar.RecuperarCenario(Sender: TObject);
var t, a, c, numTal, numAnos: Integer;

```

```

    custoDeSM: Real;
begin
    codCen := FrmSimul.QryPropCen.FieldValues['CódCenário'];
    codProp := FrmSimul.QryPropCen.FieldValues['CódPropriedade'];
    LblCod.Caption := codCen;
    TxtProp.Text := codProp;
    with FrmSimul.TblProp do
    begin
        FindKey ([codProp]);
        Edit;
        FrmReg.TblReg.FindKey ([FieldValues['CódRegião']]);
    end;
    with FrmSimul.QrySimulTal do
    begin
        Close;
        SQL.Clear;
        SQL.Add ('select sum (ÁreaDeLavoura) as ÁreaTotal');
        SQL.Add ('from "SIMULAÇÃO_Talhão"');
        SQL.Add ('where CódSimulação like "' + codProp + '"');
        Open;
        areaTotalProp := FieldValues['ÁreaTotal'];
    end;
    with FrmSimul.QrySimul do
    begin
        Close;
        SQL.Clear;
        SQL.Add ('select *');
        SQL.Add ('from "SIMULAÇÃO"');
        SQL.Add ('where CódSimulação like "' + codCen + '"');
        Open;
        turnoTrab := FieldValues['TurnoDeTrabalho'];
        precCrit := FieldValues['PrecipitaçãoCrítica'];
        TxtTurno.Text := FieldValues['TurnoDeTrabalho'];
        TxtPrec.Text := FieldValues['PrecipitaçãoCrítica'];
    end;
    with FrmSimul.QrySimulInd do
    begin
        PrecSaco := FieldValues['PreçoDoSaco'];
        Juros := FieldValues['TaxaDeJuros'];
        VidaUtil := FieldValues['VidaÚtil'];
        Eficiencia := FieldValues['Eficiência'];
        Salario := FieldValues['Salário'];
        SegAloj := FieldValues['TaxaDeSeguroAlojamento'];
        Diesel := FieldValues['PreçoDoDiesel'];
        Residual := FieldValues['ValorResidual'];
    end;
    with FrmSimul.QrySimulCol do
    begin
        Close;
        SQL.Clear;
        SQL.Add ('select *');
        SQL.Add ('from "SIMULAÇÃO_Colhedora"');
        SQL.Add ('where CódSimulação like "' + codCen + '"');
        Open;
        First;
        numCol := RecordCount;
    end;
end;

```

```

end;
ListarColhedoras (self);
with FrmSimul.QrySimulTal do
begin
    Close;
    SQL.Clear;
    SQL.Add ('select *');
    SQL.Add ('from "SIMULAÇÃO_Talhão"');
    SQL.Add ('where CódSimulação like "' + codCen + '"');
    Open;
    First;
    numTal := RecordCount;
    areaTotalCen := 0;
end;
LimparListaTal (self);
with FrmSimul.QryResultPropAno do
begin
    Close;
    SQL.Clear;
    SQL.Add ('select *');
    SQL.Add ('from "RESULTADO_PropAno"');
    SQL.Add ('where CódSimulação like "' + codCen + '"');
    Open;
    First;
    numAnos := RecordCount div numTal;
end;
LimparCenario (0);
with FrmSimul.QryResultMaqAno do
begin
    Close;
    SQL.Clear;
    SQL.Add ('select *');
    SQL.Add ('from "RESULTADO_MaqAno"');
    SQL.Add ('where CódSimulação like "' + codCen + '"');
    Open;
    First;
end;
SetLength (linAno, numAnos);
SetLength (detAno, numAnos);
SetLength (colAno, numAnos);
SetLength (resultPropAno, numTal);
SetLength (resultMaqAno, numTal);
SetLength (resultFinalTal, numTal);
for t := 1 to numTal do
begin
    SetLength (resultPropAno[t - 1], numAnos);
    SetLength (resultMaqAno[t - 1], numAnos);
    RecuperarCenTal (FrmReg.TblReg.FieldValues['CódRegião'],
FrmSimul.QrySimulTal.FieldValues['CódCultivar']);
    for a := 0 to (numAnos - 1) do
begin
    SetLength (resultMaqAno[t - 1,a], numCol);
    with FrmSimul.QryResultPropAno do
begin
    if ((t - 1) = 0) then
begin

```

```

    detAno[a,0] := FieldValues['Ano'];
    detAno[a,1] := 1;
    detAno[a,2] := 0;
    detAno[a,3] := 0;
    detAno[a,4] := 0;
    detAno[a,5] := 0;
end;
resultPropAno[t - 1,a,0] := FieldValues['PerdaAcumulada'];
resultPropAno[t - 1,a,1] := FieldValues['ColheitaAcumulada'];
resultPropAno[t - 1,a,2] := FieldValues['PrazoDeColheita'];
resultPropAno[t - 1,a,3] := FieldValues['NúmeroDeDiasÚteis'];
Next;
end;
for c := 0 to (numCol - 1) do
    with FrmSimul.QryResultMaqAno do
        begin
            if ((t - 1) = 0) then
                begin
                    SetLength (colAno[a], numCol);
                    colAno[a,c] := 0;
                end;
            resultMaqAno[t - 1,a,c,0] := FieldValues['NúmeroDeColhedoras'];
            resultMaqAno[t - 1,a,c,1] := FieldValues['FatorDeUtilProdutivo'];
            resultMaqAno[t - 1,a,c,2] := FieldValues['CustoProdutivo'];
            resultMaqAno[t - 1,a,c,3] := FieldValues['CustoImprodutivo'];
            resultMaqAno[t - 1,a,c,4] := FieldValues['CustoUnitário'];
            Next;
        end;
    end;
    FrmSimul.QrySimulTal.Next;
end;
StgCenTal.RowCount := StgCenTal.RowCount - 1;
for a := 0 to (numAnos - 1) do
    begin
        with StgCustoOpt do
            begin
                linAno[a,0] := RowCount - 1;
                Cells[0,RowCount - 1] := 'ANO - ' + FloatToStr (detAno[a,0]);
                RowCount := RowCount + 2;
                for t := 1 to numTal do
                    begin
                        Cells[0,RowCount - 1] := 'Talhão ' + letra[t];
                        if (talCultivar[t,1] <> 0) then
                            begin
                                Cells[1,RowCount - 1] := FormatDateTime ('dd/mm/' + Copy (FloatToStr (detAno[a,0]), 3, 2),
                                StrToDate (StgCenTal.Cells[7,t]));
                                Cells[2,RowCount - 1] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[1,RowCount - 1]) +
                                resultPropAno[t - 1,a,2]);
                            end
                        else
                            begin
                                Cells[1,RowCount - 1] := 'Não encontrado';
                                Cells[2,RowCount - 1] := 'Não encontrado';
                            end;
                        Cells[3,RowCount - 1] := FloatToStr (StrToInt (StgCenTal.Cells[9,t]) + resultPropAno[t - 1,a,2]);
                        Cells[4,RowCount - 1] := FloatToStr (resultPropAno[t - 1,a,3]);
                    end
                end
            end
        end
    end
end;

```

```

Cells[5,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,a,0]]);
Cells[6,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,a,1]]);
detAno[a,2] := detAno[a,2] + resultPropAno[t - 1,a,3] * turnoTrab;
detAno[a,3] := detAno[a,3] + resultPropAno[t - 1,a,0];
detAno[a,4] := detAno[a,4] + resultPropAno[t - 1,a,1];
RowCount := RowCount + 1;
end;
RowCount := RowCount + 5;
Cells[0,RowCount - 6] := 'RESULTADOS';
Cells[0,RowCount - 5] := 'Perda total (US$):';Cells[1,RowCount - 5] := Format ('%n', [detAno[a,3] *
PrecSaco]);
Cells[0,RowCount - 4] := 'Renda total (US$):';Cells[1,RowCount - 4] := Format ('%n', [detAno[a,4] *
PrecSaco]);
Cells[0,RowCount - 3] := 'Tempo de utilização anual (h):';Cells[1,RowCount - 3] := Format ('%n',
[detAno[a,2]]);
end;
with StgCustoMq do
begin
linAno[a,1] := RowCount - 1;
Cells[0,RowCount - 1] := 'ANO - ' + FloatToStr (detAno[a,0]);
RowCount := RowCount + 2;
for c := 0 to (numCol - 1) do
begin
RowCount := RowCount + 6;
Cells[0,RowCount - 7] := 'Marca: ' + modColhedora[indColhedora[c],0];
Cells[0,RowCount - 6] := 'Modelo: ' + modColhedora[indColhedora[c],1];
Cells[0,RowCount - 5] := 'Plataforma: ' + Format ('%n m', [codColhedora[indColhedora[c],3]]);
Cells[0,RowCount - 4] := 'Velocidade: ' + Format ('%n km/h', [codColhedora[indColhedora[c],1]]);
Cells[0,RowCount - 3] := 'Número de colhedoras: ' + FloatToStr (resultMqAno[0,a,c,0]);
Cells[0,RowCount - 2] := 'Capacidade: ' + Format ('%n ha/dia', [(codColhedora[indColhedora[c],1]
* resultMqAno[0,a,c,0] * codColhedora[indColhedora[c],3] * (Eficiencia / 1000)) * turnoTrab]);
Cells[0,RowCount - 1] := 'RESULTADOS';
custoDeSM := 0;
for t := 1 to numTal do
begin
RowCount := RowCount + 1;
Cells[0,RowCount - 1] := 'Talhão ' + letra[t];
Cells[1,RowCount - 1] := Format ('%n', [resultMqAno[t - 1,a,c,1]]);
Cells[2,RowCount - 1] := Format ('%n', [1 - resultMqAno[t - 1,a,c,1]]);
Cells[3,RowCount - 1] := Format ('%n', [resultMqAno[t - 1,a,c,2]]);
Cells[4,RowCount - 1] := Format ('%n', [resultMqAno[t - 1,a,c,3]]);
Cells[5,RowCount - 1] := Format ('%n', [resultMqAno[t - 1,a,c,2] + resultMqAno[t - 1,a,c,3]]);
Cells[6,RowCount - 1] := Format ('%n', [resultMqAno[t - 1,a,c,4]]);
custoDeSM := custoDeSM + resultMqAno[t - 1,a,c,2] + resultMqAno[t - 1,a,c,3];
colAno[a,c] := colAno[a,c] + resultMqAno[t - 1,a,c,4];
end;
RowCount := RowCount + 4;
detAno[a,5] := detAno[a,5] + custoDeSM;
colAno[a,c] := colAno[a,c] / numTal;
Cells[0,RowCount - 4] := 'SERVIÇO MECANIZADO (US$):';Cells[1,RowCount - 4] := Format
('%n', [custoDeSM]);
Cells[0,RowCount - 3] := 'CUSTO M. COLHEDORA (US$/sc):';Cells[1,RowCount - 3] := Format
('%n', [colAno[a,c]]);
end;
detAno[a,5] := PrecSaco * (detAno[a,4] - detAno[a,3]) - detAno[a,5];

```

```

Cells[0,RowCount - 1] := 'RENDA LÍQUIDA (US$)';Cells[1,RowCount - 1] := Format ('%n',
[detAno[a,5]]);
RowCount := RowCount + 2;
end;
end;
StgCustoOpt.RowCount := StgCustoOpt.RowCount - 2;
StgCustoMaq.RowCount := StgCustoMaq.RowCount - 2;
priAno := Trunc (detAno[0,0]);
ultAno := Trunc (detAno[numAnos - 1,0]);
TxtPriAno.Text := IntToStr (priAno);
TxtUltAno.Text := IntToStr (ultAno);
PgCtrlCenAlterar.ActivePageIndex := 0;
ResultadoFinal (self);
end;

procedure TFrmCenAlterar.RecuperarCenTal (codReg, codCul: Integer);
var ACol, aln: Integer;
begin
FrmCultivar.TblCul.FindKey ([codCul]);
with FrmCultivar.QryRegCul do
begin
Close;
SQL.Clear;
SQL.Add ('select *');
SQL.Add ('from "REGIÃO_CULTIVAR"');
SQL.Add ('where CódRegião = ' + IntToStr (codReg));
SQL.Add ('and CódCultivar = ' + IntToStr (codCul));
Open;
FindFirst;
end;
with StgCenTal do
begin
Cells[0,RowCount - 1] := letra[RowCount - 1];
Cells[1,RowCount - 1] := FrmCultivar.TblCul.FieldValues['NomeCultivar'];
Cells[2,RowCount - 1] := FrmCultivar.TblCul.FieldValues['GrupoDeMaturação'];
Cells[3,RowCount - 1] := FrmSimul.QrySimulTal.FieldValues['ÁreaDeLavoura'];
areaTotalCen := areaTotalCen + FrmSimul.QrySimulTal.FieldValues['ÁreaDeLavoura'];
talCultivar[RowCount - 1,0] := 0;
if Not (FrmCultivar.QryRegCul.Found) then
begin
for ACol := 4 to 10 do
Cells[ACol,RowCount - 1] := 'Não encontrado';
MessageDlg ('A função de perdas à qual o talhão ' + letra[RowCount - 1] + ' está vinculado foi
removida.',
MTWarning, [MbOK], 0);
talCultivar[RowCount - 1,1] := 0;
end
else
begin
Cells[4,RowCount - 1] := FrmCultivar.QryRegCul.FieldValues['CicloDeMaturação'];
Cells[5,RowCount - 1] := FormatDateTime ('dd/mm/' + Copy (IntToStr (StrToInt (FormatDateTime
('yyyy', Date)) - 1), 3, 2),
FrmSimul.QrySimulTal.FieldValues['DataDeSemeadura']);
Cells[6,RowCount - 1] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[5,RowCount - 1]) +
StrToInt (Cells[4,RowCount - 1]));

```

```

Cells[7,RowCount - 1] := FormatDateTime ('dd/mm/' + FormatDateTime ('yy', Date),
FrmSimul.QrySimulTal.FieldValues['DataDeInícioDaColheita']);
aln := Trunc (StrToDate (Cells[7,RowCount - 1]) - (StrToDate (Cells[6,RowCount - 1]) + 3));
if (aln < 1) then
    aln := 0;
Cells[8,RowCount - 1] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[7,RowCount - 1]) +
(FrmSimul.QrySimulTal.FieldValues['AtrasoPrevisto'] - aln));
Cells[9,RowCount - 1] := FloatToStr (aln);
Cells[10,RowCount - 1] := FrmSimul.QrySimulTal.FieldValues['AtrasoPrevisto'];
talCultivar[RowCount - 1,1] := codCul;
talCultivar[RowCount - 1,2] := FrmCultivar.QryRegCul.FieldValues['CoeficienteDePerdA'];
talCultivar[RowCount - 1,3] := FrmCultivar.QryRegCul.FieldValues['CoeficienteDePerdB'];
talCultivar[RowCount - 1,4] := FrmCultivar.QryRegCul.FieldValues['CoeficienteDePerdC'];
talCultivar[RowCount - 1,5] := FrmCultivar.QryRegCul.FieldValues['CoeficienteDeProdC'];
end;
RowCount := RowCount + 1;
end;
end;

procedure TFrmCenAlterar.DBLookupCboCulClick(Sender: TObject);
begin
    StgCenTal.Cells[2,StgCenTal.Row] := FrmCultivar.TblCul.FieldValues['GrupoDeMaturação'];
    talCultivar[StgCenTal.Row,0] := 1;
    if PesquisarFuncPerdas (FrmReg.TblReg.FieldValues['CódRegião'], DBLookupCboCul.KeyValue)
then
        with FrmCultivar.QryRegCul do
            begin
                talCultivar[StgCenTal.Row,1] := DBLookupCboCul.KeyValue;
                talCultivar[StgCenTal.Row,2] := FieldValues['CoeficienteDePerdA'];
                talCultivar[StgCenTal.Row,3] := FieldValues['CoeficienteDePerdB'];
                talCultivar[StgCenTal.Row,4] := FieldValues['CoeficienteDePerdC'];
                talCultivar[StgCenTal.Row,5] := FieldValues['CoeficienteDeProdC'];
            end
        else
            talCultivar[StgCenTal.Row,1] := 0;
        end;
end;

procedure TFrmCenAlterar.DBLookupCboCulExit(Sender: TObject);
var resp: Word;
    continua: Boolean;
begin
    continua := false;
    if (talCultivar[StgCenTal.Row,0] = 1) then
        begin
            if (talCultivar[StgCenTal.Row,1] <> 0) then
                StgCenTal.Cells[1,StgCenTal.Row] := DBLookupCboCul.Text
            else
                begin
                    resp := MessageDlg ('O cultivar é inválido. Pressione OK para selecionar outro ou Cancel para
excluir o talhão.',
                                MTErr, [MbOK,MbCancel], 0);
                    if (resp = mrOK) then
                        begin
                            continua := true;
                            DBLookupCboCul.SetFocus;
                        end
                    end
                end
            end
        end
end

```

```

        else
            ExcluirTal (self);
        end;
    end;
    if Not (continua) then
    begin
        DBLookupCboCul.Visible := false;
        StgCenTal.SetFocus;
    end;
end;

procedure TFrmCenAlterar.StgCenTalSelectCell(Sender: TObject; ACol,
    ARow: Integer; var CanSelect: Boolean);
var cod: Real;
    R: TRect;
begin
    if (ACol = 1) then
    begin
        R := StgCenTal.CellRect (ACol,ARow);
        R.Left := R.Left + StgCenTal.Left;
        R.Right := R.Right + StgCenTal.Left;
        R.Top := R.Top + StgCenTal.Top;
        R.Bottom := R.Bottom + StgCenTal.Top;
        with DBLookupCboCul do
        begin
            Left := R.Left + 1;
            Top := R.Top + 1;
            Width := (R.Right + 1) - R.Left;
            Height := (R.Bottom + 1) - R.Top;
            if (talCultivar[ARow,1] <> 0) then
                cod := talCultivar[ARow,1]
            else
                cod := 1;
            FrmCultivar.TblCul.Edit;
            KeyValue := cod;
            Visible := true;
            SetFocus;
        end;
    end;
end;

procedure TFrmCenAlterar.BtnConfirmarClick(Sender: TObject);
var t, anoAT, c, cs: Integer;
begin
    Screen.Cursor := crHourglass;
    try try
        if (TxtTurno.Text = "") then
            raise Exception.Create ('Preencha o campo "Turno de trabalho".');
        turnoTrab := StrToFloat (TxtTurno.Text);
        if (turnoTrab > FrmReg.TblReg.FieldValues['TurnoDeTrabalho']) then
            raise Exception.Create ('O turno de trabalho é inválido')
        else if ((numCol = 0) or (numCol > 1)) then
            raise Exception.Create ('Deve haver uma colhedora selecionada');
        areaTotalCen := 0;
        for t := 1 to (StgCenTal.RowCount - 1) do
            begin

```

```

if Not (talCultivar[t,1] <> 0) then
    raise Exception.Create ('Nenhum cultivar foi atribuído ao talhão ' + letra[t])
else if (StgCenTal.Cells[3,t] = "") then
    raise Exception.Create ('Forneça a área de lavoura do talhão ' + letra[t])
else if (StgCenTal.Cells[5,t] = "") then
    raise Exception.Create ('Forneça a data de semeadura do talhão ' + letra[t])
else if ((t > 1) and (StrToDate (StgCenTal.Cells[7,t]) <= StrToDate (StgCenTal.Cells[8,t - 1]))) then
    raise Exception.Create ('O período de colheita do talhão ' + letra[t] + ' coincide com o do talhão ' +
letra[t - 1] + '.');
    areaTotalCen := areaTotalCen + StrToFloat (StgCenTal.Cells[3,t]);
end;
if (areaTotalCen > areaTotalProp) then
    raise Exception.Create ('A área total do cenário é maior que a área da propriedade da qual se
deseja gerá-lo.');
```

```

SetLength (indColhedora, numCol);
SetLength (linAno, (ultAno + 1) - priAno);
SetLength (detAno, (ultAno + 1) - priAno);
SetLength (colAno, (ultAno + 1) - priAno);
SetLength (resultPropAno, StgCenTal.RowCount - 1);
SetLength (resultMaqAno, StgCenTal.RowCount - 1);
SetLength (resultFinalTal, StgCenTal.RowCount - 1);
c := 0;
for cs := 0 to (ChkListBoxCenCol.Items.Count - 1) do
    if ChkListBoxCenCol.Checked[cs] then
        begin
            indColhedora[c] := cs;
            c := c + 1;
        end;
for t := 1 to (StgCenTal.RowCount - 1) do
    begin
        SetLength (resultPropAno[t - 1], (ultAno + 1) - priAno);
        SetLength (resultMaqAno[t - 1], (ultAno + 1) - priAno);
        for anoAT := priAno to ultAno do
            begin
                SetLength (resultMaqAno[t - 1, (anoAT - priAno)], numCol);
                if ((t - 1) = 0) then
                    begin
                        SetLength (colAno[anoAT - priAno], numCol);
                        detAno[anoAT - priAno, 0] := anoAT;
                        detAno[anoAT - priAno, 1] := 1;
                        detAno[anoAT - priAno, 2] := 0;
                        detAno[anoAT - priAno, 3] := 0;
                        detAno[anoAT - priAno, 4] := 0;
                        detAno[anoAT - priAno, 5] := 0;
                        for c := 0 to (numCol - 1) do
                            colAno[anoAT - priAno, c] := 0;
                        end;
                    end;
            end;
        end;
    end;
    LimparCenario (1);
    FuncPerdas (self);
    ServMecanizado (self);
    ResultadoFinal (self);
    cenAlterado := true;
    MessageDlg ('O cenário foi alterado com êxito.', MTInformation, [MbOK], 0);
finally

```

```

    Screen.Cursor := crDefault;
end;
except
    on EConvertError do
        raise Exception.Create ('O valor do campo é inválido');
    end;
end;

procedure TFrmCenAlterar.StgCenTalGetEditText(Sender: TObject; ACol,
    ARow: Integer; var Value: String);
var R: TRect;
begin
    valCell := StgCenTal.Cells[ACol,ARow];
    R := StgCenTal.CellRect (ACol,ARow);
    R.Left := R.Left + StgCenTal.Left;
    R.Right := R.Right + StgCenTal.Left;
    R.Top := R.Top + StgCenTal.Top;
    R.Bottom := R.Bottom + StgCenTal.Top;
    with TxtCell do
        begin
            case ACol of
                0, 1, 2, 3, 4, 9, 10: EditMask := '';
                5, 6, 7, 8: EditMask := '!99/99/99;1;_';
            end;
            Left := R.Left + 2;
            Top := R.Top + 2;
            Width := R.Right - R.Left;
            Height := R.Bottom - R.Top;
            Text := valCell;
            Visible := true;
            SetFocus;
        end;
    end;
end;

procedure TFrmCenAlterar.TxtCellExit(Sender: TObject);
var t, atraso: Integer;
    valido: Boolean;
    data: TDateTime;
begin
    try try
        case StgCenTal.Col of
            0, 1, 2, 4, 6, 9, 10: begin
                StgCenTal.Cells[StgCenTal.Col,StgCenTal.Row] := valCell;
                if (valCell <> TxtCell.Text) then
                    MessageDlg ('O valor não pode ser alterado porque o campo "' +
                        StgCenTal.Cells[StgCenTal.Col,0] + '" está aberto somente para leitura.',
                        MTError, [MbOK], 0);
                end;
            3: StgCenTal.Cells[3,StgCenTal.Row] := FloatToStr (StrToFloat (TxtCell.Text));
            5: with StgCenTal do
                begin
                    valido := true;
                    data := StrToDate (TxtCell.Text);
                    if (data < StrToDate (FormatDateTime ('dd/mm/' + Copy (IntToStr (StrToInt (FormatDateTime
                        ('yyyy', Date)) - 1), 3, 2), FrmReg.TblReg.FieldValues['PriDataDeSemeadura'])))

```

```

    or (data > StrToDate (FormatDateTime ('dd/mm/' + Copy (IntToStr (StrToInt (FormatDateTime
('yyyy', Date)) - 1), 3, 2), FrmReg.TblReg.FieldValues['UltDataDeSemeadura']))) then
        valido := false
    else
        for t := 1 to (RowCount - 1) do
            if ((t <> Row) and (talCultivar[t,1] = talCultivar[Row,1]) and (Cells[5,t] = TxtCell.Text)) then
                valido := false;
        if valido then
            begin
                Cells[5,Row] := TxtCell.Text;
                Cells[6,Row] := FormatDateTime ('dd/mm/yy', data + StrToInt (Cells[4,Row]));
                Cells[7,Row] := Cells[6,Row];
                Cells[8,Row] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[7,Row]) + 24);
                Cells[9,Row] := '0';
                Cells[10,Row] := '24';
                t := -1;
                if ((Row > 1) and (Cells[5,Row - 1] <> '') and (StrToDate (Cells[7,Row]) <= StrToDate
(Cells[8,Row - 1]))) then
                    t := Row - 1
                else if ((Row < (RowCount - 1)) and (Cells[5,Row + 1] <> '') and (StrToDate (Cells[7,Row + 1])
<= StrToDate (Cells[8,Row]))) then
                    t := Row + 1;
                if (t >= 0) then
                    MessageDlg ('O período de colheita do talhão ' + letra[Row] + ' coincide com o do talhão ' +
letra[t] + '.',
                        MTWarning, [MbOK], 0);
            end
        else
            begin
                Cells[5,Row] := valCell;
                MessageDlg ('A data de semeadura é inválida.', MTErrors, [MbOK], 0);
            end
        end;
7: with StgCenTal do
    begin
        data := StrToDate (TxtCell.Text);
        if (data < (StrToDate (Cells[6,Row]) - 3))
        or (data > (StrToDate (Cells[6,Row]) + 27))
        or (data > StrToDate (Cells[8,Row]))
        or ((StrToDate (Cells[8,Row]) - data) > 44) then
            begin
                Cells[7,Row] := valCell;
                MessageDlg ('A data de início da colheita é inválida.', MTErrors, [MbOK], 0);
            end
        else
            begin
                atraso := 0;
                if (data > (StrToDate (Cells[6,Row]) + 3)) then
                    begin
                        atraso := Trunc (data - (StrToDate (Cells[6,Row]) + 3));
                        MessageDlg ('A data ' + TxtCell.Text + ' excede em ' + FloatToStr (atraso) + ' dia(s) o limite
máximo para o início provável da colheita.',
                            MTWarning, [MbOK], 0);
                    end
                end;
                Cells[7,Row] := TxtCell.Text;
                Cells[9,Row] := FloatToStr (atraso);

```

```

        Cells[10,Row] := FloatToStr ((StrToDate (Cells[8,Row]) + atraso) - data);
        if ((Row > 1) and (Cells[5,Row - 1] <> '') and (data <= StrToDate (Cells[8,Row - 1]))) then
            MessageDlg ('O período de colheita do talhão ' + letra[Row] + ' coincide com o do talhão ' +
letra[Row - 1] + '.',
                MTWarning, [MbOK], 0);
        end;
    end;
8: with StgCenTal do
begin
    data := StrToDate (TxtCell.Text);
    atraso := Trunc ((data + StrToInt (Cells[9,Row])) - StrToDate (Cells[7,Row]));
    if ((data < StrToDate (Cells[7,Row])) or (atraso > 44)) then
        begin
            Cells[8,Row] := valCell;
            MessageDlg ('A data de término da colheita é inválida.', MError, [MbOK], 0);
        end
    else
        begin
            Cells[8,Row] := TxtCell.Text;
            Cells[10,Row] := FloatToStr (atraso);
            if ((Row < (RowCount - 1)) and (Cells[5,Row + 1] <> '') and (StrToDate (Cells[7,Row + 1]) <=
data)) then
                MessageDlg ('O período de colheita do talhão ' + letra[Row] + ' coincide com o do talhão ' +
letra[Row + 1] + '.',
                    MTWarning, [MbOK], 0);
            end;
        end;
    end;
finally
    TxtCell.Visible := false;
    StgCenTal.SetFocus;
end;
except
on EConvertError do
begin
    StgCenTal.Cells[StgCenTal.Col,StgCenTal.Row] := valCell;
    MessageDlg ('O valor do campo é inválido.', MError, [MbOK], 0);
end;
end;
end;

procedure TFrmCenAlterar.DBLookupCboCulKeyPress(Sender: TObject;
var Key: Char);
begin
    if (Key = #13) then
        begin
            Key := #0;
            Perform (CM_DialogKey, VK_TAB, 0);
        end;
end;

procedure TFrmCenAlterar.TxtCellKeyPress(Sender: TObject; var Key: Char);
begin
    if (Key = #13) then
        begin
            Key := #0;

```

```

    Perform (CM_DialogKey, VK_TAB, 0);
end;
end;

procedure TFrmCenAlterar.FuncPerdas(Sender: TObject);
var t, anoAT, aln, aPrev, aFi, atrasoStart, atraso, diasUT, ARow: Integer;
    pAN, pAT, prodMax, ritmoTrab, r, perdTotal, prodTotal, areaC: Real;
    sitData: array of Integer;
begin
    for anoAT := priAno to ultAno do
    begin
        linAno[anoAT - priAno,0] := StgCustoOpt.RowCount - 1;
        StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 1] := 'ANO - ' + IntToStr (anoAT);
        StgCustoOpt.RowCount := StgCustoOpt.RowCount + 2;
        for t := 1 to (StgCenTal.RowCount - 1) do
        begin
            StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 1] := 'TALHÃO ' + letra[t];
            aln := StrToInt (StgCenTal.Cells[9,t]);
            aPrev := StrToInt (StgCenTal.Cells[10,t]);
            pAT := talCultivar[t,4];
            atrasoStart := -1;
            repeat
                atrasoStart := atrasoStart + 1;
                pAN := pAT;
                pAT := talCultivar[t,2] * Sqr (atrasoStart) + talCultivar[t,3] * atrasoStart + talCultivar[t,4];
            until (pAT > pAN);
            atrasoStart := atrasoStart - 1;
            prodMax := (talCultivar[t,2] * -1) * Sqr(atrasoStart) + (talCultivar[t,3] * -1) * atrasoStart +
            talCultivar[t,5];
            with FrmSimul.QryPrec do
            begin
                Close;
                SQL.Clear;
                SQL.Add ('select *');
                SQL.Add ('from "REGIÃO_Precipitação");
                SQL.Add ('where CódRegião = ' + IntToStr (FrmReg.TblReg.FieldValues['CódRegião']));
                SQL.Add ('and Data between "' + FormatDateTime ('mm/dd/' + IntToStr (anoAT), StrToDate
                (StgCenTal.Cells[7,t])) + '"');
                SQL.Add ('and "' + FormatDateTime ('mm/dd/' + IntToStr (anoAT), StrToDate
                (StgCenTal.Cells[8,t])) + '"');
                Open;
                First;
            end;
            aFi := 0;
            diasUT := 0;
            pAT := 0;
            SetLength (sitData, (aPrev + 1) - aln);
            for atraso := 0 to (aPrev - aln) do
            begin
                pAN := pAT;
                pAT := FrmSimul.QryPrec.FieldValues['Precipitação'];
                if ((pAT + pAN) <= precCrit) then
                begin
                    sitData[atraso] := 1;
                    aFi := atraso;
                    diasUT := diasUT + 1;
                end;
            end;
        end;
    end;
end;

```

```

end
else
    sitData[atraso] := 0;
    StgCustoOpt.RowCount := StgCustoOpt.RowCount + 1;
    StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 1] := FormatDateTime ('dd/mm/yy',
FrmSimul.QryPrec.FieldValues['Data']);
    StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 1] := Format ('%n', [pAT]);
    FrmSimul.QryPrec.Next;
end;
if (diasUT > 0) then
begin
    detAno[anoAT - priAno,2] := detAno[anoAT - priAno,2] + diasUT * turnoTrab;
    ritmoTrab := StrToFloat (StgCenTal.Cells[3,t]) / diasUT;
end
else
begin
    detAno[anoAT - priAno,1] := 0;
    ritmoTrab := 0;
end;
perdTotal := 0;
prodTotal := 0;
areaC := 0;
for atraso := aln to aPrev do
begin
    ARow := StgCustoOpt.RowCount - ((aPrev + 1) - atraso);
    if (sitData[atraso - aln] = 1) then
begin
        r := ritmoTrab;
        StgCustoOpt.Cells[2,ARow] := 'Útil';
end
else
begin
        r := 0;
        StgCustoOpt.Cells[2,ARow] := 'Inútil';
end;
if (atraso > atrasoStart) then
    pAT := talCultivar[t,2] * Sqr(atraso) + talCultivar[t,3] * atraso + talCultivar[t,4]
else
    pAT := 0;
perdTotal := perdTotal + pAT * r;
prodTotal := prodTotal + (prodMax - pAT) * r;
areaC := areaC + r;
StgCustoOpt.Cells[3,ARow] := IntToStr (atraso);
StgCustoOpt.Cells[4,ARow] := Format ('%n', [pAT * r]);
StgCustoOpt.Cells[5,ARow] := Format ('%n', [perdTotal]);
StgCustoOpt.Cells[6,ARow] := Format ('%n', [(prodMax - pAT) * r]);
StgCustoOpt.Cells[7,ARow] := Format ('%n', [prodTotal]);
StgCustoOpt.Cells[8,ARow] := Format ('%n', [areaC]);
end;
detAno[anoAT - priAno,3] := detAno[anoAT - priAno,3] + perdTotal;
detAno[anoAT - priAno,4] := detAno[anoAT - priAno,4] + prodTotal;
resultPropAno[t - 1,(anoAT - priAno),0] := perdTotal;
resultPropAno[t - 1,(anoAT - priAno),1] := prodTotal;
resultPropAno[t - 1,(anoAT - priAno),2] := aFi;
resultPropAno[t - 1,(anoAT - priAno),3] := diasUT;
StgCustoOpt.RowCount := StgCustoOpt.RowCount + 9;

```

```

    StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 9] := 'RESULTADOS (talhão ' + letra[t] + ')';
    StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 8] := 'Início da colheita: ';
    StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 8] := FormatDateTime ('dd/mm/' + Copy (IntToStr
(anoAT), 3, 2), StrToDate (StgCenTal.Cells[7,t]));
    StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 7] := 'Término da colheita: ';
    StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 6] := 'Nº de dias úteis: ';
    StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 5] := 'Ritmo de trabalho (ha/dia): ';
    StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 4] := 'Perda total (US$): ';
    StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 3] := 'Renda total (US$): ';
    if (ritmoTrab > 0) then
    begin
        StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 7] := FormatDateTime ('dd/mm/yy', StrToDate
(StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 8]) + aFi);
        StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 6] := IntToStr (diasUT);
        StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 5] := Format ('%n', [ritmoTrab]);
        StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 4] := Format ('%n', [perdTotal * PrecSaco]);
        StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 3] := Format ('%n', [prodTotal * PrecSaco]);
    end
    else
    begin
        StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 7] := 'Colheita não realizada';
        StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 6] := 'Nenhum dia foi constatado';
        StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 5] := 'Colheita não realizada';
        StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 4] := 'Colheita não realizada';
        StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 3] := 'Colheita não realizada';
    end;
end;
with StgCustoOpt do
begin
    RowCount := StgCustoOpt.RowCount + 6;
    Cells[0,RowCount - 7] := 'RESULTADOS (ano de ' + IntToStr (anoAT) + ')';
    Cells[0,RowCount - 6] := 'Perda total (US$):'; Cells[1,RowCount - 6] := Format ('%n', [detAno[anoAT
- priAno,3] * PrecSaco]);
    Cells[0,RowCount - 5] := 'Renda total (US$):'; Cells[1,RowCount - 5] := Format ('%n', [detAno[anoAt
- priAno,4] * PrecSaco]);
    Cells[0,RowCount - 4] := 'Tempo de utilização anual (h):'; Cells[1,RowCount - 4] := Format ('%n',
[detAno[anoAT - priAno,2]]);
    Cells[0,RowCount - 3] := 'Situação do ano: ';
    if (detAno[anoAT - priAno,1] = 1) then
        Cells[1,RowCount - 3] := 'Válido para a estatística'
    else
        Cells[1,RowCount - 3] := 'Inválido para a estatística';
    end;
end;
StgCustoOpt.RowCount := StgCustoOpt.RowCount - 2;
end;

procedure TFrmCenAlterar.ServMecanizado(Sender: TObject);
var t, anoAT, c: Integer;
    numColAN, numColAT, custoDePR, custoDeUT: Real;
    custoPT, custoImPT, custoDeSM, custoDeSMTTotal, custoUnitario: Real;
    capColhedora: Real;
begin
    for anoAT := priAno to ultAno do
    begin
        linAno[anoAT - priAno,1] := StgCustoMaq.RowCount - 1;

```

```

StgCustoMq.Cells[0,StgCustoMq.RowCount - 1] := 'ANO - ' + IntToStr (anoAT);
StgCustoMq.RowCount := StgCustoMq.RowCount + 2;
if (detAno[anoAT - priAno,1] = 1) then
begin
  StgCustoMq.Cells[0,StgCustoMq.RowCount - 3] :=
StgCustoMq.Cells[0,StgCustoMq.RowCount - 3] + ' (válido)';
  for c := 0 to (numCol - 1) do
  begin
    capColhedora := codColhedora[indColhedora[c],1] * codColhedora[indColhedora[c],3] * (Eficiencia
/ 1000) * turnoTrab;
    numColAN := 0;
    for t := 1 to (StgCenTal.RowCount - 1) do
    begin
      numColAT := (StrToFloat (StgCenTal.Cells[3,t]) / resultPropAno[t - 1,anoAT - priAno,3]) /
capColhedora;
      if (numColAT > numColAN) then
        numColAN := numColAT;
      end;
      numColAT := Int (numColAN);
      if ((numColAN - numColAT) > 0) then
        numColAT := numColAT + 1;
      with StgCustoMq do
      begin
        RowCount := RowCount + 6;
        Cells[0,RowCount - 7] := 'Marca: ' + modColhedora[indColhedora[c],0];
        Cells[0,RowCount - 6] := 'Modelo: ' + modColhedora[indColhedora[c],1];
        Cells[0,RowCount - 5] := 'Plataforma: ' + Format ('%n m', [codColhedora[indColhedora[c],3]]);
        Cells[0,RowCount - 4] := 'Velocidade: ' + Format ('%n km/h', [codColhedora[indColhedora[c],1]]);
        Cells[0,RowCount - 3] := 'Número de colhedoras: ' + FloatToStr (numColAT);
        Cells[0,RowCount - 2] := 'Capacidade: ' + Format ('%n ha/dia', [capColhedora * numColAT]);
        Cells[0,RowCount - 1] := 'RESULTADOS';
        custoDeSMTtotal := 0;
        for t := 1 to (StgCenTal.RowCount - 1) do
        begin
          numColAN := (StrToFloat (StgCenTal.Cells[3,t]) / resultPropAno[t - 1,anoAT - priAno,3]) /
capColhedora;
          custoDePR := (codColhedora[indColhedora[c],4] * (1 - Residual / 100) * ((exp (ln (1 + (Juros /
100)) * VidaUtil) * (Juros / 100)) / (exp (ln (1 + (Juros / 100)) * VidaUtil) - 1))) +
((codColhedora[indColhedora[c],4] * (Residual / 100)) * (Juros / 100)) +
(codColhedora[indColhedora[c],4] * (SegAloj / 100));
          custoDeUT := (((Salario / 220) * 2.1) + (0.184 * codColhedora[indColhedora[c],2]) * Diesel +
(((0.00059 * codColhedora[indColhedora[c],2]) + 0.02169) * (Diesel * 4.5)) + (0.4 *
codColhedora[indColhedora[c],4]) / (VidaUtil * detAno[anoAT - priAno,2])) * detAno[anoAT - priAno,2];
          custoPT := ((custoDePR + custoDeUT) * numColAN) * (StrToFloat (StgCenTal.Cells[3,t]) /
areaTotalCen);
          custolmPT := ((custoDePR + (((Salario / 220) * 2.1) * detAno[anoAT - priAno,2])) * (numColAT -
numColAN)) * (StrToFloat (StgCenTal.Cells[3,t]) / areaTotalCen);
          custoDeSM := custoPT + custolmPT;
          custoUnitario := (custoDeSM + (resultPropAno[t - 1,(anoAT - priAno),0] * PrecSaco)) /
resultPropAno[t - 1,(anoAT - priAno),1];
          RowCount := RowCount + 1;
          Cells[0,RowCount - 1] := 'Talhão ' + letra[t];
          Cells[1,RowCount - 1] := Format ('%n', [numColAN / numColAT]);
          Cells[2,RowCount - 1] := Format ('%n', [1 - numColAN / numColAT]);
          Cells[3,RowCount - 1] := Format ('%n', [custoPT]);
          Cells[4,RowCount - 1] := Format ('%n', [custolmPT]);

```

```

Cells[5,RowCount - 1] := Format ('%n', [custoDeSM]);
Cells[6,RowCount - 1] := Format ('%n', [custoUnitario]);
custoDeSMTTotal := custoDeSMTTotal + custoDeSM;
colAno[anoAT - priAno,c] := colAno[anoAT - priAno,c] + custoUnitario;
resultMaqAno[t - 1,(anoAT - priAno),c,0] := numColAT;
resultMaqAno[t - 1,(anoAT - priAno),c,1] := numColAN / numColAT;
resultMaqAno[t - 1,(anoAT - priAno),c,2] := custoPT;
resultMaqAno[t - 1,(anoAT - priAno),c,3] := custoImPT;
resultMaqAno[t - 1,(anoAT - priAno),c,4] := custoUnitario;
end;
RowCount := RowCount + 4;
colAno[anoAT - priAno,c] := colAno[anoAT - priAno,c] / (StgCenTal.RowCount - 1);
Cells[0,RowCount - 4] := 'SERVIÇO MECANIZADO (US$)';Cells[1,RowCount - 4] := Format
('%n', [custoDeSMTTotal]);
Cells[0,RowCount - 3] := 'CUSTO M. COLHEDORA (US$/sc)';Cells[1,RowCount - 3] := Format
('%n', [colAno[anoAT - priAno,c]]);
end;
detAno[anoAT - priAno,5] := PrecSaco * (detAno[anoAT - priAno,4] - detAno[anoAT - priAno,3]) -
custoDeSMTTotal;
StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 1] := 'RENDA LÍQUIDA
(US$)';StgCustoMaq.Cells[1,StgCustoMaq.RowCount - 1] := Format ('%n', [detAno[anoAT -
priAno,5]]);
StgCustoMaq.RowCount := StgCustoMaq.RowCount + 2;
end;
end
else
StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 3] :=
StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 3] + ' (inválido)';
end;
StgCustoMaq.RowCount := StgCustoMaq.RowCount - 2;
end;

procedure TFrmCenAlterar.ResultadoFinal(Sender: TObject);
var a, i, ind, ACol: Integer;
valoresVar: array[0..6] of array of Real;
valoresTal: array of array[0..1] of array of Real;
valoresCol: array of array of Real;
begin
SetLength (valoresTal, StgCenTal.RowCount - 1);
ind := 0;
for a := 0 to High (detAno) do
if (detAno[a,1] = 1) then
begin
for i := 0 to 6 do
SetLength (valoresVar[i], ind + 1);
valoresVar[0,ind] := (detAno[a,3] * PrecSaco) / areaTotalCen;
valoresVar[1,ind] := detAno[a,5] / areaTotalCen;
valoresVar[2,ind] := (detAno[a,4] * PrecSaco - detAno[a,5]) / areaTotalCen;
valoresVar[3,ind] := colAno[a,0];
valoresVar[4,ind] := valoresVar[2,ind] - (detAno[a,3] * PrecSaco) / areaTotalCen;
valoresVar[5,ind] := detAno[a,2];
valoresVar[6,ind] := resultMaqAno[0,a,0,0];
for i := 1 to (StgCenTal.RowCount - 1) do
begin
SetLength (valoresTal[i - 1,0], ind + 1);
SetLength (valoresTal[i - 1,1], ind + 1);

```

```

    valoresTal[i - 1,0,ind] := resultPropAno[i - 1,a,2] + 1;
    valoresTal[i - 1,1,ind] := resultPropAno[i - 1,a,3];
end;
ind := ind + 1;
end;
with StgResultFinalEst do
begin
Cells[0,0] := 'Nº de anos da série:';Cells[1,0] := IntToStr ((ultAno - priAno) + 1);
Cells[0,1] := 'Nº de anos válidos:';Cells[1,1] := IntToStr (ind);
if (ind > 0) then
begin
    RowCount := RowCount + 3;
    Cells[0,RowCount - 2] := 'ESTATÍSTICA';
    for ACol := 0 to 4 do
        Cells[ACol,RowCount - 1] := atributo[ACol];
    for i := 0 to 6 do
    begin
        RowCount := RowCount + 1;
        CalcularMedias (0, i, valoresVar[i]);
        Cells[0,RowCount - 1] := variavel[i];
        for ACol := 1 to 4 do
            Cells[ACol,RowCount - 1] := Format ('%n', [resultFinalVar[i,ACol - 1]]);
        end;
        RowCount := RowCount + 3;
        Cells[0,RowCount - 2] := 'DETALHES DO TALHÃO';
        Cells[0,RowCount - 1] := 'Prazo (dias)';
        for i := 1 to 4 do
        begin
            RowCount := RowCount + 1;
            Cells[0,RowCount - 1] := atributo[i];
        end;
        ind := RowCount;
        RowCount := RowCount + 1;
        Cells[0,RowCount - 1] := 'Nº de dias úteis';
        for i := 1 to 4 do
        begin
            RowCount := RowCount + 1;
            Cells[0,RowCount - 1] := atributo[i];
        end;
        for ACol := 1 to (StgCenTal.RowCount - 1) do
        begin
            if (ACol > (ColCount - 1)) then
                ColCount := ColCount + 1;
            Cells[ACol,ind - 6] := 'Talhão ' + letra[ACol];
            CalcularMedias (1, ACol - 1, valoresTal[ACol - 1,0]);
            CalcularMedias (2, ACol - 1, valoresTal[ACol - 1,1]);
            for i := 0 to 3 do
            begin
                Cells[ACol,ind - (4 - i)] := Format ('%n', [resultFinalTal[ACol - 1,0,i]]);
                Cells[ACol,RowCount - (4 - i)] := Format ('%n', [resultFinalTal[ACol - 1,1,i]]);
            end;
        end;
        RowCount := RowCount + 3;
        Cells[0,RowCount - 2] := 'DISTRIBUIÇÃO DE COLHEDORA';
        Cells[0,RowCount - 1] := 'Nº de colhedoras';
        Cells[1,RowCount - 1] := 'Freq. obs.';
    end;
end;

```

```

Cells[2,RowCount - 1] := 'Freq. acum.';
Cells[3,RowCount - 1] := 'Prob. obs. (%)';
Cells[4,RowCount - 2] := 'F.U. PROD. (%)';
for ACol := 1 to (StgCenTal.RowCount - 1) do
begin
  if ((ACol + 4) > ColCount) then
    ColCount := ColCount + 1;
    Cells[ACol + 3,RowCount - 1] := 'Talhão ' + letra[ACol];
  end;
  DistribuirValores (0, valoresVar[6]);
  SetLength (valoresCol, High (categoria[0]) + 1);
  for ind := High (categoria[0]) downto 0 do
  begin
    SetLength (valoresCol[ind], StgCenTal.RowCount - 1);
    for i := 1 to (StgCenTal.RowCount - 1) do
      valoresCol[ind,i - 1] := 0;
      for a := 0 to High (detAno) do
        if (detAno[a,1] = 1) then
          if (resultMaqAno[0,a,0,0] = categoria[0,ind,0]) then
            for i := 1 to (StgCenTal.RowCount - 1) do
              valoresCol[ind,i - 1] := valoresCol[ind,i - 1] + (resultMaqAno[i - 1,a,0,1] * 100) /
categoria[0,ind,1];
            RowCount := RowCount + 1;
            for ACol := 0 to 3 do
              Cells[ACol,RowCount - 1] := Format ('%n', [categoria[0,ind,ACol]]);
            for ACol := 1 to (StgCenTal.RowCount - 1) do
              Cells[ACol + 3,RowCount - 1] := Format ('%n', [valoresCol[ind,ACol - 1]]);
            end;
          end
        else
          raise Exception.Create ('Não foi possível alterar o cenário porque não há nenhum ano na série
temporal');
        end;
      end;
    end;
  end;
end;

```

```

function TFrmCenAlterar.SalvarCenario: Boolean;
var t, anoAT, c: Integer;
    data: TDateTime;
begin
  data := FrmSimul.QrySimul.FieldValues['DataDeCriação'];
  FrmSimul.ExcluirSimulacao (codCen);
  FrmSimul.QryPropCen.Delete;
  FrmSimul.QryPropCen.InsertRecord ([codCen,codProp]);
  FrmSimul.QrySimul.InsertRecord ([codCen,data,turnoTrab,precCrit,StgCenTal.RowCount - 1]);
  FrmSimul.QrySimulInd.InsertRecord ([codCen,PrecSaco,Juros,VidaUtil,Eficiencia,Salario,
    SegAloj,Diesel,Residual]);
  for c := 0 to (numCol - 1) do
    FrmSimul.QrySimulCol.InsertRecord
([codCen,c,codColhedora[indColhedora[c],0],codColhedora[indColhedora[c],1]]);
  for t := 1 to (StgCenTal.RowCount - 1) do
  begin
    FrmSimul.QrySimulTal.InsertRecord
([codCen,t,talCultivar[t,1],StgCenTal.Cells[5,t],StgCenTal.Cells[7,t],StgCenTal.Cells[10,t],StgCenTal.C
ells[3,t]]);
    for anoAT := priAno to ultAno do
      if (detAno[anoAT - priAno,1] = 1) then

```

```

begin
    FrmSimul.QryResultPropAno.InsertRecord ([codCen,t,anoAT,Null,StrToDate
(StgCenTal.Cells[7,t]) + resultPropAno[t - 1,(anoAT - priAno),2],
                                resultPropAno[t - 1,(anoAT - priAno),0],resultPropAno[t - 1,(anoAT -
priAno),1],
                                resultPropAno[t - 1,(anoAT - priAno),2],resultPropAno[t - 1,(anoAT -
priAno),3]]);
    for c := 0 to (numCol - 1) do
        FrmSimul.QryResultMaqAno.InsertRecord ([codCen,t,anoAT,c,resultMaqAno[t - 1,(anoAT -
priAno),c,0],resultMaqAno[t - 1,(anoAT - priAno),c,1],
                                resultMaqAno[t - 1,(anoAT - priAno),c,2],resultMaqAno[t - 1,(anoAT -
priAno),c,3],resultMaqAno[t - 1,(anoAT - priAno),c,4]]);
    end;
end;
SalvarCenario := true;
FrmSimul.QryPropCen.Refresh;
MessageDlg ('As alterações do cenário foram salvas com êxito.', MTInformation, [MbOK], 0);
end;

procedure TFrmCenAlterar.OpSalvarClick(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    try
        if Not (cenAlterado) then
            raise Exception.Create ('O cenário não foi alterado')
        else if (cenSalvado) then
            raise Exception.Create ('As alterações do cenário já foram salvas')
        else
            cenSalvado := SalvarCenario;
        finally
            Screen.Cursor := crDefault;
        end;
    end;
end;

procedure TFrmCenAlterar.OpIndicadorClick(Sender: TObject);
begin
    try
        FrmIndicador := TFrmIndicador.Create (Application);
        FrmIndicador.ShowModal;
    finally
        FrmIndicador.Free;
    end;
end;

procedure TFrmCenAlterar.FormClose(Sender: TObject; var Action: TCloseAction);
var resp: Word;
begin
    Screen.Cursor := crHourglass;
    try
        if (cenAlterado and Not (cenSalvado)) then
            begin
                resp := MessageDlg ('Deseja salvar as alterações do cenário?', MTWarning,
[MbYes,MbNo,MbCancel], 0);
                if (resp = mrYes) then
                    cenSalvado := SalvarCenario
                else if (resp = mrCancel) then

```

```

        Action := caNone;
    end;
finally
    Screen.Cursor := crDefault;
end;
end;

procedure TFrmCenAlterar.BtnCancelarClick(Sender: TObject);
begin
    Close;
end;

function TFrmCenAlterar.LocalizarAno(anoAT, indGrad: Integer): Boolean;
var a, ARow: Integer;
    valido, achou: Boolean;
begin
    valido := true;
    a := anoAT - priAno;
    if ((anoAT < priAno) or (anoAT > ultAno)) then
        valido := false
    else if Not (anoAT = detAno[a,0]) then
        begin
            a := 0;
            achou := false;
            repeat
                if (anoAT = detAno[a,0]) then
                    achou := true
                else
                    a := a + 1;
            until (achou or (a > High (detAno)));
            valido := achou;
        end
    else if ((indGrad = 1) and (detAno[a,1] = 0)) then
        valido := false;
    if valido then
        case indGrad of
            0: with StgCustoOpt do
                begin
                    ARow := RowCount - (linAno[a,0] + 1);
                    if (ARow >= 20) then
                        ARow := 20;
                    Col := 0;
                    Row := linAno[a,0] + ARow;
                    Row := Row - ARow;
                end;
            1: with StgCustoMaq do
                begin
                    ARow := RowCount - (linAno[a,1] + 1);
                    if (ARow >= 20) then
                        ARow := 20;
                    Col := 0;
                    Row := linAno[a,1] + ARow;
                    Row := Row - ARow;
                end;
        end;
    LocalizarAno := valido;
end;

```

end;

```
procedure TFrmCenAlterar.OpLocalizarAno1Click(Sender: TObject);
var strAno: String;
begin
  try
    strAno := '';
    if InputQuery ('Série temporal - localizar ano', 'Ano (aaaa)', strAno) then
      if Not (LocalizarAno (StrToInt (strAno), 0)) then
        MessageDlg ('O ano especificado não existe.', MTInformation, [MbOK], 0);
  except
    on EConvertError do
      raise Exception.Create ('O valor fornecido é inválido');
  end;
end;
```

```
procedure TFrmCenAlterar.OpLocalizarAno2Click(Sender: TObject);
var strAno: String;
begin
  try
    strAno := '';
    if InputQuery ('Série temporal - localizar ano', 'Ano (aaaa)', strAno) then
      if Not (LocalizarAno (StrToInt (strAno), 1)) then
        MessageDlg ('O ano especificado não existe.', MTInformation, [MbOK], 0);
  except
    on EConvertError do
      raise Exception.Create ('O valor fornecido é inválido');
  end;
end;
```

```
procedure TFrmCenAlterar.ChkListBoxCenColMouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var APoint: TPoint;
begin
  APoint.X := X;
  APoint.Y := Y;
  indCol := ChkListBoxCenCol.ItemAtPos (APoint, true);
end;
```

```
procedure TFrmCenAlterar.ChkListBoxCenColKeyDown(Sender: TObject;
  var Key: Word; Shift: TShiftState);
begin
  indCol := ChkListBoxCenCol.ItemIndex;
  if (indCol < 0) then
    indCol := 0;
end;
```

```
procedure TFrmCenAlterar.DistribuirValores(indVar: Integer; valor: array of Real);
var i, ind, freq: Integer;
    vAT: Real;
    ordenou: Boolean;
begin
  ind := 0;
  repeat
    ind := ind + 1;
    ordenou := true;
```

```

for i := High (valor) downto ind do
  if (valor[i] > valor[i - 1]) then
    begin
      vAT := valor[i - 1];
      valor[i - 1] := valor[i];
      valor[i] := vAT;
      ordenou := false;
    end;
until ordenou or (ind = High (valor));
ind := -1;
for i := 0 to High (valor) do
  if ((i > 0) and (valor[i] = valor[i - 1])) then
    categoria[indVar,ind,1] := categoria[indVar,ind,1] + 1
  else
    begin
      ind := ind + 1;
      SetLength (categoria[indVar], ind + 1);
      categoria[indVar,ind,0] := valor[i];
      categoria[indVar,ind,1] := 1;
    end;
freq := 0;
for ind := High (categoria[indVar]) downto 0 do
  begin
    freq := freq + Trunc (categoria[indVar,ind,1]);
    categoria[indVar,ind,2] := freq;
    categoria[indVar,ind,3] := (freq / (High (valor) + 1)) * 100;
  end;
end;

```

```

procedure TFrmCenAlterar.CalcularMedias(opResult, indVar: Integer; valor: array of Real);
var i: Integer;
    min, max, media, desvio: Real;
begin
  min := valor[0];
  max := valor[0];
  media := 0;
  for i := 0 to High (valor) do
    begin
      if (min > valor[i]) then
        min := valor[i]
      else if (max < valor[i]) then
        max := valor[i];
      media := media + valor[i];
    end;
  media := media / (High (valor) + 1);
  desvio := 0;
  for i := 0 to High (valor) do
    desvio := desvio + Sqr (valor[i] - media);
  desvio := Sqrt (desvio / High (valor));
  case opResult of
    0: begin
        resultFinalVar[indVar,0] := min;
        resultFinalVar[indVar,1] := max;
        resultFinalVar[indVar,2] := media;
        resultFinalVar[indVar,3] := desvio;
      end;
  end;

```

```

1, 2: begin
    resultFinalTal[indVar,opResult - 1,0] := min;
    resultFinalTal[indVar,opResult - 1,1] := max;
    resultFinalTal[indVar,opResult - 1,2] := media;
    resultFinalTal[indVar,opResult - 1,3] := desvio;
end;
end;
end;

procedure TFrmCenAlterar.OpExportarClick(Sender: TObject);
var t, ind, i: Integer;
    achou: Boolean;
begin
    if Not (StgResultFinalEst.RowCount > 2) then
        raise Exception.Create ('Não há nenhuma informação para exportar');
    try
        FrmExpExcel := TFrmExpExcel.Create (Application);
        with FrmExpExcel do
            begin
                SetLength (indCultivar, 1);
                SetLength (indColhedora, numCol);
                opSimul := 1;
                codSimul := codCen;
                areaTotalSimul := areaTotalCen;
                stgVirtual := StgResultFinalEst;
                ind := 0;
                for t := 1 to (StgCenTal.RowCount - 1) do
                    if (t > 1) then
                        begin
                            achou := false;
                            i := 0;
                            repeat
                                if (talCultivar[t,1] = indCultivar[i,0]) then
                                    achou := true
                                else
                                    i := i + 1;
                            until (achou or (i > ind));
                            if achou then
                                begin
                                    indCultivar[i,1] := indCultivar[i,1] + StrToFloat (StgCenTal.Cells[3,t]);
                                    indCultivar[i,2] := indCultivar[i,2] + 1;
                                end
                            else
                                begin
                                    ind := ind + 1;
                                    SetLength (indCultivar, ind + 1);
                                    indCultivar[ind,0] := talCultivar[t,1];
                                    indCultivar[ind,1] := StrToFloat (StgCenTal.Cells[3,t]);
                                    indCultivar[ind,2] := 1;
                                end;
                            end
                        end
                    else
                        begin
                            indCultivar[0,0] := talCultivar[1,1];
                            indCultivar[0,1] := StrToFloat (StgCenTal.Cells[3,1]);
                            indCultivar[0,2] := 1;
                        end
                    end
                end
            end
        end
    end
end;

```

```

        end;
    for i := 0 to High (indColhedora) do
    begin
        indColhedora[i,0] := codColhedora[FrmCenAlterar.indColhedora[i],3];
        indColhedora[i,1] := codColhedora[FrmCenAlterar.indColhedora[i],2];
    end;
    ShowModal;
end;
finally
    FrmExpExcel.Free;
end;
end;

end.

```

F - Módulo 9: Módulo Geração de Propriedades (Arquivos UPropriedadeNovo.pas, UPropriedadeAlterar.pas)

G - Código Fonte 16: UPropriedadeNovo.pas <<código>>

```

unit UPropriedadeNovo;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    Menus, StdCtrls, ComCtrls, ToolWin, Buttons, ExtCtrls, DBCtrls, CheckLst,
    Grids, Mask, DB;

type
    TFrmPropNovo = class(TForm)
        Lbl1: TLabel;
        MainMenuPropNovo: TMainMenu;
        MenuPropriedade: TMenuItem;
        MenuEditar: TMenuItem;
        OpNovo: TMenuItem;
        OpSalvar: TMenuItem;
        OpIndicador: TMenuItem;
        ToolBarPropNovo: TToolBar;
        ToolBtnNovo: TToolButton;
        ToolBtnSalvar: TToolButton;
        Pnl1: TPanel;
        Pnl3: TPanel;
        Pnl2: TPanel;
        BtnConfirmar: TBitBtn;
        BtnCancelar: TBitBtn;
        LblCod: TLabel;
        PgeCtrlPropNovo: TPageControl;
        TabShtProp: TTabSheet;
        TabShtCustoOpt: TTabSheet;
        TabShtCustoMaq: TTabSheet;
        StatusBarPropNovo: TStatusBar;
        Pnl4: TPanel;
        GrpBoxPropCol: TGroupBox;
    end;

```

```

Pnl5: TPanel;
GrpBoxProp: TGroupBox;
GrpBoxPer: TGroupBox;
Pnl6: TPanel;
Lbl2: TLabel;
Lbl3: TLabel;
Lbl4: TLabel;
Lbl5: TLabel;
Lbl6: TLabel;
DBLookupCboReg: TDBLookupComboBox;
TxtTurno: TEdit;
TxtPrec: TEdit;
CboPriAno: TComboBox;
CboUltAno: TComboBox;
Pnl7: TPanel;
ChkListBoxPropCol: TCheckListBox;
GrpBoxPropTal: TGroupBox;
Pnl8: TPanel;
StgPropTal: TStringGrid;
PopupMenuPropTal: TPopupMenu;
OpIncluirTal: TMenuItem;
OpExcluirTal: TMenuItem;
MenuN1: TMenuItem;
OpLimparListaTal: TMenuItem;
DBLookupCboCul: TDBLookupComboBox;
TxtCell: TMaskEdit;
StgCustoOpt: TStringGrid;
StgCustoMaq: TStringGrid;
ToolBarCustoOpt: TToolBar;
ToolBtnLocalizarAno1: TToolButton;
ToolBarCustoMaq: TToolBar;
ToolBtnLocalizarAno2: TToolButton;
PopupMenuGrad1: TPopupMenu;
OpLocalizarAno1: TMenuItem;
PopupMenuGrad2: TPopupMenu;
OpLocalizarAno2: TMenuItem;
ToolBarPropTal: TToolBar;
ToolBtnIncluirTal: TToolButton;
ToolBtnExcluirTal: TToolButton;
ToolN1: TToolButton;
ToolBtnLimparListaTal: TToolButton;
TabShtResultFinal: TTabSheet;
StgResultFinalEst: TStringGrid;
PopupMenuGradResultFinal: TPopupMenu;
OpExportar: TMenuItem;
ToolN2: TToolButton;
ToolBtnIndicador: TToolButton;
procedure ChkListBoxPropColClickCheck(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure OpIncluirTalClick(Sender: TObject);
procedure OpExcluirTalClick(Sender: TObject);
procedure OpLimparListaTalClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure DBLookupCboCulClick(Sender: TObject);
procedure DBLookupCboCulExit(Sender: TObject);
procedure StgPropTalSelectCell(Sender: TObject; ACol, ARow: Integer);

```

```

    var CanSelect: Boolean);
procedure DBLookupCboRegClick(Sender: TObject);
procedure BtnConfirmarClick(Sender: TObject);
procedure StgPropTalGetEditText(Sender: TObject; ACol, ARow: Integer;
    var Value: String);
procedure TxtCellExit(Sender: TObject);
procedure DBLookupCboCulKeyPress(Sender: TObject; var Key: Char);
procedure TxtCellKeyPress(Sender: TObject; var Key: Char);
procedure OpNovoClick(Sender: TObject);
procedure OpSalvarClick(Sender: TObject);
procedure OpIndicadorClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure BtnCancelarClick(Sender: TObject);
procedure OpLocalizarAno1Click(Sender: TObject);
procedure OpLocalizarAno2Click(Sender: TObject);
procedure ChkListBoxPropColMouseDown(Sender: TObject;
    Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
procedure ChkListBoxPropColKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
procedure OpExportarClick(Sender: TObject);
private
{ Private declarations }
priAno, ultAno, numCol, indCol: Integer;
turnoTrab, precCrit, areaTotalProp: Real;
codColhedora: array of array[0..5] of Real; {tornar-se-á 'public'}
modColhedora: array of array[0..1] of String;
indColhedora: array of Integer;
talCultivar: array[1..26,0..5] of Real;
linAno: array of array[0..1] of Integer;
detAno: array of array[0..6] of Real;
colAno: array of array of Real;
resultPropAno: array of array of array[0..6] of Real;
resultMaqAno: array of array of array of array[0..2] of Real;
resultFinalVar: array[0..3] of array[0..3] of Real;
resultFinalTal: array of array[0..1] of array[0..3] of Real;
//categoria: array[0..n] of array of array[0..2] of Real;
valCell, codProp: String;
regValido, propGerado, propSalvado: Boolean;
public
{ Public declarations }
function PesquisarPrec (codReg: Integer): Boolean;
function PesquisarFuncPerdas (codReg, codCul: Integer): Boolean;
function GerarCodigo: Boolean;
function SalvarPropriedade: Boolean;
function LocalizarAno (anoAT, indGrad: Integer): Boolean;
function CaixaDeEntrada (cs: Integer; var str1, str2: String): Boolean;
procedure ListarColhedoras (Sender: TObject);
procedure NovoPropriedade (Sender: TObject);
procedure IncluirTal (Sender: TObject);
procedure ExcluirTal (Sender: TObject);
procedure LimparListaTal (Sender: TObject);
procedure FuncPerdas (Sender: TObject);
procedure ServMecanizado (Sender: TObject);
procedure ResultadoFinal (Sender: TObject);
//procedure DistribuirValores (indVar: Integer; valor: array of Real);
procedure CalcularMedias (opResult, indVar: Integer; valor: array of Real);

```

```

end;

var
  FrmPropNovo: TFrmPropNovo;

implementation

uses USimulacao, UCaixaDeEntrada, UIndicador, URegiao, UMaquina, UCultivar, UExpExcel;

{$R *.DFM}

procedure TFrmPropNovo.ListarColhedoras(Sender: TObject);
var cs: Integer;
    strModelo: String;
begin
  ChkListBoxPropCol.Items.Clear;
  SetLength (codColhedora, FrmMaq.TblCol.RecordCount);
  SetLength (modColhedora, FrmMaq.TblCol.RecordCount);
  FrmMaq.TblCol.First;
  cs := 0;
  repeat
    with FrmMaq.TblCol do
      begin
        codColhedora[cs,0] := FieldValues['CódColhedora'];
        codColhedora[cs,1] := 0;
        codColhedora[cs,2] := 0;
        codColhedora[cs,3] := FieldValues['Potência'];
        codColhedora[cs,4] := FieldValues['LarguraDePlataforma'];
        codColhedora[cs,5] := FieldValues['PreçoDeAquisição'];
        modColhedora[cs,0] := FieldValues['Marca'];
        modColhedora[cs,1] := FieldValues['Modelo'];
      end;
      strModelo := modColhedora[cs,1];
      strModelo := strModelo + ' (plataforma de ' ;
      strModelo := strModelo + FloatToStr (codColhedora[cs,4]);
      strModelo := strModelo + ' metros)';
      ChkListBoxPropCol.Items.Add (strModelo);
      cs := cs + 1;
      FrmMaq.TblCol.Next;
    until FrmMaq.TblCol.Eof;
    numCol := 0;
  end;

function TFrmPropNovo.PesquisarFuncPerdas(codReg, codCul: Integer): Boolean;
var ACol, t: Integer;
    msg: String;
begin
  PesquisarFuncPerdas := true;
  with FrmCultivar.QryRegCul do
    begin
      Close;
      SQL.Clear;
      SQL.Add ('select *');
      SQL.Add ('from "REGIÃO_CULTIVAR"');
      SQL.Add ('where CódRegião = ' + IntToStr (codReg));
      SQL.Add ('and CódCultivar = ' + IntToStr (codCul));
    end;
  end;
end;

```

```

    Open;
    FindFirst;
end;
if Not (FrmCultivar.QryRegCul.Found) then
begin
    PesquisarFuncPerdas := false;
    for ACol := 4 to 8 do
        StgPropTal.Cells[ACol,StgPropTal.Row] := 'Não encontrado';
        MessageDlg ('Não há nenhuma função de perdas que corresponda às descrições fornecidas.',
            MTInformation, [MbOK], 0);
    end
else
    with StgPropTal do
    begin
        Cells[4,Row] := FrmCultivar.QryRegCul.FieldValues['CicloDeMaturação'];
        if (Cells[5,Row] <> "") then
            if (Cells[5,Row] = 'Não encontrado') then
                for ACol := 5 to 8 do
                    Cells[ACol,Row] := "
                else
                begin
                    Cells[6,Row] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[5,Row]) + StrToInt
(Cells[4,Row]));
                    Cells[7,Row] := Cells[6,Row];
                    Cells[8,Row] := '0';
                    t := -1;
                    if ((Row > 1) and (Cells[7,Row - 1] <> "") and (StrToDate (Cells[7,Row]) <= StrToDate
(Cells[7,Row - 1]))) then
                        begin
                            t := Row - 1;
                            msg := 'antecede';
                        end
                    else if ((Row < (RowCount - 1)) and (Cells[7,Row + 1] <> "") and (StrToDate (Cells[7,Row]) >=
StrToDate (Cells[7,Row + 1]))) then
                        begin
                            t := Row + 1;
                            msg := 'sucede';
                        end;
                    if (t <> -1) then
                        MessageDlg ('O início previsto para a colheita do talhão ' + letra[Row] + ' coincide ou ' + msg +
' o início do talhão ' + letra[t] + '.',
                            MTWarning, [MbOK], 0);
                    end;
                end;
            end;
        end;
    end;
end;

function TFrmPropNovo.PesquisarPrec(codReg: Integer): Boolean;
begin
    PesquisarPrec := true;
    CboPriAno.Clear;
    CboPriAno.Items.Clear;
    CboUltAno.Clear;
    CboUltAno.Items.Clear;
    with FrmSimul.QryPrec do
    begin
        Close;

```

```

SQL.Clear;
SQL.Add ('select distinct extract (year from Data) as Ano');
SQL.Add ('from "REGIÃO_Precipitação"');
SQL.Add ('where CódRegião = ' + IntToStr (codReg));
Open;
FindFirst;
end;
if Not (FrmSimul.QryPrec.Found) then
begin
    PesquisarPrec := false;
    MessageDlg ('Não há nenhum ano cadastrado na série temporal da região.',
        MTError, [MbOK], 0);
end
else
begin
    repeat
        CboPriAno.Items.Add (IntToStr (FrmSimul.QryPrec.FieldValues['Ano']));
        FrmSimul.QryPrec.Next;
    until FrmSimul.QryPrec.Eof;
    CboPriAno.ItemIndex := 0;
    CboUltAno.Items := CboPriAno.Items;
    CboUltAno.ItemIndex := 0;
    if (FrmSimul.QryPrec.RecordCount < 2) then
    begin
        PesquisarPrec := false;
        MessageDlg ('Deve haver no mínimo dois anos cadastrados na série temporal da região.',
            MTWarning, [MbOK], 0);
    end;
end;
end;
end;

procedure TFrmPropNovo.ChkListBoxPropColClickCheck(Sender: TObject);
var n: Integer;
    v: Real;
    strVeloc, strNumCol: String;
    valido, continua: Boolean;
begin
    try
        if (indCol = ChkListBoxPropCol.ItemIndex) then
            if ChkListBoxPropCol.Checked[indCol] then
                begin
                    numCol := numCol + 1;
                    strVeloc := FloatToStr (codColhedora[indCol,1]);
                    strNumCol := FloatToStr (codColhedora[indCol,2]);
                    valido := false;
                    repeat
                        continua := CaixaDeEntrada (indCol, strVeloc, strNumCol);
                        if continua then
                            begin
                                v := StrToFloat (strVeloc);
                                n := StrToInt (strNumCol);
                                if ((v < 3) or (v > 7)) then
                                    MessageDlg ('A velocidade de operação é inválida.', MTError, [MbOK], 0)
                                else if (n < 1) then
                                    MessageDlg ('O número de colhedoras é inválido.', MTError, [MbOK], 0)
                                else

```

```

begin
    valido := true;
    codColhedora[indCol,1] := v;
    codColhedora[indCol,2] := n;
end;
end;
until (valido or Not (continua));
if Not (continua) then
begin
    ChkListBoxPropCol.Checked[indCol] := false;
    numCol := numCol - 1;
end;
end
else
    numCol := numCol - 1
else
    ChkListBoxPropCol.Checked[indCol] := Not (ChkListBoxPropCol.Checked[indCol]);
except
on EConvertError do
begin
    MessageDlg ('O valor fornecido é inválido.', MTErrors, [MbOK], 0);
    indCol := ChkListBoxPropCol.ItemIndex;
    if ((codColhedora[indCol,1] = 0) or (codColhedora[indCol,2] = 0)) then
begin
    ChkListBoxPropCol.Checked[indCol] := false;
    numCol := numCol - 1;
end;
end;
end;
end;
end;

procedure TFrmPropNovo.FormShow(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    try
        NovoPropriedade (self);
    finally
        Screen.Cursor := crDefault;
    end;
end;

procedure TFrmPropNovo.IncluirTal(Sender: TObject);
begin
    with StgPropTal do
begin
    RowCount := RowCount + 1;
    Rows[RowCount - 1].Clear;
    Cells[0,RowCount - 1] := letra[RowCount - 1];
    talCultivar[RowCount - 1,0] := 0;
    talCultivar[RowCount - 1,1] := 0;
end;
end;

procedure TFrmPropNovo.ExcluirTal(Sender: TObject);
var ARow: Integer;
begin

```

```

with StgPropTal do
begin
  if (Row < (RowCount - 1)) then
    for ARow := Row to (RowCount - 2) do
      begin
        Rows[ARow] := Rows[ARow + 1];
        Cells[0,ARow] := letra[ARow];
        talCultivar[ARow] := talCultivar[ARow + 1];
      end;
    if (RowCount > 2) then
      RowCount := RowCount - 1
    else
      begin
        Rows[1].Clear;
        Cells[0,1] := letra[1];
        talCultivar[1,0] := 0;
        talCultivar[1,1] := 0;
      end;
    end;
end;

procedure TFrmPropNovo.LimparListaTal(Sender: TObject);
var ARow: Integer;
begin
  with StgPropTal do
    begin
      for ARow := (RowCount - 1) downto 1 do
        Rows[ARow].Clear;
        RowCount := 2;
        Cells[0,1] := letra[1];
        talCultivar[1,0] := 0;
        talCultivar[1,1] := 0;
      end;
    end;
end;

procedure TFrmPropNovo.OpIncluirTalClick(Sender: TObject);
begin
  if (Not (propGerado) and (StgPropTal.RowCount < 27) and (talCultivar[StgPropTal.RowCount - 1,1]
<> 0)) then
    IncluirTal (self);
end;

procedure TFrmPropNovo.OpExcluirTalClick(Sender: TObject);
begin
  if Not (propGerado) then
    ExcluirTal (self);
end;

procedure TFrmPropNovo.OpLimparListaTalClick(Sender: TObject);
begin
  if Not (propGerado) then
    LimparListaTal (self);
end;

procedure TFrmPropNovo.FormCreate(Sender: TObject);
begin

```

```

with StgPropTal do
begin
  Cells[0,0] := 'Talhão';ColWidths[0] := 40;
  Cells[1,0] := 'Cultivar';ColWidths[1] := 127;
  Cells[2,0] := 'Grupo de maturação';ColWidths[2] := 104;
  Cells[3,0] := 'Área de lavoura (ha)';ColWidths[3] := 103;
  Cells[4,0] := 'Ciclo de maturação (dias)';ColWidths[4] := 126;
  Cells[5,0] := 'Data de semeadura';ColWidths[5] := 100;
  Cells[6,0] := 'Data provável de colheita';ColWidths[6] := 129;
  Cells[7,0] := 'Início previsto da colheita';ColWidths[7] := 129;
  Cells[8,0] := 'Atraso inicial (dias)';ColWidths[8] := 94;
  DefaultRowHeight := DBLookupCboCul.Height;
end;
with StgCustoOpt do
begin
  Cells[0,0] := 'Data (dd/mm/aa)';ColWidths[0] := 148;
  Cells[1,0] := 'Precipitação (mm)';ColWidths[1] := 120;
  Cells[2,0] := 'Situação';ColWidths[2] := 49;
  Cells[3,0] := 'Atraso (dias)';ColWidths[3] := 65;
  Cells[4,0] := 'Perda física diária (sc)';ColWidths[4] := 112;
  Cells[5,0] := 'Perda física acumulada (sc)';ColWidths[5] := 139;
  Cells[6,0] := 'Renda física diária (sc)';ColWidths[6] := 116;
  Cells[7,0] := 'Renda física acumulada (sc)';ColWidths[7] := 143;
  Cells[8,0] := 'Área colhida (ha)';ColWidths[8] := 87;
  DefaultRowHeight := StgPropTal.DefaultRowHeight;
end;
with StgCustoMaq do
begin
  Cells[0,0] := 'Ano/Colhedora/Talhão';ColWidths[0] := 177;
  Cells[1,0] := 'Área colhida (ha)';ColWidths[1] := 87;
  Cells[2,0] := 'Perda total (US$)';ColWidths[2] := 110;
  Cells[3,0] := 'Renda total (US$)';ColWidths[3] := 92;
  Cells[4,0] := 'Custo de serviço mecanizado (US$)';ColWidths[4] := 176;
  Cells[5,0] := 'Custo unitário (US$/sc)';ColWidths[5] := 117;
  DefaultRowHeight := StgPropTal.DefaultRowHeight;
end;
with StgResultFinalEst do
begin
  ColWidths[0] := 136;
  ColWidths[4] := 76;
  DefaultRowHeight := StgPropTal.DefaultRowHeight;
end;
TxtCell.BorderStyle := bsNone;
end;

procedure TFrmPropNovo.NovoPropriedade(Sender: TObject);
var ARow: Integer;
begin
  propGerado := false;
  propSalvado := false;
  PrecSaco := 12;
  Juros := 12;
  VidaUtil := 10;
  Eficiencia := 75;
  Salario := 150;
  SegAloj := 3.2;

```

```

Diesel := 0.45;
Residual := 20;
with FrmReg.TblReg do
begin
    First;
    Edit;
    DBLookupCboReg.KeyValue := FieldValues['CódRegião'];
end;
LblCod.Caption := '';
TxtTurno.Clear;
TxtPrec.Clear;
PgCtrlPropNovo.ActivePageIndex := 0;
ListarColhedoras (self);
LimparListaTal (self);
for ARow := (StgCustoOpt.RowCount - 1) downto 1 do
    StgCustoOpt.Rows[ARow].Clear;
StgCustoOpt.RowCount := 2;
for ARow := (StgCustoMaq.RowCount - 1) downto 1 do
    StgCustoMaq.Rows[ARow].Clear;
StgCustoMaq.RowCount := 2;
for ARow := (StgResultFinalEst.RowCount - 1) downto 0 do
    StgResultFinalEst.Rows[ARow].Clear;
StgResultFinalEst.ColCount := 5;
StgResultFinalEst.RowCount := 2;
regValido := PesquisarPrec (DBLookupCboReg.KeyValue);
end;

procedure TFrmPropNovo.DBLookupCboCulClick(Sender: TObject);
begin
    StgPropTal.Cells[2,StgPropTal.Row] := FrmCultivar.TblCul.FieldValues['GrupoDeMaturação'];
    talCultivar[StgPropTal.Row,0] := 1;
    if PesquisarFuncPerdas (DBLookupCboReg.KeyValue, DBLookupCboCul.KeyValue) then
        with FrmCultivar.QryRegCul do
            begin
                talCultivar[StgPropTal.Row,1] := DBLookupCboCul.KeyValue;
                talCultivar[StgPropTal.Row,2] := FieldValues['CoeficienteDePerdA'];
                talCultivar[StgPropTal.Row,3] := FieldValues['CoeficienteDePerdB'];
                talCultivar[StgPropTal.Row,4] := FieldValues['CoeficienteDePerdC'];
                talCultivar[StgPropTal.Row,5] := FieldValues['CoeficienteDeProdC'];
            end
        else
            talCultivar[StgPropTal.Row,1] := 0;
    end;

procedure TFrmPropNovo.DBLookupCboCulExit(Sender: TObject);
var resp: Word;
    continua: Boolean;
begin
    continua := false;
    if (talCultivar[StgPropTal.Row,0] = 1) then
        begin
            if (talCultivar[StgPropTal.Row,1] <> 0) then
                StgPropTal.Cells[1,StgPropTal.Row] := DBLookupCboCul.Text
            else
                begin

```

```

    resp := MessageDlg ('O cultivar é inválido. Pressione OK para selecionar outro ou Cancel para
excluir o talhão.',

```

```

    MTErrror, [MbOK,MbCancel], 0);
    if (resp = mrOK) then
    begin
        continua := true;
        DBLookupCboCul.SetFocus;
    end
    else
        ExcluirTal (self);
    end;
end;
if Not (continua) then
begin
    DBLookupCboCul.Visible := false;
    StgPropTal.SetFocus;
end;
end;
end;

```

```

procedure TFrmPropNovo.StgPropTalSelectCell(Sender: TObject; ACol,
    ARow: Integer; var CanSelect: Boolean);
var cod: Real;
    R: TRect;
begin
    CanSelect := false;
    if Not (regValido) then
        raise Exception.Create ('A região é inválida. Selecione outra para poder continuar');
    if (ACol = 1) then
    begin
        R := StgPropTal.CellRect (ACol,ARow);
        R.Left := R.Left + StgPropTal.Left;
        R.Right := R.Right + StgPropTal.Left;
        R.Top := R.Top + StgPropTal.Top;
        R.Bottom := R.Bottom + StgPropTal.Top;
        with DBLookupCboCul do
        begin
            Left := R.Left + 1;
            Top := R.Top + 1;
            Width := (R.Right + 1) - R.Left;
            Height := (R.Bottom + 1) - R.Top;
            if (talCultivar[ARow,1] <> 0) then
                cod := talCultivar[ARow,1]
            else
                cod := 1;
            FrmCultivar.TblCul.Edit;
            KeyValue := cod;
            Visible := true;
            SetFocus;
        end;
    end;
    CanSelect := true;
end;

```

```

procedure TFrmPropNovo.DBLookupCboRegClick(Sender: TObject);
begin
    Screen.Cursor := crHourglass;

```

```

try
  regValido := PesquisarPrec (DBLookupCboReg.KeyValue);
finally
  Screen.Cursor := crDefault;
end;
end;

procedure TFrmPropNovo.BtnConfirmarClick(Sender: TObject);
var t, anoAT, c, cs: Integer;
begin
  Screen.Cursor := crHourglass;
  try try
    if propGerado then
      raise Exception.Create ('A propriedade já foi gerada')
    else if Not (regValido) then
      raise Exception.Create ('A região é inválida. Selecione outra para poder continuar')
    else if (TxtTurno.Text = '') then
      raise Exception.Create ('Preencha o campo "Turno de trabalho".')
    else if (TxtPrec.Text = '') then
      raise Exception.Create ('Preencha o campo "Precipitação crítica".');
    turnoTrab := StrToFloat (TxtTurno.Text);
    precCrit := StrToFloat (TxtPrec.Text);
    priAno := StrToInt (CboPriAno.Items[CboPriAno.ItemIndex]);
    ultAno := StrToInt (CboUltAno.Items[CboUltAno.ItemIndex]);
    if (turnoTrab > FrmReg.TblReg.FieldValues['TurnoDeTrabalho']) then
      raise Exception.Create ('O turno de trabalho é inválido')
    else if ((ultAno - priAno) < 1) then
      raise Exception.Create ('O período de simulação é inválido')
    else if (numCol < 1) then
      raise Exception.Create ('Deve haver no mínimo uma colhedora selecionada');
    areaTotalProp := 0;
    for t := 1 to (StgPropTal.RowCount - 1) do
      begin
        if Not (talCultivar[t,1] <> 0) then
          raise Exception.Create ('Nenhum cultivar foi atribuído ao talhão ' + letra[t])
        else if (StgPropTal.Cells[3,t] = '') then
          raise Exception.Create ('Forneça a área de lavoura do talhão ' + letra[t])
        else if (StgPropTal.Cells[5,t] = '') then
          raise Exception.Create ('Forneça a data de semeadura do talhão ' + letra[t])
        else if ((t > 1) and (StrToDate (StgPropTal.Cells[7,t - 1]) >= StrToDate (StgPropTal.Cells[7,t]))) then
          raise Exception.Create ('O início previsto para a colheita do talhão ' + letra[t - 1] + ' coincide ou
sucedo o início do talhão ' + letra[t] + '.');
        areaTotalProp := areaTotalProp + StrToFloat (StgPropTal.Cells[3,t]);
      end;
    SetLength (indColhedora, numCol);
    SetLength (linAno, (ultAno + 1) - priAno);
    SetLength (detAno, (ultAno + 1) - priAno);
    SetLength (colAno, (ultAno + 1) - priAno);
    SetLength (resultPropAno, StgPropTal.RowCount - 1);
    SetLength (resultMaqAno, StgPropTal.RowCount - 1);
    SetLength (resultFinalTal, StgPropTal.RowCount - 1);
    c := 0;
    for cs := 0 to (ChkListBoxPropCol.Items.Count - 1) do
      if ChkListBoxPropCol.Checked[cs] then
        begin
          indColhedora[c] := cs;

```

```

        c := c + 1;
    end;
    for t := 1 to (StgPropTal.RowCount - 1) do
    begin
        SetLength (resultPropAno[t - 1], (ultAno + 1) - priAno);
        SetLength (resultMaqAno[t - 1], (ultAno + 1) - priAno);
        for anoAT := priAno to ultAno do
        begin
            SetLength (resultMaqAno[t - 1,(anoAT - priAno)], numCol);
            if (t = 1) then
            begin
                SetLength (colAno[anoAT - priAno], numCol);
                detAno[anoAT - priAno,0] := 1;
                detAno[anoAT - priAno,1] := 0;
                detAno[anoAT - priAno,2] := 0;
                detAno[anoAT - priAno,3] := 0;
                detAno[anoAT - priAno,4] := 0;
                detAno[anoAT - priAno,5] := 0;
                detAno[anoAT - priAno,6] := 0;
                for c := 0 to (numCol - 1) do
                    colAno[anoAT - priAno,c] := 0;
                end;
            end;
        end;
    end;
    FuncPerdas (self);
    ServMecanizado (self);
    ResultadoFinal (self);
    propGerado := GerarCodigo;
finally
    Screen.Cursor := crDefault;
end;
except
    on EConvertError do
        raise Exception.Create ('O valor do campo é inválido');
    end;
end;

procedure TFrmPropNovo.StgPropTalGetEditText(Sender: TObject; ACol,
    ARow: Integer; var Value: String);
var R: TRect;
begin
    valCell := StgPropTal.Cells[ACol,ARow];
    R := StgPropTal.CellRect (ACol,ARow);
    R.Left := R.Left + StgPropTal.Left;
    R.Right := R.Right + StgPropTal.Left;
    R.Top := R.Top + StgPropTal.Top;
    R.Bottom := R.Bottom + StgPropTal.Top;
    with TxtCell do
    begin
        case ACol of
            0, 1, 2, 3, 4, 8: EditMask := '';
            5, 6, 7: EditMask := '!99/99/99;1;_';
        end;
        Left := R.Left + 2;
        Top := R.Top + 2;
        Width := R.Right - R.Left;
    end;
end;

```

```

Height := R.Bottom - R.Top;
Text := valCell;
Visible := true;
SetFocus;
end;
end;

procedure TFrmPropNovo.TxtCellExit(Sender: TObject);
var t, atraso: Integer;
    msg: String;
    valido: Boolean;
    data: TDateTime;
begin
    try try
        case StgPropTal.Col of
            0, 1, 2, 4, 6, 8: begin
                StgPropTal.Cells[StgPropTal.Col,StgPropTal.Row] := valCell;
                if (valCell <> TxtCell.Text) then
                    MessageDlg ('O valor não pode ser alterado porque o campo "' +
StgPropTal.Cells[StgPropTal.Col,0] + '" está aberto somente para leitura.',
                        MTErr, [MbOK], 0);
                end;
            3: StgPropTal.Cells[3,StgPropTal.Row] := FloatToStr (StrToFloat (TxtCell.Text));
            5: with StgPropTal do
                begin
                    valido := true;
                    data := StrToDate (TxtCell.Text);
                    if (data < StrToDate (FormatDateTime ('dd/mm/' + Copy (IntToStr (StrToInt (FormatDateTime
('yyyy', Date)) - 1), 3, 2), FrmReg.TblReg.FieldValues['PriDataDeSemeadura'])))
                    or (data > StrToDate (FormatDateTime ('dd/mm/' + Copy (IntToStr (StrToInt (FormatDateTime
('yyyy', Date)) - 1), 3, 2), FrmReg.TblReg.FieldValues['UltDataDeSemeadura']))) then
                        valido := false
                    else
                        for t := 1 to (RowCount - 1) do
                            if ((t <> Row) and (talCultivar[t,1] = talCultivar[Row,1]) and (Cells[5,t] = TxtCell.Text)) then
                                valido := false;
                        if valido then
                            begin
                                Cells[5,Row] := TxtCell.Text;
                                Cells[6,Row] := FormatDateTime ('dd/mm/yy', data + StrToInt (Cells[4,Row]));
                                Cells[7,Row] := Cells[6,Row];
                                Cells[8,Row] := '0';
                                t := -1;
                                if ((Row > 1) and (Cells[7,Row - 1] <> '') and (StrToDate (Cells[7,Row]) <= StrToDate
(Cells[7,Row - 1]))) then
                                    begin
                                        t := Row - 1;
                                        msg := 'antecede';
                                    end
                                else if ((Row < (RowCount - 1)) and (Cells[7,Row + 1] <> '') and (StrToDate (Cells[7,Row]) >=
StrToDate (Cells[7,Row + 1]))) then
                                    begin
                                        t := Row + 1;
                                        msg := 'sucede';
                                    end;
                                if (t <> - 1) then

```

```

        MessageDlg ('O início previsto para a colheita do talhão ' + letra[Row] + ' coincide ou ' + msg
+ ' o início do talhão ' + letra[t] + '.');
        MTWarning, [MbOK], 0);
    end
    else
    begin
        Cells[5,Row] := valCell;
        MessageDlg ('A data de semeadura é inválida.', MError, [MbOK], 0);
    end;
end;
7: with StgPropTal do
begin
    data := StrToDate (TxtCell.Text);
    if ((data < (StrToDate (Cells[6,Row]) - 3)) or (data > (StrToDate (Cells[6,Row]) + 27))) then
    begin
        Cells[7,Row] := valCell;
        MessageDlg ('A data de início previsto da colheita é inválida.', MError, [MbOK], 0);
    end
    else
    begin
        atraso := 0;
        if (data > (StrToDate (Cells[6,Row]) + 3)) then
        begin
            atraso := Trunc (data - (StrToDate (Cells[6,Row]) + 3));
            MessageDlg ('A data ' + TxtCell.Text + ' excede em ' + FloatToStr (atraso) + ' dia(s) o limite
máximo para o início previsto da colheita.',
                MTWarning, [MbOK], 0);
        end;
        Cells[7,Row] := TxtCell.Text;
        Cells[8,Row] := FloatToStr (atraso);
        t := -1;
        if ((Row > 1) and (Cells[7,Row - 1] <> '') and (data <= StrToDate (Cells[7,Row - 1]))) then
        begin
            t := Row - 1;
            msg := 'antecede';
        end
        else if ((Row < (RowCount - 1)) and (Cells[7,Row + 1] <> '') and (data >= StrToDate
(Cells[7,Row + 1]))) then
        begin
            t := Row + 1;
            msg := 'sucede';
        end;
        if (t <> -1) then
            MessageDlg ('O início previsto para a colheita do talhão ' + letra[Row] + ' coincide ou ' + msg
+ ' o início do talhão ' + letra[t] + '.');
            MTWarning, [MbOK], 0);
        end;
    end;
end;
finally
    TxtCell.Visible := false;
    StgPropTal.SetFocus;
end;
except
    on EConvertError do
    begin

```

```

        StgPropTal.Cells[StgPropTal.Col,StgPropTal.Row] := valCell;
        MessageDlg ('O valor do campo é inválido.', MTErrors, [MbOK], 0);
    end;
end;
end;

procedure TFrmPropNovo.DBLookupCboCulKeyPress(Sender: TObject; var Key: Char);
begin
    if (Key = #13) then
    begin
        Key := #0;
        Perform (CM_DialogKey, VK_TAB, 0);
    end;
end;

procedure TFrmPropNovo.TxtCellKeyPress(Sender: TObject; var Key: Char);
begin
    if (Key = #13) then
    begin
        Key := #0;
        Perform (CM_DialogKey, VK_TAB, 0);
    end;
end;

procedure TFrmPropNovo.FuncPerdas(Sender: TObject);
var t, anoAT, c, aln, aPrev, aFi, atrasoStart, atraso, diasPT, diasUT, ARow: Integer;
    pAN, pAT, prodMax, caplInstal, cln, perdTotal, prodTotal, areaC: Real;
    sitData: array of Integer;
begin
    caplInstal := 0;
    for c := 0 to (numCol - 1) do
        caplInstal := caplInstal + (codColhedora[indColhedora[c],1] * codColhedora[indColhedora[c],2] *
codColhedora[indColhedora[c],4] * (Eficiencia / 1000)) * turnoTrab;
    for anoAT := priAno to ultAno do
    begin
        linAno[anoAT - priAno,0] := StgCustoOpt.RowCount - 1;
        StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 1] := 'ANO - ' + IntToStr (anoAT);
        StgCustoOpt.RowCount := StgCustoOpt.RowCount + 2;
        with FrmSimul.QryPrec do
        begin
            Close;
            SQL.Clear;
            SQL.Add ('select *');
            SQL.Add ('from "REGIÃO_Precipitação"');
            SQL.Add ('where CódRegião = ' + IntToStr (DBLookupCboReg.KeyValue));
            SQL.Add ('and extract (year from Data) = ' + IntToStr (anoAT));
            Open;
        end;
        for t := 1 to (StgPropTal.RowCount - 1) do
        begin
            StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 1] := 'TALHÃO ' + letra[t];
            aln := StrToInt (StgPropTal.Cells[8,t]);
            diasPT := 0;
            if (t > 1) then
                if ((StrToDate (StgPropTal.Cells[7,t - 1]) + resultPropAno[t - 2,(anoAT - priAno),2] +
resultPropAno[t - 2,(anoAT - priAno),4]) - StrToDate (StgPropTal.Cells[7,t]) >= 0) then

```

```

begin
  diasPT := Trunc ((StrToDate (StgPropTal.Cells[7,t - 1]) + resultPropAno[t - 2,(anoAT - priAno),2]
+ resultPropAno[t - 2,(anoAT - priAno),4] + 1) - StrToDate (StgPropTal.Cells[7,t]));
  aln := Trunc ((StrToDate (StgPropTal.Cells[7,t]) + diasPT) - (StrToDate (StgPropTal.Cells[6,t]) +
3));
  if (aln < 0) then
    aln := 0;
  end;
  if (aln < 25) then
    begin
      pAT := talCultivar[t,4];
      atrasoStart := -1;
      repeat
        atrasoStart := atrasoStart + 1;
        pAN := pAT;
        pAT := talCultivar[t,2] * Sqr (atrasoStart) + talCultivar[t,3] * atrasoStart + talCultivar[t,4];
      until (pAT > pAN);
      atrasoStart := atrasoStart - 1;
      prodMax := (talCultivar[t,2] * -1) * Sqr (atrasoStart) + (talCultivar[t,3] * -1) * atrasoStart +
talCultivar[t,5];
      with FrmSimul.QryPrec do
        Locate ('Data', FormatDateTime ('dd/mm/' + IntToStr (anoAT), StrToDate (StgPropTal.Cells[7,t])
+ diasPT), [loPartialKey]);
        diasUT := 0;
        pAT := 0;
        atraso := 0;
        repeat
          SetLength (sitData, atraso + 1);
          pAN := pAT;
          pAT := FrmSimul.QryPrec.FieldValues['Precipitação'];
          if ((pAT + pAN) <= precCrit) then
            begin
              sitData[atraso] := 1;
              aFi := atraso;
              diasUT := diasUT + 1;
            end
          else
            sitData[atraso] := 0;
            StgCustoOpt.RowCount := StgCustoOpt.RowCount + 1;
            StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 1] := FormatDateTime ('dd/mm/yy',
FrmSimul.QryPrec.FieldValues['Data']);
            StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 1] := Format ('%n', [pAT]);
            atraso := atraso + 1;
            FrmSimul.QryPrec.Next;
          until (((aln + atraso) > 44) or ((caplInstal * diasUT) >= StrToFloat (StgPropTal.Cells[3,t]));
          aPrev := aln + (atraso - 1);
          perdTotal := 0;
          prodTotal := 0;
          areaC := 0;
          for atraso := aln to aPrev do
            begin
              ARow := StgCustoOpt.RowCount - ((aPrev + 1) - atraso);
              if (sitData[atraso - aln] = 1) then
                begin
                  cIn := caplInstal;
                  if (((atraso - aln) = aFi) and ((caplInstal * diasUT) > StrToFloat (StgPropTal.Cells[3,t]))) then

```

```

    cIn := capInstal - (capInstal * diasUT - StrToFloat (StgPropTal.Cells[3,t]));
    StgCustoOpt.Cells[2,ARow] := 'Útil';
end
else
begin
    cIn := 0;
    StgCustoOpt.Cells[2,ARow] := 'Inútil';
end;
if (atraso > atrasoStart) then
    pAT := talCultivar[t,2] * Sqr(atraso) + talCultivar[t,3] * atraso + talCultivar[t,4]
else
    pAT := 0;
perdTotal := perdTotal + pAT * cIn;
prodTotal := prodTotal + (prodMax - pAT) * cIn;
areaC := areaC + cIn;
StgCustoOpt.Cells[3,ARow] := IntToStr (atraso);
StgCustoOpt.Cells[4,ARow] := Format ('%n', [pAT * cIn]);
StgCustoOpt.Cells[5,ARow] := Format ('%n', [perdTotal]);
StgCustoOpt.Cells[6,ARow] := Format ('%n', [(prodMax - pAT) * cIn]);
StgCustoOpt.Cells[7,ARow] := Format ('%n', [prodTotal]);
StgCustoOpt.Cells[8,ARow] := Format ('%n', [areaC]);
end;
perdTotal := perdTotal + (StrToFloat (StgPropTal.Cells[3,t]) - areaC) * prodMax;
detAno[anoAT - priAno,1] := detAno[anoAT - priAno,1] + diasUT * turnoTrab;
detAno[anoAT - priAno,2] := detAno[anoAT - priAno,2] + perdTotal;
detAno[anoAT - priAno,3] := detAno[anoAT - priAno,3] + prodTotal;
detAno[anoAT - priAno,6] := detAno[anoAT - priAno,6] + areaC / areaTotalProp;
resultPropAno[t - 1,(anoAT - priAno),0] := perdTotal;
resultPropAno[t - 1,(anoAT - priAno),1] := prodTotal;
resultPropAno[t - 1,(anoAT - priAno),2] := diasPT;
resultPropAno[t - 1,(anoAT - priAno),3] := aln;
resultPropAno[t - 1,(anoAT - priAno),4] := aFi;
resultPropAno[t - 1,(anoAT - priAno),5] := diasUT;
resultPropAno[t - 1,(anoAT - priAno),6] := areaC;
end
else
    detAno[anoAT - priAno,0] := 0;
with StgCustoOpt do
begin
    RowCount := RowCount + 8;
    Cells[0,RowCount - 8] := 'RESULTADOS (talhão ' + letra[t] + ')';
    Cells[0,RowCount - 7] := 'Início da colheita:'; Cells[1,RowCount - 7] := FormatDateTime ('dd/mm/'
+ Copy (IntToStr (anoAT), 3, 2), StrToDate (StgPropTal.Cells[7,t]) + diasPT);
    Cells[0,RowCount - 6] := 'Término da colheita:';
    Cells[0,RowCount - 5] := 'Nº de dias úteis:';
    Cells[0,RowCount - 4] := 'Perda total (US$):';
    Cells[0,RowCount - 3] := 'Renda total (US$):';
    if (aln < 25) then
    begin
        Cells[1,RowCount - 6] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[1,RowCount - 7]) +
aFi);
        Cells[1,RowCount - 5] := IntToStr (diasUT);
        Cells[1,RowCount - 4] := Format ('%n', [perdTotal * PrecSaco]);
        Cells[1,RowCount - 3] := Format ('%n', [prodTotal * PrecSaco]);
    end
    else

```

```

        for ARow := (RowCount - 6) to (RowCount - 3) do
            Cells[1,ARow] := 'Colheita não realizada';
        end;
    end;
    with StgCustoOpt do
    begin
        RowCount := StgCustoOpt.RowCount + 7;
        Cells[0,RowCount - 8] := 'RESULTADOS (ano de ' + IntToStr (anoAT) + ')';
        Cells[0,RowCount - 7] := 'Perda total (US$):';Cells[1,RowCount - 7] := Format ('%n', [detAno[anoAT
- priAno,2] * PrecSaco]);
        Cells[0,RowCount - 6] := 'Renda total (US$):';Cells[1,RowCount - 6] := Format ('%n', [detAno[anoAT
- priAno,3] * PrecSaco]);
        Cells[0,RowCount - 5] := 'Ritmo de trabalho (ha/dia):';Cells[1,RowCount - 5] := Format ('%n',
[capInstal]);
        Cells[0,RowCount - 4] := 'Tempo de utilização anual (h):';Cells[1,RowCount - 4] := Format ('%n',
[detAno[anoAT - priAno,1]]);
        Cells[0,RowCount - 3] := 'Situação do ano:';
        if (detAno[anoAT - priAno,0] = 1) then
            Cells[1,RowCount - 3] := 'Válido para a estatística'
        else
            Cells[1,RowCount - 3] := 'Inválido para a estatística';
        end;
    end;
    StgCustoOpt.RowCount := StgCustoOpt.RowCount - 2;
end;

procedure TFrmPropNovo.ServMecanizado(Sender: TObject);
var t, anoAT, c: Integer;
    capInstal, custoDePR, custoDeUT, custoDeSM, custoUnitario: Real;
    capColhedora: array of Real;
begin
    SetLength (capColhedora, numCol);
    capInstal := 0;
    for c := 0 to (numCol - 1) do
    begin
        capColhedora[c] := (codColhedora[indColhedora[c],1] * codColhedora[indColhedora[c],2] *
codColhedora[indColhedora[c],4] * (Eficiencia / 1000)) * turnoTrab;
        capInstal := capInstal + capColhedora[c];
    end;
    for anoAT := priAno to ultAno do
    begin
        linAno[anoAT - priAno,1] := StgCustoMaq.RowCount - 1;
        StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 1] := 'ANO - ' + IntToStr (anoAT);
        StgCustoMaq.RowCount := StgCustoMaq.RowCount + 2;
        if (detAno[anoAT - priAno,0] = 1) then
        begin
            StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 3] :=
StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 3] + ' (válido)';
            for c := 0 to (numCol - 1) do
                with StgCustoMaq do
                begin
                    RowCount := RowCount + 5;
                    Cells[0,RowCount - 7] := 'Marca: ' + modColhedora[indColhedora[c],0];
                    Cells[0,RowCount - 6] := 'Modelo: ' + modColhedora[indColhedora[c],1];
                    Cells[0,RowCount - 5] := 'Plataforma: ' + Format ('%n m', [codColhedora[indColhedora[c],4]]);
                    Cells[0,RowCount - 4] := 'Velocidade: ' + Format ('%n km/h', [codColhedora[indColhedora[c],1]]);

```

```

Cells[0,RowCount - 3] := 'Número de colhedoras: ' + FloatToStr
(codColhedora[indColhedora[c],2]);
Cells[0,RowCount - 2] := 'Capacidade: ' + Format ('%n ha/dia', [capColhedora[c]]);
Cells[0,RowCount - 1] := 'RESULTADOS';
for t := 1 to (StgPropTal.RowCount - 1) do
begin
    custoDePR := (codColhedora[indColhedora[c],5] * (1 - Residual / 100) * ((exp (ln (1 + (Juros /
100)) * VidaUtil) * (Juros / 100)) / (exp (ln (1 + (Juros / 100)) * VidaUtil) - 1))) +
((codColhedora[indColhedora[c],5] * (Residual / 100)) * (Juros / 100)) +
(codColhedora[indColhedora[c],5] * (SegAloj / 100));
    custoDeUT := (((Salario / 220) * 2.1) + (0.184 * codColhedora[indColhedora[c],3]) * Diesel +
(((0.00059 * codColhedora[indColhedora[c],3]) + 0.02169) * (Diesel * 4.5)) + (0.4 *
codColhedora[indColhedora[c],5]) / (VidaUtil * detAno[anoAT - priAno,1])) * detAno[anoAT - priAno,1];
    custoDeSM := (custoDePR + custoDeUT) * codColhedora[indColhedora[c],2];
    custoUnitario := custoDeSM * (resultPropAno[t - 1,(anoAT - priAno),6] / (areaTotalProp *
detAno[anoAT - priAno,6])) + (resultPropAno[t - 1,(anoAT - priAno),0] * PrecSaco * (capColhedora[c] /
caplInstal));
    custoUnitario := custoUnitario / (resultPropAno[t - 1,(anoAT - priAno),1] * (capColhedora[c] /
caplInstal));
    RowCount := RowCount + 1;
    Cells[0,RowCount - 1] := 'Talhão ' + letra[t];
    Cells[1,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,(anoAT - priAno),6] *
(capColhedora[c] / caplInstal)]);
    Cells[2,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,(anoAT - priAno),0] * PrecSaco *
(capColhedora[c] / caplInstal)]);
    Cells[3,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,(anoAT - priAno),1] * PrecSaco *
(capColhedora[c] / caplInstal)]);
    Cells[4,RowCount - 1] := Format ('%n', [custoDeSM * (resultPropAno[t - 1,(anoAT - priAno),6] /
(areaTotalProp * detAno[anoAT - priAno,6]))]);
    Cells[5,RowCount - 1] := Format ('%n', [custoUnitario]);
    colAno[anoAT - priAno,c] := colAno[anoAT - priAno,c] + custoUnitario;
    resultMaqAno[t - 1,(anoAT - priAno),c,0] := custoDeSM * (resultPropAno[t - 1,(anoAT -
priAno),6] / (areaTotalProp * detAno[anoAT - priAno,6]));
    resultMaqAno[t - 1,(anoAT - priAno),c,1] := custoUnitario;
    resultMaqAno[t - 1,(anoAT - priAno),c,2] := resultPropAno[t - 1,(anoAT - priAno),6] *
(capColhedora[c] / caplInstal);
end;
RowCount := RowCount + 5;
detAno[anoAT - priAno,4] := detAno[anoAT - priAno,4] + custoDeSM;
colAno[anoAT - priAno,c] := colAno[anoAT - priAno,c] / (StgPropTal.RowCount - 1);
detAno[anoAT - priAno,5] := detAno[anoAT - priAno,5] + colAno[anoAT - priAno,c];
Cells[0,RowCount - 5] := 'SERVIÇO MECANIZADO (US$)';Cells[1,RowCount - 5] := Format
('%n', [custoDeSM]);
Cells[0,RowCount - 4] := 'CUSTO M. COLHEDORA (US$/sc)';Cells[1,RowCount - 4] := Format
('%n', [colAno[anoAT - priAno,c]]);
end;
detAno[anoAT - priAno,4] := PrecSaco * (detAno[anoAT - priAno,3] - detAno[anoAT - priAno,2]) -
detAno[anoAT - priAno,4];
detAno[anoAT - priAno,5] := detAno[anoAT - priAno,5] / numCol;
StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 2] := 'RENTA LÍQUIDA
(US$)';StgCustoMaq.Cells[1,StgCustoMaq.RowCount - 2] := Format ('%n', [detAno[anoAT -
priAno,4]]);
StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 1] := 'CUSTO M. FROTA
(US$/sc)';StgCustoMaq.Cells[1,StgCustoMaq.RowCount - 1] := Format ('%n', [detAno[anoAT -
priAno,5]]);
StgCustoMaq.RowCount := StgCustoMaq.RowCount + 2;

```

```

end
else
  StgCustoMq.Cells[0,StgCustoMq.RowCount - 3] :=
StgCustoMq.Cells[0,StgCustoMq.RowCount - 3] + ' (inválido)';
end;
StgCustoMq.RowCount := StgCustoMq.RowCount - 2;
end;

procedure TFrmPropNovo.ResultadoFinal(Sender: TObject);
var i, ind, anoAT, ACol: Integer;
    valoresVar: array[0..3] of array of Real;
    valoresTal: array[0..1] of array of Real;
begin
  SetLength (valoresTal, StgPropTal.RowCount - 1);
  ind := 0;
  for anoAT := priAno to ultAno do
    if (detAno[anoAT - priAno,0] = 1) then
      begin
        for i := 0 to 3 do
          SetLength (valoresVar[i], ind + 1);
          valoresVar[0,ind] := (detAno[anoAT - priAno,2] * PrecSaco) / areaTotalProp;
          valoresVar[1,ind] := detAno[anoAT - priAno,4] / areaTotalProp;
          valoresVar[2,ind] := (detAno[anoAT - priAno,3] * PrecSaco - detAno[anoAT - priAno,4]) /
areaTotalProp;
          valoresVar[3,ind] := detAno[anoAT - priAno,5];
        for i := 1 to (StgPropTal.RowCount - 1) do
          begin
            SetLength (valoresTal[i - 1,0], ind + 1);
            SetLength (valoresTal[i - 1,1], ind + 1);
            valoresTal[i - 1,0,ind] := resultPropAno[i - 1,(anoAT - priAno),4] + 1;
            valoresTal[i - 1,1,ind] := resultPropAno[i - 1,(anoAT - priAno),5];
          end;
          ind := ind + 1;
        end;
      with StgResultFinalEst do
        begin
          Cells[0,0] := 'Nº de anos da série:'; Cells[1,0] := IntToStr ((ultAno - priAno) + 1);
          Cells[0,1] := 'Nº de anos válidos:'; Cells[1,1] := IntToStr (ind);
          if (ind > 0) then
            begin
              RowCount := RowCount + 3;
              Cells[0,RowCount - 2] := 'ESTATÍSTICA';
              for ACol := 0 to 4 do
                Cells[ACol,RowCount - 1] := atributo[ACol];
              for i := 0 to 3 do
                begin
                  RowCount := RowCount + 1;
                  CalcularMedias (0, i, valoresVar[i]);
                  Cells[0,RowCount - 1] := variavel[i];
                  for ACol := 1 to 4 do
                    Cells[ACol,RowCount - 1] := Format ('%n', [resultFinalVar[i,ACol - 1]]);
                  end;
                  RowCount := RowCount + 3;
                  Cells[0,RowCount - 2] := 'DETALHES DO TALHÃO';
                  Cells[0,RowCount - 1] := 'Prazo (dias)';
                  for i := 1 to 4 do

```

```

begin
  RowCount := RowCount + 1;
  Cells[0, RowCount - 1] := atributo[i];
end;
ind := RowCount;
RowCount := RowCount + 1;
Cells[0, RowCount - 1] := 'Nº de dias úteis';
for i := 1 to 4 do
begin
  RowCount := RowCount + 1;
  Cells[0, RowCount - 1] := atributo[i];
end;
for ACol := 1 to (StgPropTal.RowCount - 1) do
begin
  if (ACol > (ColCount - 1)) then
    ColCount := ColCount + 1;
    Cells[ACol, ind - 6] := 'Talhão ' + letra[ACol];
    CalcularMedias (1, ACol - 1, valoresTal[ACol - 1, 0]);
    CalcularMedias (2, ACol - 1, valoresTal[ACol - 1, 1]);
    for i := 0 to 3 do
    begin
      Cells[ACol, ind - (4 - i)] := Format ('%n', [resultFinalTal[ACol - 1, 0, i]]);
      Cells[ACol, RowCount - (4 - i)] := Format ('%n', [resultFinalTal[ACol - 1, 1, i]]);
    end;
  end;
end
else
  raise Exception.Create ('Não foi possível gerar a propriedade porque não há nenhum ano na série
temporal');
end;
end;

function TFrmPropNovo.GerarCodigo: Boolean;
var ind, e: Integer;
    strInd: String[3];
begin
  with FrmSimul.QrySimul do
  begin
    Close;
    SQL.Clear;
    SQL.Add ('select *');
    SQL.Add ('from "SIMULAÇÃO"');
    SQL.Add ('where CódSimulação like "PROP.' + DBLookupCboReg.Text + '%"');
    SQL.Add ('and extract (year from DataDeCriação) = ' + FormatDateTime ('yyyy', Date));
    Open;
    FindLast;
    if Found then
      Val (Copy (FieldValues['CódSimulação'], 8, 3), ind, e)
    else
      ind := 0;
      ind := ind + 1;
      if (ind < 100) then
        begin
          strInd := '0';
          if (ind < 10) then
            strInd := strInd + '0';

```

```

end;
strInd := strInd + IntToStr (ind);
codProp := 'PROP.' + DBLookupCboReg.Text + strInd + '-' + FormatDateTime ('yy', Date);
end;
GerarCodigo := true;
LblCod.Caption := codProp;
MessageDlg ('O código da nova propriedade é ' + codProp + '.', MTInformation, [MbOK], 0);
end;

function TFrmPropNovo.SalvarPropriedade: Boolean;
var t, anoAT, c: Integer;
begin
    FrmSimul.QryProp.InsertRecord ([codProp,DBLookupCboReg.KeyValue]);
    FrmSimul.QrySimul.InsertRecord ([codProp,Date,turnoTrab,precCrit,StgPropTal.RowCount - 1]);
    FrmSimul.QrySimulInd.InsertRecord ([codProp,PrecSaco,Juros,VidaUtil,Eficiencia,Salario,
        SegAloj,Diesel,Residual]);
    for c := 0 to (numCol - 1) do
        FrmSimul.QrySimulCol.InsertRecord
        ([codProp,c,codColhedora[indColhedora[c],0],codColhedora[indColhedora[c],1]]);
        for t := 1 to (StgPropTal.RowCount - 1) do
            begin
                FrmSimul.QrySimulTal.InsertRecord
                ([codProp,t,talCultivar[t,1],StgPropTal.Cells[5,t],StgPropTal.Cells[7,t],Null,StgPropTal.Cells[3,t]]);
                for anoAT := priAno to ultAno do
                    if (detAno[anoAT - priAno,0] = 1) then
                        begin
                            FrmSimul.QryResultPropAno.InsertRecord ([codProp,t,anoAT,StrToDate (StgPropTal.Cells[7,t]) +
                                resultPropAno[t - 1,(anoAT - priAno),2],
                                StrToDate (StgPropTal.Cells[7,t]) + resultPropAno[t - 1,(anoAT -
                                priAno),2] + resultPropAno[t - 1,(anoAT - priAno),4],
                                resultPropAno[t - 1,(anoAT - priAno),0],resultPropAno[t - 1,(anoAT -
                                priAno),1],resultPropAno[t - 1,(anoAT - priAno),4],
                                resultPropAno[t - 1,(anoAT - priAno),5]]);
                            for c := 0 to (numCol - 1) do
                                FrmSimul.QryResultMaqAno.InsertRecord
                                ([codProp,t,anoAT,c,codColhedora[indColhedora[c],2],Null,resultMaqAno[t - 1,(anoAT - priAno),c,0],
                                    Null,resultMaqAno[t - 1,(anoAT - priAno),c,1],resultMaqAno[t - 1,(anoAT
                                - priAno),c,2]]);
                            end;
                        end;
                    SalvarPropriedade := true;
                    FrmSimul.QryProp.Refresh;
                    MessageDlg ('A nova propriedade foi salva com êxito.', MTInformation, [MbOK], 0);
                end;
            end;
        end;
    end;

procedure TFrmPropNovo.OpNovoClick(Sender: TObject);
var resp: Word;
    continua: Boolean;
begin
    Screen.Cursor := crHourglass;
    try
        continua := true;
        if (propGerado and Not (propSalvado)) then
            begin
                resp := MessageDlg ('Deseja salvar a nova propriedade?', MTWarning, [MbYes,MbNo,MbCancel],
                    0);
            end;
        end;
    end;
end;

```

```

        if (resp = mrYes) then
            propSalvado := SalvarPropriedade
        else if (resp = mrCancel) then
            continua := false;
        end;
        if continua then
            NovoPropriedade (self);
        finally
            Screen.Cursor := crDefault;
        end;
    end;

procedure TFrmPropNovo.OpSalvarClick(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    try
        if Not (propGerado) then
            raise Exception.Create ('Não há nenhuma propriedade gerada')
        else if (propSalvado) then
            raise Exception.Create ('A nova propriedade já foi salva')
        else
            propSalvado := SalvarPropriedade;
        finally
            Screen.Cursor := crDefault;
        end;
    end;

procedure TFrmPropNovo.OpIndicadorClick(Sender: TObject);
begin
    try
        FrmIndicador := TFrmIndicador.Create (Application);
        FrmIndicador.ShowModal;
    finally
        FrmIndicador.Free;
    end;

procedure TFrmPropNovo.FormClose(Sender: TObject; var Action: TCloseAction);
var resp: Word;
begin
    Screen.Cursor := crHourglass;
    try
        if (propGerado and Not (propSalvado)) then
            begin
                resp := MessageDlg ('Deseja salvar a nova propriedade?', MTWarning, [MbYes,MbNo,MbCancel],
0);
                if (resp = mrYes) then
                    propSalvado := SalvarPropriedade
                else if (resp = mrCancel) then
                    Action := caNone;
            end;
        finally
            Screen.Cursor := crDefault;
        end;
    end;

```

```

procedure TFrmPropNovo.BtnCancelarClick(Sender: TObject);
begin
    Close;
end;

function TFrmPropNovo.LocalizarAno(anoAT, indGrad: Integer): Boolean;
var ARow: Integer;
    valido: Boolean;
begin
    valido := true;
    if ((anoAT < priAno) or (anoAT > ultAno)) then
        valido := false;
    if valido then
        case indGrad of
            0: with StgCustoOpt do
                begin
                    ARow := RowCount - (linAno[anoAT - priAno,0] + 1);
                    if (ARow >= 20) then
                        ARow := 20;
                    Col := 0;
                    Row := linAno[anoAT - priAno,0] + ARow;
                    Row := Row - ARow;
                end;
            1: with StgCustoMaq do
                begin
                    ARow := RowCount - (linAno[anoAT - priAno,1] + 1);
                    if (ARow >= 20) then
                        ARow := 20;
                    Col := 0;
                    Row := linAno[anoAT - priAno,1] + ARow;
                    Row := Row - ARow;
                end;
        end;
    LocalizarAno := valido;
end;

procedure TFrmPropNovo.OpLocalizarAno1Click(Sender: TObject);
var strAno: String;
begin
    try
        if Not (propGerado) then
            raise Exception.Create ('Não há nenhuma propriedade gerada');
        strAno := '';
        if InputQuery ('Série temporal - localizar ano', 'Ano (aaaa)', strAno) then
            if Not (LocalizarAno (StrToInt (strAno), 0)) then
                MessageDlg ('O ano especificado não existe.', MTInformation, [MbOK], 0);
    except
        on EConvertError do
            raise Exception.Create ('O valor fornecido é inválido');
    end;
end;

procedure TFrmPropNovo.OpLocalizarAno2Click(Sender: TObject);
var strAno: String;
begin
    try

```

```

if Not (propGerado) then
    raise Exception.Create ('Não há nenhuma propriedade gerada');
strAno := '';
if InputQuery ('Série temporal - localizar ano', 'Ano (aaaa)', strAno) then
    if Not (LocalizarAno (StrToInt (strAno), 1)) then
        MessageDlg ('O ano especificado não existe.', MTInformation, [MbOK], 0);
except
    on EConvertError do
        raise Exception.Create ('O valor fornecido é inválido');
end;
end;

function TFrmPropNovo.CaixaDeEntrada(cs: Integer; var str1, str2: String): Boolean;
begin
    CaixaDeEntrada := false;
    FrmCaixaDeEntrada := TFrmCaixaDeEntrada.Create (Application);
    try
        with FrmCaixaDeEntrada do
            begin
                Caption := ChkListBoxPropCol.Items.Strings[cs];
                TxtVeloc.Text := str1;
                TxtNumCol.Text := str2;
                if (ShowModal = mrOK) then
                    begin
                        CaixaDeEntrada := true;
                        str1 := TxtVeloc.Text;
                        str2 := TxtNumCol.Text;
                    end;
                end;
            finally
                FrmCaixaDeEntrada.Free;
            end;
        end;
    end;

procedure TFrmPropNovo.ChkListBoxPropColMouseDown(Sender: TObject;
    Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var APoint: TPoint;
begin
    APoint.X := X;
    APoint.Y := Y;
    indCol := ChkListBoxPropCol.ItemAtPos (APoint, true);
end;

procedure TFrmPropNovo.ChkListBoxPropColKeyDown(Sender: TObject;
    var Key: Word; Shift: TShiftState);
begin
    indCol := ChkListBoxPropCol.ItemIndex;
    if (indCol < 0) then
        indCol := 0;
end;

{procedure TFrmPropNovo.DistribuirValores(indVar: Integer; valor: array of Real);
var i, ind, freq: Integer;
    vAT: Real;
    ordenou: Boolean;
begin

```

```

ind := 0;
repeat
  ind := ind + 1;
  ordenou := true;
  for i := High (valor) downto ind do
    if (valor[i] > valor[i - 1]) then
      begin
        vAT := valor[i - 1];
        valor[i - 1] := valor[i];
        valor[i] := vAT;
        ordenou := false;
      end;
  until ordenou or (ind = High (valor));
ind := -1;
for i := 0 to High (valor) do
  if ((i > 0) and (valor[i] = valor[i - 1])) then
    categoria[indVar,ind,1] := categoria[indVar,ind,1] + 1
  else
    begin
      ind := ind + 1;
      SetLength (categoria[indVar], ind + 1);
      categoria[indVar,ind,0] := valor[i];
      categoria[indVar,ind,1] := 1;
    end;
  freq := 0;
  for ind := 0 to High (categoria[indVar]) do
    begin
      freq := freq + Trunc (categoria[indVar,ind,1]);
      categoria[indVar,ind,2] := (freq / (High (valor) + 2)) * 100;
    end;
  end;}
end;}

```

```

procedure TFrmPropNovo.CalcularMedias(opResult, indVar: Integer; valor: array of Real);
var i: Integer;
    min, max, media, desvio: Real;
begin
  min := valor[0];
  max := valor[0];
  media := 0;
  for i := 0 to High (valor) do
    begin
      if (min > valor[i]) then
        min := valor[i]
      else if (max < valor[i]) then
        max := valor[i];
      media := media + valor[i];
    end;
  media := media / (High (valor) + 1);
  desvio := 0;
  for i := 0 to High (valor) do
    desvio := desvio + Sqr (valor[i] - media);
  desvio := Sqrt (desvio / High (valor));
  case opResult of
    0: begin
        resultFinalVar[indVar,0] := min;
        resultFinalVar[indVar,1] := max;
      end;
  end;

```

```

        resultFinalVar[indVar,2] := media;
        resultFinalVar[indVar,3] := desvio;
    end;
1, 2: begin
    resultFinalTal[indVar,opResult - 1,0] := min;
    resultFinalTal[indVar,opResult - 1,1] := max;
    resultFinalTal[indVar,opResult - 1,2] := media;
    resultFinalTal[indVar,opResult - 1,3] := desvio;
end;
end;
end;

procedure TFrmPropNovo.OpExportarClick(Sender: TObject);
var t, ind, i: Integer;
    achou: Boolean;
begin
    if Not (StgResultFinalEst.RowCount > 2) then
        raise Exception.Create ('Não há nenhuma informação para exportar');
    try
        FrmExpExcel := TFrmExpExcel.Create (Application);
        with FrmExpExcel do
            begin
                SetLength (indCultivar, 1);
                SetLength (indColhedora, numCol);
                opSimul := 0;
                codSimul := codProp;
                areaTotalSimul := areaTotalProp;
                stgVirtual := StgResultFinalEst;
                ind := 0;
                for t := 1 to (StgPropTal.RowCount - 1) do
                    if (t > 1) then
                        begin
                            achou := false;
                            i := 0;
                            repeat
                                if (talCultivar[t,1] = indCultivar[i,0]) then
                                    achou := true
                                else
                                    i := i + 1;
                            until (achou or (i > ind));
                            if achou then
                                begin
                                    indCultivar[i,1] := indCultivar[i,1] + StrToFloat (StgPropTal.Cells[3,t]);
                                    indCultivar[i,2] := indCultivar[i,2] + 1;
                                end
                            else
                                begin
                                    ind := ind + 1;
                                    SetLength (indCultivar, ind + 1);
                                    indCultivar[ind,0] := talCultivar[t,1];
                                    indCultivar[ind,1] := StrToFloat (StgPropTal.Cells[3,t]);
                                    indCultivar[ind,2] := 1;
                                end;
                            end;
                        end
                    else
                        begin
                            ind := ind + 1;
                            SetLength (indCultivar, ind + 1);
                            indCultivar[ind,0] := talCultivar[t,1];
                            indCultivar[ind,1] := StrToFloat (StgPropTal.Cells[3,t]);
                            indCultivar[ind,2] := 1;
                        end;
                    end;
                end
            end
        end;
    end;
end;
end;

```

```

        indCultivar[0,0] := talCultivar[1,1];
        indCultivar[0,1] := StrToFloat (StgPropTal.Cells[3,1]);
        indCultivar[0,2] := 1;
    end;
    capInstal := 0;
    for i := 0 to High (indColhedora) do
    begin
        indColhedora[i,0] := codColhedora[FrmPropNovo.indColhedora[i],4];
        indColhedora[i,1] := codColhedora[FrmPropNovo.indColhedora[i],3];
        indColhedora[i,2] := codColhedora[FrmPropNovo.indColhedora[i],2];
        capInstal := capInstal + (codColhedora[FrmPropNovo.indColhedora[i],1] *
codColhedora[FrmPropNovo.indColhedora[i],2] * codColhedora[FrmPropNovo.indColhedora[i],4] *
(Eficiencia / 1000)) * turnoTrab;
    end;
    ShowModal;
    end;
    finally
        FrmExpExcel.Free;
    end;
end;

end.

```

G - Código Fonte 17: UPropriedadeAlterar.pas <<código>>

```

unit UPropriedadeAlterar;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    Menus, StdCtrls, ComCtrls, ToolWin, Buttons, ExtCtrls, DBCtrls, CheckLst,
    Grids, Mask, DB;

type
    TFrmPropAlterar = class(TForm)
        Lbl1: TLabel;
        MainMenuPropAlterar: TMainMenu;
        MenuPropriedade: TMenuItem;
        MenuEditar: TMenuItem;
        OpSalvar: TMenuItem;
        OpIndicador: TMenuItem;
        ToolBarPropAlterar: TToolBar;
        ToolBtnSalvar: TToolButton;
        Pnl1: TPanel;
        Pnl3: TPanel;
        Pnl2: TPanel;
        BtnConfirmar: TBitBtn;
        BtnCancelar: TBitBtn;
        LblCod: TLabel;
        PgeCtrlPropAlterar: TPageControl;
        TabShtProp: TTabSheet;
        TabShtCustoOpt: TTabSheet;
        TabShtCustoMaq: TTabSheet;
    end;

```

```

StatusBarPropAlterar: TStatusBar;
Pnl4: TPanel;
GrpBoxPropCol: TGroupBox;
Pnl7: TPanel;
Pnl8: TPanel;
ChkListBoxPropCol: TCheckListBox;
GrpBoxPropTal: TGroupBox;
Pnl9: TPanel;
StgPropTal: TStringGrid;
PopupMenuPropTal: TPopupMenu;
OpIncluirTal: TMenuItem;
OpExcluirTal: TMenuItem;
MenuN1: TMenuItem;
OpLimparListaTal: TMenuItem;
DBLookupCboCul: TDBLookupComboBox;
TxtCell: TMaskEdit;
StgCustoOpt: TStringGrid;
StgCustoMaq: TStringGrid;
ToolBarCustoOpt: TToolBar;
ToolBtnLocalizarAno1: TToolButton;
ToolBarCustoMaq: TToolBar;
ToolBtnLocalizarAno2: TToolButton;
PopupMenuGrad1: TPopupMenu;
OpLocalizarAno1: TMenuItem;
PopupMenuGrad2: TPopupMenu;
OpLocalizarAno2: TMenuItem;
ToolBarPropTal: TToolBar;
ToolBtnIncluirTal: TToolButton;
ToolBtnExcluirTal: TToolButton;
ToolN1: TToolButton;
ToolBtnLimparListaTal: TToolButton;
Pnl5: TPanel;
GrpBoxProp: TGroupBox;
GrpBoxPer: TGroupBox;
Pnl6: TPanel;
Lbl2: TLabel;
Lbl3: TLabel;
Lbl4: TLabel;
Lbl5: TLabel;
Lbl6: TLabel;
TxtReg: TEdit;
TxtTurno: TEdit;
TxtPrec: TEdit;
CboPriAno: TComboBox;
CboUltAno: TComboBox;
TabShtResultFinal: TTabSheet;
StgResultFinalEst: TStringGrid;
PopupMenuGradResultFinal: TPopupMenu;
OpExportar: TMenuItem;
ToolN2: TToolButton;
ToolBtnIndicador: TToolButton;
procedure ChkListBoxPropColClickCheck(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure OpIncluirTalClick(Sender: TObject);
procedure OpExcluirTalClick(Sender: TObject);
procedure OpLimparListaTalClick(Sender: TObject);

```

```

procedure FormCreate(Sender: TObject);
procedure DBLookupCboCulClick(Sender: TObject);
procedure DBLookupCboCulExit(Sender: TObject);
procedure StgPropTalSelectCell(Sender: TObject; ACol, ARow: Integer;
    var CanSelect: Boolean);
procedure BtnConfirmarClick(Sender: TObject);
procedure StgPropTalGetEditText(Sender: TObject; ACol, ARow: Integer;
    var Value: String);
procedure TxtCellExit(Sender: TObject);
procedure DBLookupCboCulKeyPress(Sender: TObject; var Key: Char);
procedure TxtCellKeyPress(Sender: TObject; var Key: Char);
procedure OpSalvarClick(Sender: TObject);
procedure OpIndicadorClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure BtnCancelarClick(Sender: TObject);
procedure OpLocalizarAno1Click(Sender: TObject);
procedure OpLocalizarAno2Click(Sender: TObject);
procedure ChkListBoxPropColMouseDown(Sender: TObject;
    Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
procedure ChkListBoxPropColKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
procedure OpExportarClick(Sender: TObject);
private
{ Private declarations }
priAno, ultAno, numCol, indCol: Integer;
turnoTrab, precCrit, areaTotalProp: Real;
codColhedora: array of array[0..5] of Real;
modColhedora: array of array[0..1] of String;
indColhedora: array of Integer;
talCultivar: array[1..26,0..5] of Real;
linAno: array of array[0..1] of Integer;
detAno: array of array[0..7] of Real;
colAno: array of array of Real;
resultPropAno: array of array of array[0..6] of Real;
resultMaqAno: array of array of array of array[0..2] of Real;
resultFinalVar: array[0..3] of array[0..3] of Real;
resultFinalTal: array of array[0..1] of array[0..3] of Real;
//categoria: array[0..n] of array of array[0..2] of Real;
valCell, codProp: String;
propAlterado, propSalvado: Boolean;
public
{ Public declarations }
function PesquisarFuncPerdas (codReg, codCul: Integer): Boolean;
function SalvarPropriedade: Boolean;
function LocalizarAno (anoAT, indGrad: Integer): Boolean;
function CaixaDeEntrada (cs: Integer; var str1, str2: String): Boolean;
procedure ListarAnos (codReg: Integer);
procedure ListarColhedoras (Sender: TObject);
procedure LimparPropriedade (opCab: Integer);
procedure RecuperarPropriedade (Sender: TObject);
procedure RecuperarPropTal (codReg, codCul: Integer);
procedure IncluirTal (Sender: TObject);
procedure ExcluirTal (Sender: TObject);
procedure LimparListaTal (Sender: TObject);
procedure FuncPerdas (Sender: TObject);
procedure ServMecanizado (Sender: TObject);

```

```

    procedure ResultadoFinal (Sender: TObject);
    //procedure DistribuirValores (indVar: Integer; valor: array of Real);
    procedure CalcularMedias (opResult, indVar: Integer; valor: array of Real);
end;

var
    FrmPropAlterar: TFrmPropAlterar;

implementation

uses USimulacao, UCaixaDeEntrada, UIndicador, URegiao, UMaquina, UCultivar, UExpExcel;

{$R *.DFM}

procedure TFrmPropAlterar.ListarColhedoras(Sender: TObject);
var cs, c: Integer;
    strModelo: String;
begin
    ChkListBoxPropCol.Items.Clear;
    SetLength (codColhedora, FrmMaq.TblCol.RecordCount);
    SetLength (modColhedora, FrmMaq.TblCol.RecordCount);
    SetLength (indColhedora, numCol);
    FrmMaq.TblCol.First;
    cs := 0;
    c := 0;
    repeat
        with FrmMaq.TblCol do
            begin
                codColhedora[cs,0] := FieldValues['CódColhedora'];
                codColhedora[cs,1] := 0;
                codColhedora[cs,2] := 0;
                codColhedora[cs,3] := FieldValues['Potência'];
                codColhedora[cs,4] := FieldValues['LarguraDePlataforma'];
                codColhedora[cs,5] := FieldValues['PreçoDeAquisição'];
                modColhedora[cs,0] := FieldValues['Marca'];
                modColhedora[cs,1] := FieldValues['Modelo'];
            end;
            strModelo := modColhedora[cs,1];
            strModelo := strModelo + ' (plataforma de ' ;
            strModelo := strModelo + FloatToStr (codColhedora[cs,4]);
            strModelo := strModelo + ' metros)';
            ChkListBoxPropCol.Items.Add (strModelo);
            if ((numCol > 0) and (c < numCol)) then
                with FrmSimul.QrySimulCol do
                    if (codColhedora[cs,0] = FieldValues['CódColhedora']) then
                        begin
                            codColhedora[cs,1] := FieldValues['VelocidadeDeOperação'];
                            indColhedora[c] := cs;
                            ChkListBoxPropCol.Checked[cs] := true;
                            c := c + 1;
                            Next;
                        end;
                    cs := cs + 1;
                    FrmMaq.TblCol.Next;
                until FrmMaq.TblCol.Eof;
            end;
end;

```

```

function TFrmPropAlterar.PesquisarFuncPerdas(codReg, codCul: Integer): Boolean;
var ACol, t: Integer;
    msg: String;
begin
    PesquisarFuncPerdas := true;
    with FrmCultivar.QryRegCul do
    begin
        Close;
        SQL.Clear;
        SQL.Add ('select *');
        SQL.Add ('from "REGIÃO_CULTIVAR"');
        SQL.Add ('where CódRegião = ' + IntToStr (codReg));
        SQL.Add ('and CódCultivar = ' + IntToStr (codCul));
        Open;
        FindFirst;
    end;
    if Not (FrmCultivar.QryRegCul.Found) then
    begin
        PesquisarFuncPerdas := false;
        for ACol := 4 to 8 do
            StgPropTal.Cells[ACol,StgPropTal.Row] := 'Não encontrado';
        MessageDlg ('Não há nenhuma função de perdas que corresponda às descrições fornecidas.',
            MTInformation, [MbOK], 0);
    end
    else
        with StgPropTal do
        begin
            Cells[4,Row] := FrmCultivar.QryRegCul.FieldValues['CicloDeMaturação'];
            if (Cells[5,Row] <> "") then
                if (Cells[5,Row] = 'Não encontrado') then
                    for ACol := 5 to 8 do
                        Cells[ACol,Row] := "
                    else
                        begin
                            Cells[6,Row] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[5,Row]) + StrToInt
(Cells[4,Row]));
                            Cells[7,Row] := Cells[6,Row];
                            Cells[8,Row] := '0';
                            t := -1;
                            if ((Row > 1) and (Cells[7,Row - 1] <> "") and (StrToDate (Cells[7,Row]) <= StrToDate
(Cells[7,Row - 1]))) then
                                begin
                                    t := Row - 1;
                                    msg := 'antecede';
                                end
                                else if ((Row < (RowCount - 1)) and (Cells[7,Row + 1] <> "") and (StrToDate (Cells[7,Row]) >=
StrToDate (Cells[7,Row + 1]))) then
                                    begin
                                        t := Row + 1;
                                        msg := 'sucede';
                                    end;
                                if (t <> -1) then
                                    MessageDlg ('O início previsto para a colheita do talhão ' + letra[Row] + ' coincide ou ' + msg +
' o início do talhão ' + letra[t] + '.',
                                        MTWarning, [MbOK], 0);

```

```

        end;
    end;
end;

procedure TFrmPropAlterar.ChkListBoxPropColClickCheck(Sender: TObject);
var n: Integer;
    v: Real;
    strVeloc, strNumCol: String;
    valido, continua: Boolean;
begin
    try
        if (indCol = ChkListBoxPropCol.ItemIndex) then
            if ChkListBoxPropCol.Checked[indCol] then
                begin
                    numCol := numCol + 1;
                    strVeloc := FloatToStr (codColhedora[indCol,1]);
                    strNumCol := FloatToStr (codColhedora[indCol,2]);
                    valido := false;
                    repeat
                        continua := CaixaDeEntrada (indCol, strVeloc, strNumCol);
                        if continua then
                            begin
                                v := StrToFloat (strVeloc);
                                n := StrToInt (strNumCol);
                                if ((v < 3) or (v > 7)) then
                                    MessageDlg ('A velocidade de operação é inválida.', MError, [MbOK], 0)
                                else if (n < 1) then
                                    MessageDlg ('O número de colhedoras é inválido.', MError, [MbOK], 0)
                                else
                                    begin
                                        valido := true;
                                        codColhedora[indCol,1] := v;
                                        codColhedora[indCol,2] := n;
                                    end;
                                end;
                            until (valido or Not (continua));
                        if Not (continua) then
                            begin
                                ChkListBoxPropCol.Checked[indCol] := false;
                                numCol := numCol - 1;
                            end;
                        end
                    else
                        numCol := numCol - 1
                    end
                ChkListBoxPropCol.Checked[indCol] := Not (ChkListBoxPropCol.Checked[indCol]);
            except
                on EConvertError do
                    begin
                        MessageDlg ('O valor fornecido é inválido.', MError, [MbOK], 0);
                        indCol := ChkListBoxPropCol.ItemIndex;
                        if ((codColhedora[indCol,1] = 0) or (codColhedora[indCol,2] = 0)) then
                            begin
                                ChkListBoxPropCol.Checked[indCol] := false;
                                numCol := numCol - 1;
                            end;
                        end;
                    end;
            end;
        end;
    end;
end;

```

```

    end;
    end;
end;

procedure TFrmPropAlterar.FormShow(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    try
        RecuperarPropriedade (self);
    finally
        Screen.Cursor := crDefault;
    end;
end;

procedure TFrmPropAlterar.IncluirTal(Sender: TObject);
begin
    with StgPropTal do
    begin
        RowCount := RowCount + 1;
        Rows[RowCount - 1].Clear;
        Cells[0, RowCount - 1] := letra[RowCount - 1];
        talCultivar[RowCount - 1, 0] := 0;
        talCultivar[RowCount - 1, 1] := 0;
    end;
end;

procedure TFrmPropAlterar.ExcluirTal(Sender: TObject);
var ARow: Integer;
begin
    with StgPropTal do
    begin
        if (Row < (RowCount - 1)) then
            for ARow := Row to (RowCount - 2) do
            begin
                Rows[ARow] := Rows[ARow + 1];
                Cells[0, ARow] := letra[ARow];
                talCultivar[ARow] := talCultivar[ARow + 1];
            end;
            if (RowCount > 2) then
                RowCount := RowCount - 1
            else
                begin
                    Rows[1].Clear;
                    Cells[0, 1] := letra[1];
                    talCultivar[1, 0] := 0;
                    talCultivar[1, 1] := 0;
                end;
            end;
        end;
    end;

procedure TFrmPropAlterar.LimparListaTal(Sender: TObject);
var ARow: Integer;
begin
    with StgPropTal do
    begin
        for ARow := (RowCount - 1) downto 1 do

```

```

    Rows[ARow].Clear;
    RowCount := 2;
    Cells[0,1] := letra[1];
    talCultivar[1,0] := 0;
    talCultivar[1,1] := 0;
end;
end;

procedure TFrmPropAlterar.OpIncluirTalClick(Sender: TObject);
begin
    if ((StgPropTal.RowCount < 27) and (talCultivar[StgPropTal.RowCount - 1,1] <> 0)) then
        IncluirTal (self);
    end;
end;

procedure TFrmPropAlterar.OpExcluirTalClick(Sender: TObject);
begin
    ExcluirTal (self);
end;

procedure TFrmPropAlterar.OpLimparListaTalClick(Sender: TObject);
begin
    LimparListaTal (self);
end;

procedure TFrmPropAlterar.FormCreate(Sender: TObject);
begin
    with StgPropTal do
        begin
            Cells[0,0] := 'Talhão';ColWidths[0] := 40;
            Cells[1,0] := 'Cultivar';ColWidths[1] := 127;
            Cells[2,0] := 'Grupo de maturação';ColWidths[2] := 104;
            Cells[3,0] := 'Área de lavoura (ha)';ColWidths[3] := 103;
            Cells[4,0] := 'Ciclo de maturação (dias)';ColWidths[4] := 126;
            Cells[5,0] := 'Data de semeadura';ColWidths[5] := 100;
            Cells[6,0] := 'Data provável de colheita';ColWidths[6] := 129;
            Cells[7,0] := 'Início previsto da colheita';ColWidths[7] := 129;
            Cells[8,0] := 'Atraso inicial (dias)';ColWidths[8] := 94;
            DefaultRowHeight := DBLookupCboCul.Height;
        end;
    StgCustoOpt.DefaultRowHeight := StgPropTal.DefaultRowHeight;
    with StgCustoMaq do
        begin
            Cells[0,0] := 'Ano/Colhedora/Talhão';ColWidths[0] := 177;
            Cells[1,0] := 'Área colhida (ha)';ColWidths[1] := 87;
            Cells[2,0] := 'Perda total (US$)';ColWidths[2] := 110;
            Cells[3,0] := 'Renda total (US$)';ColWidths[3] := 92;
            Cells[4,0] := 'Custo de serviço mecanizado (US$)';ColWidths[4] := 176;
            Cells[5,0] := 'Custo unitário (US$/sc)';ColWidths[5] := 117;
            DefaultRowHeight := StgPropTal.DefaultRowHeight;
        end;
    with StgResultFinalEst do
        begin
            ColWidths[0] := 136;
            ColWidths[4] := 76;
            DefaultRowHeight := StgPropTal.DefaultRowHeight;
        end;
    end;
end;

```

```

    TxtCell.BorderStyle := bsNone;
end;

procedure TFrmPropAlterar.DBLookupCboCulClick(Sender: TObject);
begin
    StgPropTal.Cells[2,StgPropTal.Row] := FrmCultivar.TblCul.FieldValues['GrupoDeMaturação'];
    talCultivar[StgPropTal.Row,0] := 1;
    if PesquisarFuncPerdas (FrmReg.TblReg.FieldValues['CódRegião'], DBLookupCboCul.KeyValue)
then
    with FrmCultivar.QryRegCul do
    begin
        talCultivar[StgPropTal.Row,1] := DBLookupCboCul.KeyValue;
        talCultivar[StgPropTal.Row,2] := FieldValues['CoeficienteDePerdA'];
        talCultivar[StgPropTal.Row,3] := FieldValues['CoeficienteDePerdB'];
        talCultivar[StgPropTal.Row,4] := FieldValues['CoeficienteDePerdC'];
        talCultivar[StgPropTal.Row,5] := FieldValues['CoeficienteDeProdC'];
    end
    else
        talCultivar[StgPropTal.Row,1] := 0;
    end;

procedure TFrmPropAlterar.DBLookupCboCulExit(Sender: TObject);
var resp: Word;
    continua: Boolean;
begin
    continua := false;
    if (talCultivar[StgPropTal.Row,0] = 1) then
    begin
        if (talCultivar[StgPropTal.Row,1] <> 0) then
            StgPropTal.Cells[1,StgPropTal.Row] := DBLookupCboCul.Text
        else
            begin
                resp := MessageDlg ('O cultivar é inválido. Pressione OK para selecionar outro ou Cancel para
excluir o talhão.',
                                MTError, [MbOK,MbCancel], 0);
                if (resp = mrOK) then
                begin
                    continua := true;
                    DBLookupCboCul.SetFocus;
                end
                else
                    ExcluirTal (self);
                end;
            end;
        if Not (continua) then
        begin
            DBLookupCboCul.Visible := false;
            StgPropTal.SetFocus;
        end;
    end;

procedure TFrmPropAlterar.StgPropTalSelectCell(Sender: TObject; ACol,
    ARow: Integer; var CanSelect: Boolean);
var cod: Real;
    R: TRect;
begin

```

```

if (ACol = 1) then
begin
  R := StgPropTal.CellRect (ACol,ARow);
  R.Left := R.Left + StgPropTal.Left;
  R.Right := R.Right + StgPropTal.Left;
  R.Top := R.Top + StgPropTal.Top;
  R.Bottom := R.Bottom + StgPropTal.Top;
  with DBLookupCboCul do
  begin
    Left := R.Left + 1;
    Top := R.Top + 1;
    Width := (R.Right + 1) - R.Left;
    Height := (R.Bottom + 1) - R.Top;
    if (talCultivar[ARow,1] <> 0) then
      cod := talCultivar[ARow,1]
    else
      cod := 1;
    FrmCultivar.TblCul.Edit;
    KeyValue := cod;
    Visible := true;
    SetFocus;
  end;
end;
end;

procedure TFrmPropAlterar.BtnConfirmarClick(Sender: TObject);
var t, anoAT, c, cs: Integer;
begin
  Screen.Cursor := crHourglass;
  try try
    if (TxtTurno.Text = "") then
      raise Exception.Create ('Preencha o campo "Turno de trabalho".');
    else if (TxtPrec.Text = "") then
      raise Exception.Create ('Preencha o campo "Precipitação crítica".');
    turnoTrab := StrToFloat (TxtTurno.Text);
    precCrit := StrToFloat (TxtPrec.Text);
    priAno := StrToInt (CboPriAno.Items[CboPriAno.ItemIndex]);
    ultAno := StrToInt (CboUltAno.Items[CboUltAno.ItemIndex]);
    if (turnoTrab > FrmReg.TblReg.FieldValues['TurnoDeTrabalho']) then
      raise Exception.Create ('O turno de trabalho é inválido')
    else if ((ultAno - priAno) < 1) then
      raise Exception.Create ('O período de simulação é inválido')
    else if (numCol < 1) then
      raise Exception.Create ('Deve haver no mínimo uma colhedora selecionada');
    areaTotalProp := 0;
    for t := 1 to (StgPropTal.RowCount - 1) do
      begin
        if Not (talCultivar[t,1] <> 0) then
          raise Exception.Create ('Nenhum cultivar foi atribuído ao talhão ' + letra[t])
        else if (StgPropTal.Cells[3,t] = "") then
          raise Exception.Create ('Forneça a área de lavoura do talhão ' + letra[t])
        else if (StgPropTal.Cells[5,t] = "") then
          raise Exception.Create ('Forneça a data de semeadura do talhão ' + letra[t])
        else if ((t > 1) and (StrToDate (StgPropTal.Cells[7,t - 1]) >= StrToDate (StgPropTal.Cells[7,t]))) then
          raise Exception.Create ('O início previsto para a colheita do talhão ' + letra[t - 1] + ' coincide ou
sucedo o início do talhão ' + letra[t] + '.');
      end;
    end;
  except
    raise Exception.Create ('Erro ao confirmar a alteração');
  end;
end;

```

```

    areaTotalProp := areaTotalProp + StrToFloat (StgPropTal.Cells[3,t]);
end;
SetLength (indColhedora, numCol);
SetLength (linAno, (ultAno + 1) - priAno);
SetLength (detAno, (ultAno + 1) - priAno);
SetLength (colAno, (ultAno + 1) - priAno);
SetLength (resultPropAno, StgPropTal.RowCount - 1);
SetLength (resultMaqAno, StgPropTal.RowCount - 1);
SetLength (resultFinalTal, StgPropTal.RowCount - 1);
c := 0;
for cs := 0 to (ChkListBoxPropCol.Items.Count - 1) do
    if ChkListBoxPropCol.Checked[cs] then
        begin
            indColhedora[c] := cs;
            c := c + 1;
        end;
for t := 1 to (StgPropTal.RowCount - 1) do
    begin
        SetLength (resultPropAno[t - 1], (ultAno + 1) - priAno);
        SetLength (resultMaqAno[t - 1], (ultAno + 1) - priAno);
        for anoAT := priAno to ultAno do
            begin
                SetLength (resultMaqAno[t - 1,(anoAT - priAno)], numCol);
                if (t = 1) then
                    begin
                        SetLength (colAno[anoAT - priAno], numCol);
                        detAno[anoAT - priAno,0] := anoAT;
                        detAno[anoAT - priAno,1] := 1;
                        detAno[anoAT - priAno,2] := 0;
                        detAno[anoAT - priAno,3] := 0;
                        detAno[anoAT - priAno,4] := 0;
                        detAno[anoAT - priAno,5] := 0;
                        detAno[anoAT - priAno,6] := 0;
                        detAno[anoAT - priAno,7] := 0;
                        for c := 0 to (numCol - 1) do
                            colAno[anoAT - priAno,c] := 0;
                        end;
                    end;
                end;
            end;
        LimparPropriedade (1);
        FuncPerdas (self);
        ServMecanizado (self);
        ResultadoFinal (self);
        propAlterado := true;
        MessageDlg ('A propriedade foi alterada com êxito.', MTInformation, [MbOK], 0);
    finally
        Screen.Cursor := crDefault;
    end;
except
    on EConvertError do
        raise Exception.Create ('O valor do campo é inválido');
    end;
end;

procedure TFrmPropAlterar.StgPropTal.GetEditText(Sender: TObject; ACol,
    ARow: Integer; var Value: String);

```

```

var R: TRect;
begin
  valCell := StgPropTal.Cells[ACol,ARow];
  R := StgPropTal.CellRect (ACol,ARow);
  R.Left := R.Left + StgPropTal.Left;
  R.Right := R.Right + StgPropTal.Left;
  R.Top := R.Top + StgPropTal.Top;
  R.Bottom := R.Bottom + StgPropTal.Top;
  with TxtCell do
  begin
    case ACol of
      0, 1, 2, 3, 4, 8: EditMask := "";
      5, 6, 7: EditMask := '!99/99/99;1;_';
    end;
    Left := R.Left + 2;
    Top := R.Top + 2;
    Width := R.Right - R.Left;
    Height := R.Bottom - R.Top;
    Text := valCell;
    Visible := true;
    SetFocus;
  end;
end;

procedure TFrmPropAlterar.TxtCellExit(Sender: TObject);
var t, atraso: Integer;
    msg: String;
    valido: Boolean;
    data: TDateTime;
begin
  try try
    case StgPropTal.Col of
      0, 1, 2, 4, 6, 8: begin
        StgPropTal.Cells[StgPropTal.Col,StgPropTal.Row] := valCell;
        if (valCell <> TxtCell.Text) then
          MessageDlg ('O valor não pode ser alterado porque o campo "' +
            StgPropTal.Cells[StgPropTal.Col,0] + '" está aberto somente para leitura.',
            MTError, [MbOK], 0);
        end;
      3: StgPropTal.Cells[3,StgPropTal.Row] := FloatToStr (StrToFloat (TxtCell.Text));
      5: with StgPropTal do
        begin
          valido := true;
          data := StrToDate (TxtCell.Text);
          if (data < StrToDate (FormatDateTime ('dd/mm/' + Copy (IntToStr (StrToInt (FormatDateTime
('yyyy', Date)) - 1), 3, 2), FrmReg.TblReg.FieldValues['PriDataDeSemeadura'])))
            or (data > StrToDate (FormatDateTime ('dd/mm/' + Copy (IntToStr (StrToInt (FormatDateTime
('yyyy', Date)) - 1), 3, 2), FrmReg.TblReg.FieldValues['UltDataDeSemeadura']))) then
            valido := false
          else
            for t := 1 to (RowCount - 1) do
              if ((t <> Row) and (talCultivar[t,1] = talCultivar[Row,1]) and (Cells[5,t] = TxtCell.Text)) then
                valido := false;
            if valido then
              begin
                Cells[5,Row] := TxtCell.Text;

```

```

Cells[6,Row] := FormatDateTime ('dd/mm/yy', data + StrToInt (Cells[4,Row]));
Cells[7,Row] := Cells[6,Row];
Cells[8,Row] := '0';
t := -1;
if ((Row > 1) and (Cells[7,Row - 1] <> '') and (StrToDate (Cells[7,Row]) <= StrToDate
(Cells[7,Row - 1]))) then
begin
t := Row - 1;
msg := 'antecede';
end
else if ((Row < (RowCount - 1)) and (Cells[7,Row + 1] <> '') and (StrToDate (Cells[7,Row]) >=
StrToDate (Cells[7,Row + 1]))) then
begin
t := Row + 1;
msg := 'sucede';
end;
if (t <> - 1) then
MessageDlg ('O início previsto para a colheita do talhão ' + letra[Row] + ' coincide ou ' + msg
+ ' o início do talhão ' + letra[t] + ',
MTWarning, [MbOK], 0);
end
else
begin
Cells[5,Row] := valCell;
MessageDlg ('A data de semeadura é inválida.', MTErrror, [MbOK], 0);
end;
end;
7: with StgPropTal do
begin
data := StrToDate (TxtCell.Text);
if ((data < (StrToDate (Cells[6,Row]) - 3)) or (data > (StrToDate (Cells[6,Row]) + 27))) then
begin
Cells[7,Row] := valCell;
MessageDlg ('A data de início previsto da colheita é inválida.', MTErrror, [MbOK], 0);
end
else
begin
atraso := 0;
if (data > (StrToDate (Cells[6,Row]) + 3)) then
begin
atraso := Trunc (data - (StrToDate (Cells[6,Row]) + 3));
MessageDlg ('A data ' + TxtCell.Text + ' excede em ' + FloatToStr (atraso) + ' dia(s) o limite
máximo para o início provável da colheita.',
MTWarning, [MbOK], 0);
end;
Cells[7,Row] := TxtCell.Text;
Cells[8,Row] := IntToStr (atraso);
t := -1;
if ((Row > 1) and (Cells[7,Row - 1] <> '') and (data <= StrToDate (Cells[7,Row - 1]))) then
begin
t := Row - 1;
msg := 'antecede';
end
else if ((Row < (RowCount - 1)) and (Cells[7,Row + 1] <> '') and (data >= StrToDate
(Cells[7,Row + 1]))) then
begin

```

```

        t := Row + 1;
        msg := 'sucede';
    end;
    if (t <> -1) then
        MessageDlg ('O início previsto para a colheita do talhão ' + letra[Row] + ' coincide ou ' + msg
+ ' o início do talhão ' + letra[t] + '.');
        MTWarning, [MbOK], 0);
    end;
end;
end;
finally
    TxtCell.Visible := false;
    StgPropTal.SetFocus;
end;
except
    on EConvertError do
    begin
        StgPropTal.Cells[StgPropTal.Col,StgPropTal.Row] := valCell;
        MessageDlg ('O valor do campo é inválido.', MTError, [MbOK], 0);
    end;
end;
end;
end;

```

```

procedure TFrmPropAlterar.DBLookupCboCulKeyPress(Sender: TObject;
var Key: Char);
begin
    if (Key = #13) then
    begin
        Key := #0;
        Perform (CM_DialogKey, VK_TAB, 0);
    end;
end;

```

```

procedure TFrmPropAlterar.TxtCellKeyPress(Sender: TObject; var Key: Char);
begin
    if (Key = #13) then
    begin
        Key := #0;
        Perform (CM_DialogKey, VK_TAB, 0);
    end;
end;

```

```

procedure TFrmPropAlterar.FuncPerdas(Sender: TObject);
var t, anoAT, c, aln, aPrev, aFi, atrasoStart, atraso, diasPT, diasUT, ARow: Integer;
    pAN, pAT, prodMax, caplInstal, cIn, perdTotal, prodTotal, areaC: Real;
    sitData: array of Integer;
begin
    caplInstal := 0;
    for c := 0 to (numCol - 1) do
        caplInstal := caplInstal + (codColhedora[indColhedora[c],1] * codColhedora[indColhedora[c],2] *
codColhedora[indColhedora[c],4] * (Eficiencia / 1000)) * turnoTrab;
    for anoAT := priAno to ultAno do
    begin
        linAno[anoAT - priAno,0] := StgCustoOpt.RowCount - 1;
        StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 1] := 'ANO - ' + IntToStr (anoAT);
        StgCustoOpt.RowCount := StgCustoOpt.RowCount + 2;
    end;
end;

```

```

with FrmSimul.QryPrec do
begin
  Close;
  SQL.Clear;
  SQL.Add ('select *');
  SQL.Add ('from "REGIÃO_Precipitação"');
  SQL.Add ('where CódRegião = ' + IntToStr (FrmReg.TblReg.FieldValues['CódRegião']));
  SQL.Add ('and extract (year from Data) = ' + IntToStr (anoAT));
  Open;
end;
for t := 1 to (StgPropTal.RowCount - 1) do
begin
  StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 1] := 'TALHÃO ' + letra[t];
  aln := StrToInt (StgPropTal.Cells[8,t]);
  diasPT := 0;
  if (t > 1) then
    if ((StrToDate (StgPropTal.Cells[7,t - 1]) + resultPropAno[t - 2,(anoAT - priAno),2] +
resultPropAno[t - 2,(anoAT - priAno),4]) - StrToDate (StgPropTal.Cells[7,t]) >= 0) then
      begin
        diasPT := Trunc ((StrToDate (StgPropTal.Cells[7,t - 1]) + resultPropAno[t - 2,(anoAT - priAno),2]
+ resultPropAno[t - 2,(anoAT - priAno),4] + 1) - StrToDate (StgPropTal.Cells[7,t]));
        aln := Trunc ((StrToDate (StgPropTal.Cells[7,t]) + diasPT) - (StrToDate (StgPropTal.Cells[6,t]) +
3));
        if (aln < 0) then
          aln := 0;
        end;
        if (aln < 25) then
          begin
            pAT := talCultivar[t,4];
            atrasoStart := -1;
            repeat
              atrasoStart := atrasoStart + 1;
              pAN := pAT;
              pAT := talCultivar[t,2] * Sqr (atrasoStart) + talCultivar[t,3] * atrasoStart + talCultivar[t,4];
            until (pAT > pAN);
            atrasoStart := atrasoStart - 1;
            prodMax := (talCultivar[t,2] * -1) * Sqr (atrasoStart) + (talCultivar[t,3] * -1) * atrasoStart +
talCultivar[t,5];
            with FrmSimul.QryPrec do
              Locate ('Data', FormatDateTime ('dd/mm/' + IntToStr (anoAT), StrToDate (StgPropTal.Cells[7,t])
+ diasPT), [loPartialKey]);
              diasUT := 0;
              pAT := 0;
              atraso := 0;
              repeat
                SetLength (sitData, atraso + 1);
                pAN := pAT;
                pAT := FrmSimul.QryPrec.FieldValues['Precipitação'];
                if ((pAT + pAN) <= precCrit) then
                  begin
                    sitData[atraso] := 1;
                    aFi := atraso;
                    diasUT := diasUT + 1;
                  end
                else
                  sitData[atraso] := 0;
              end
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

    StgCustoOpt.RowCount := StgCustoOpt.RowCount + 1;
    StgCustoOpt.Cells[0,StgCustoOpt.RowCount - 1] := FormatDateTime ('dd/mm/yy',
FrmSimul.QryPrec.FieldValues['Data']);
    StgCustoOpt.Cells[1,StgCustoOpt.RowCount - 1] := Format ('%n', [pAT]);
    atraso := atraso + 1;
    FrmSimul.QryPrec.Next;
until (((aln + atraso) > 44) or ((capInstal * diasUT) >= StrToFloat (StgPropTal.Cells[3,t])));
aPrev := aln + (atraso - 1);
perdTotal := 0;
prodTotal := 0;
areaC := 0;
for atraso := aln to aPrev do
begin
    ARow := StgCustoOpt.RowCount - ((aPrev + 1) - atraso);
    if (sitData[atraso - aln] = 1) then
    begin
        cln := capInstal;
        if (((atraso - aln) = aFi) and ((capInstal * diasUT) > StrToFloat (StgPropTal.Cells[3,t]))) then
            cln := capInstal - (capInstal * diasUT - StrToFloat (StgPropTal.Cells[3,t]));
        StgCustoOpt.Cells[2,ARow] := 'Útil';
    end
    else
    begin
        cln := 0;
        StgCustoOpt.Cells[2,ARow] := 'Inútil';
    end;
    if (atraso > atrasoStart) then
        pAT := talCultivar[t,2] * Sqr(atraso) + talCultivar[t,3] * atraso + talCultivar[t,4]
    else
        pAT := 0;
    perdTotal := perdTotal + pAT * cln;
    prodTotal := prodTotal + (prodMax - pAT) * cln;
    areaC := areaC + cln;
    StgCustoOpt.Cells[3,ARow] := IntToStr (atraso);
    StgCustoOpt.Cells[4,ARow] := Format ('%n', [pAT * cln]);
    StgCustoOpt.Cells[5,ARow] := Format ('%n', [perdTotal]);
    StgCustoOpt.Cells[6,ARow] := Format ('%n', [(prodMax - pAT) * cln]);
    StgCustoOpt.Cells[7,ARow] := Format ('%n', [prodTotal]);
    StgCustoOpt.Cells[8,ARow] := Format ('%n', [areaC]);
end;
perdTotal := perdTotal + (StrToFloat (StgPropTal.Cells[3,t]) - areaC) * prodMax;
detAno[anoAT - priAno,2] := detAno[anoAT - priAno,2] + diasUT * turnoTrab;
detAno[anoAT - priAno,3] := detAno[anoAT - priAno,3] + perdTotal;
detAno[anoAT - priAno,4] := detAno[anoAT - priAno,4] + prodTotal;
detAno[anoAT - priAno,7] := detAno[anoAT - priAno,7] + areaC / areaTotalProp;
resultPropAno[t - 1,(anoAT - priAno),0] := perdTotal;
resultPropAno[t - 1,(anoAT - priAno),1] := prodTotal;
resultPropAno[t - 1,(anoAT - priAno),2] := diasPT;
resultPropAno[t - 1,(anoAT - priAno),3] := aln;
resultPropAno[t - 1,(anoAT - priAno),4] := aFi;
resultPropAno[t - 1,(anoAT - priAno),5] := diasUT;
resultPropAno[t - 1,(anoAT - priAno),6] := areaC;
end
else
    detAno[anoAT - priAno,1] := 0;
with StgCustoOpt do

```

```

begin
  RowCount := RowCount + 8;
  Cells[0, RowCount - 8] := 'RESULTADOS (talhão ' + letra[t] + ')';
  Cells[0, RowCount - 7] := 'Início da colheita:'; Cells[1, RowCount - 7] := FormatDateTime('dd/mm/' +
Copy (IntToStr (anoAT), 3, 2), StrToDate (StgPropTal.Cells[7,t]) + diasPT);
  Cells[0, RowCount - 6] := 'Término da colheita:';
  Cells[0, RowCount - 5] := 'Nº de dias úteis:';
  Cells[0, RowCount - 4] := 'Perda total (US$):';
  Cells[0, RowCount - 3] := 'Renda total (US$):';
  if (aln < 25) then
    begin
      Cells[1, RowCount - 6] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[1, RowCount - 7]) +
aFi);
      Cells[1, RowCount - 5] := IntToStr (diasUT);
      Cells[1, RowCount - 4] := Format ('%n', [perdTotal * PrecSaco]);
      Cells[1, RowCount - 3] := Format ('%n', [prodTotal * PrecSaco]);
    end
  else
    for ARow := (RowCount - 6) to (RowCount - 3) do
      Cells[1, ARow] := 'Colheita não realizada';
    end;
  end;
end;
with StgCustoOpt do
begin
  RowCount := StgCustoOpt.RowCount + 7;
  Cells[0, RowCount - 8] := 'RESULTADOS (ano de ' + IntToStr (anoAT) + ')';
  Cells[0, RowCount - 7] := 'Perda total (US$):'; Cells[1, RowCount - 7] := Format ('%n', [detAno[anoAT
- priAno,3] * PrecSaco]);
  Cells[0, RowCount - 6] := 'Renda total (US$):'; Cells[1, RowCount - 6] := Format ('%n', [detAno[anoAt
- priAno,4] * PrecSaco]);
  Cells[0, RowCount - 5] := 'Ritmo de trabalho (ha/dia):'; Cells[1, RowCount - 5] := Format ('%n',
[capInstal]);
  Cells[0, RowCount - 4] := 'Tempo de utilização anual (h):'; Cells[1, RowCount - 4] := Format ('%n',
[detAno[anoAT - priAno,2]]);
  Cells[0, RowCount - 3] := 'Situação do ano:';
  if (detAno[anoAT - priAno,1] = 1) then
    Cells[1, RowCount - 3] := 'Válido para a estatística'
  else
    Cells[1, RowCount - 3] := 'Inválido para a estatística';
  end;
end;
end;
StgCustoOpt.RowCount := StgCustoOpt.RowCount - 2;
end;

procedure TFrmPropAlterar.ServMecanizado(Sender: TObject);
var t, anoAT, c: Integer;
    capInstal, custoDePR, custoDeUT, custoDeSM, custoUnitario: Real;
    capColhedora: array of Real;
begin
  SetLength (capColhedora, numCol);
  capInstal := 0;
  for c := 0 to (numCol - 1) do
    begin
      capColhedora[c] := (codColhedora[indColhedora[c],1] * codColhedora[indColhedora[c],2] *
codColhedora[indColhedora[c],4] * (Eficiencia / 1000)) * turnoTrab;
      capInstal := capInstal + capColhedora[c];
    end;
  end;
end;

```

```

end;
for anoAT := priAno to ultAno do
begin
  linAno[anoAT - priAno,1] := StgCustoMaq.RowCount - 1;
  StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 1] := 'ANO - ' + IntToStr (anoAT);
  StgCustoMaq.RowCount := StgCustoMaq.RowCount + 2;
  if (detAno[anoAT - priAno,1] = 1) then
  begin
    StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 3] :=
    StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 3] + ' (válido)';
    for c := 0 to (numCol - 1) do
      with StgCustoMaq do
      begin
        RowCount := RowCount + 5;
        Cells[0,RowCount - 7] := 'Marca: ' + modColhedora[indColhedora[c],0];
        Cells[0,RowCount - 6] := 'Modelo: ' + modColhedora[indColhedora[c],1];
        Cells[0,RowCount - 5] := 'Plataforma: ' + Format ('%n m', [codColhedora[indColhedora[c],4]]);
        Cells[0,RowCount - 4] := 'Velocidade: ' + Format ('%n km/h', [codColhedora[indColhedora[c],1]]);
        Cells[0,RowCount - 3] := 'Número de colhedoras: ' + FloatToStr
        (codColhedora[indColhedora[c],2]);
        Cells[0,RowCount - 2] := 'Capacidade: ' + Format ('%n ha/dia', [capColhedora[c]]);
        Cells[0,RowCount - 1] := 'RESULTADOS';
        for t := 1 to (StgPropTal.RowCount - 1) do
          begin
            custoDePR := (codColhedora[indColhedora[c],5] * (1 - Residual / 100) * ((exp (ln (1 + (Juros /
            100)) * VidaUtil) * (Juros / 100)) / (exp (ln (1 + (Juros / 100)) * VidaUtil) - 1))) +
            (((codColhedora[indColhedora[c],5] * (Residual / 100)) * (Juros / 100)) +
            (codColhedora[indColhedora[c],5] * (SegAloj / 100)));
            custoDeUT := (((Salario / 220) * 2.1) + (0.184 * codColhedora[indColhedora[c],3]) * Diesel +
            (((0.00059 * codColhedora[indColhedora[c],3]) + 0.02169) * (Diesel * 4.5)) + (0.4 *
            codColhedora[indColhedora[c],5]) / (VidaUtil * detAno[anoAT - priAno,2])) * detAno[anoAT - priAno,2];
            custoDeSM := (custoDePR + custoDeUT) * codColhedora[indColhedora[c],2];
            custoUnitario := custoDeSM * (resultPropAno[t - 1,(anoAT - priAno),6] / (areaTotalProp *
            detAno[anoAT - priAno,7])) + (resultPropAno[t - 1,(anoAT - priAno),0] * PrecSaco * (capColhedora[c] /
            capInstal));
            custoUnitario := custoUnitario / (resultPropAno[t - 1,(anoAT - priAno),1] * (capColhedora[c] /
            capInstal));
            RowCount := RowCount + 1;
            Cells[0,RowCount - 1] := 'Talhão ' + letra[t];
            Cells[1,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,(anoAT - priAno),6] *
            (capColhedora[c] / capInstal)]);
            Cells[2,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,(anoAT - priAno),0] * PrecSaco *
            (capColhedora[c] / capInstal)]);
            Cells[3,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,(anoAT - priAno),1] * PrecSaco *
            (capColhedora[c] / capInstal)]);
            Cells[4,RowCount - 1] := Format ('%n', [custoDeSM * (resultPropAno[t - 1,(anoAT - priAno),6] /
            (areaTotalProp * detAno[anoAT - priAno,7]))]);
            Cells[5,RowCount - 1] := Format ('%n', [custoUnitario]);
            colAno[anoAT - priAno,c] := colAno[anoAT - priAno,c] + custoUnitario;
            resultMaqAno[t - 1,(anoAT - priAno),c,0] := custoDeSM * (resultPropAno[t - 1,(anoAT -
            priAno),6] / (areaTotalProp * detAno[anoAT - priAno,7]));
            resultMaqAno[t - 1,(anoAT - priAno),c,1] := custoUnitario;
            resultMaqAno[t - 1,(anoAT - priAno),c,2] := resultPropAno[t - 1,(anoAT - priAno),6] *
            (capColhedora[c] / capInstal);
          end;
          RowCount := RowCount + 5;
        end;
      end;
    end;
  end;
end;

```

```

    detAno[anoAT - priAno,5] := detAno[anoAT - priAno,5] + custoDeSM;
    colAno[anoAT - priAno,c] := colAno[anoAT - priAno,c] / (StgPropTal.RowCount - 1);
    detAno[anoAT - priAno,6] := detAno[anoAT - priAno,6] + colAno[anoAT - priAno,c];
    Cells[0,RowCount - 5] := 'SERVIÇO MECANIZADO (US$)';Cells[1,RowCount - 5] := Format
('%n', [custoDeSM]);
    Cells[0,RowCount - 4] := 'CUSTO M. COLHEDORA (US$/sc)';Cells[1,RowCount - 4] := Format
('%n', [colAno[anoAT - priAno,c]]);
    end;
    detAno[anoAT - priAno,5] := PrecSaco * (detAno[anoAT - priAno,4] - detAno[anoAT - priAno,3]) -
detAno[anoAT - priAno,5];
    detAno[anoAT - priAno,6] := detAno[anoAT - priAno,6] / numCol;
    StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 2] := 'RENTA LÍQUIDA
(US$)';StgCustoMaq.Cells[1,StgCustoMaq.RowCount - 2] := Format ('%n', [detAno[anoAT -
priAno,5]]);
    StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 1] := 'CUSTO M. FROTA
(US$/sc)';StgCustoMaq.Cells[1,StgCustoMaq.RowCount - 1] := Format ('%n', [detAno[anoAT -
priAno,6]]);
    StgCustoMaq.RowCount := StgCustoMaq.RowCount + 2;
    end
    else
        StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 3] :=
StgCustoMaq.Cells[0,StgCustoMaq.RowCount - 3] + ' (inválido)';
        end;
        StgCustoMaq.RowCount := StgCustoMaq.RowCount - 2;
    end;

procedure TFrmPropAlterar.ResultadoFinal(Sender: TObject);
var a, i, ind, ACol: Integer;
    valoresVar: array[0..3] of array of Real;
    valoresTal: array of array[0..1] of array of Real;
begin
    SetLength (valoresTal, StgPropTal.RowCount - 1);
    ind := 0;
    for a := 0 to High (detAno) do
        if (detAno[a,1] = 1) then
            begin
                for i := 0 to 3 do
                    SetLength (valoresVar[i], ind + 1);
                    valoresVar[0,ind] := (detAno[a,3] * PrecSaco) / areaTotalProp;
                    valoresVar[1,ind] := detAno[a,5] / areaTotalProp;
                    valoresVar[2,ind] := (detAno[a,4] * PrecSaco - detAno[a,5]) / areaTotalProp;
                    valoresVar[3,ind] := detAno[a,6];
                    for i := 1 to (StgPropTal.RowCount - 1) do
                        begin
                            SetLength (valoresTal[i - 1,0], ind + 1);
                            SetLength (valoresTal[i - 1,1], ind + 1);
                            valoresTal[i - 1,0,ind] := resultPropAno[i - 1,a,4] + 1;
                            valoresTal[i - 1,1,ind] := resultPropAno[i - 1,a,5];
                        end;
                        ind := ind + 1;
                    end;
                with StgResultFinalEst do
                    begin
                        Cells[0,0] := 'Nº de anos da série';Cells[1,0] := IntToStr ((ultAno - priAno) + 1);
                        Cells[0,1] := 'Nº de anos válidos';Cells[1,1] := IntToStr (ind);
                        if (ind > 0) then

```

```

begin
  RowCount := RowCount + 3;
  Cells[0, RowCount - 2] := 'ESTATÍSTICA';
  for ACol := 0 to 4 do
    Cells[ACol, RowCount - 1] := atributo[ACol];
    for i := 0 to 3 do
      begin
        RowCount := RowCount + 1;
        CalcularMedias (0, i, valoresVar[i]);
        Cells[0, RowCount - 1] := variavel[i];
        for ACol := 1 to 4 do
          Cells[ACol, RowCount - 1] := Format ('%n', [resultFinalVar[i, ACol - 1]]);
        end;
        RowCount := RowCount + 3;
        Cells[0, RowCount - 2] := 'DETALHES DO TALHÃO';
        Cells[0, RowCount - 1] := 'Prazo (dias)';
        for i := 1 to 4 do
          begin
            RowCount := RowCount + 1;
            Cells[0, RowCount - 1] := atributo[i];
          end;
          ind := RowCount;
          RowCount := RowCount + 1;
          Cells[0, RowCount - 1] := 'Nº de dias úteis';
          for i := 1 to 4 do
            begin
              RowCount := RowCount + 1;
              Cells[0, RowCount - 1] := atributo[i];
            end;
            for ACol := 1 to (StgPropTal.RowCount - 1) do
              begin
                if (ACol > (ColCount - 1)) then
                  ColCount := ColCount + 1;
                  Cells[ACol, ind - 6] := 'Talhão ' + letra[ACol];
                  CalcularMedias (1, ACol - 1, valoresTal[ACol - 1, 0]);
                  CalcularMedias (2, ACol - 1, valoresTal[ACol - 1, 1]);
                  for i := 0 to 3 do
                    begin
                      Cells[ACol, ind - (4 - i)] := Format ('%n', [resultFinalTal[ACol - 1, 0, i]]);
                      Cells[ACol, RowCount - (4 - i)] := Format ('%n', [resultFinalTal[ACol - 1, 1, i]]);
                    end;
                  end;
                end;
              end;
            else
              raise Exception.Create ('Não foi possível alterar a propriedade porque não há nenhum ano na
série temporal');
            end;
          end;
        end;

function TFrmPropAlterar.SalvarPropriedade: Boolean;
var t, anoAT, c: Integer;
    data: TDateTime;
begin
  data := FrmSimul.QrySimul.FieldValues['DataDeCriação'];
  FrmSimul.ExcluirSimulacao (codProp);
  FrmSimul.QryProp.Delete;

```

```

FrmSimul.QryProp.InsertRecord ([codProp,FrmReg.TblReg.FieldValues['CódRegião']]);
FrmSimul.QrySimul.InsertRecord ([codProp,data,turnoTrab,precCrit,StgPropTal.RowCount - 1]);
FrmSimul.QrySimulInd.InsertRecord ([codProp,PrecSaco,Juros,VidaUtil,Eficiencia,Salario,
    SegAloj,Diesel,Residual]);
for c := 0 to (numCol - 1) do
    FrmSimul.QrySimulCol.InsertRecord
([codProp,c,codColhedora[indColhedora[c],0],codColhedora[indColhedora[c],1]);
    for t := 1 to (StgPropTal.RowCount - 1) do
        begin
            FrmSimul.QrySimulTal.InsertRecord
([codProp,t,talCultivar[t,1],StgPropTal.Cells[5,t],StgPropTal.Cells[7,t],Null,StgPropTal.Cells[3,t]);
            for anoAT := priAno to ultAno do
                if (detAno[anoAT - priAno,1] = 1) then
                    begin
                        FrmSimul.QryResultPropAno.InsertRecord ([codProp,t,anoAT,StrToDate (StgPropTal.Cells[7,t]) +
resultPropAno[t - 1,(anoAT - priAno),2],
                            StrToDate (StgPropTal.Cells[7,t]) + resultPropAno[t - 1,(anoAT -
priAno),2] + resultPropAno[t - 1,(anoAT - priAno),4],
                            resultPropAno[t - 1,(anoAT - priAno),0],resultPropAno[t - 1,(anoAT -
priAno),1],resultPropAno[t - 1,(anoAT - priAno),4],
                            resultPropAno[t - 1,(anoAT - priAno),5]]);
                        for c := 0 to (numCol - 1) do
                            FrmSimul.QryResultMaqAno.InsertRecord
([codProp,t,anoAT,c,codColhedora[indColhedora[c],2],Null,resultMaqAno[t - 1,(anoAT - priAno),c,0],
                                Null,resultMaqAno[t - 1,(anoAT - priAno),c,1],resultMaqAno[t - 1,(anoAT
- priAno),c,2]]);
                            end;
                        end;
                        SalvarPropriedade := true;
                        FrmSimul.QryProp.Refresh;
                        MessageDlg ('As alterações da propriedade foram salvas com êxito.', MTInformation, [MbOK], 0);
                    end;
                end;
            end;
        end;
    end;

procedure TFrmPropAlterar.OpSalvarClick(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    try
        if Not (propAlterado) then
            raise Exception.Create ('A propriedade não foi alterada')
        else if (propSalvado) then
            raise Exception.Create ('As alterações da propriedade já foram salvas')
        else
            propSalvado := SalvarPropriedade;
        finally
            Screen.Cursor := crDefault;
        end;
    end;
end;

procedure TFrmPropAlterar.OpIndicadorClick(Sender: TObject);
begin
    try
        FrmIndicador := TFrmIndicador.Create (Application);
        FrmIndicador.ShowModal;
    finally
        FrmIndicador.Free;
    end;
end;

```

```

end;

procedure TFrmPropAlterar.FormClose(Sender: TObject; var Action: TCloseAction);
var resp: Word;
begin
  Screen.Cursor := crHourglass;
  try
    if (propAlterado and Not (propSalvado)) then
      begin
        resp := MessageDlg ('Deseja salvar as alterações da propriedade?', MTWarning,
[ MbYes, MbNo, MbCancel ], 0);
        if (resp = mrYes) then
          propSalvado := SalvarPropriedade
        else if (resp = mrCancel) then
          Action := caNone;
        end;
      finally
        Screen.Cursor := crDefault;
      end;
    end;
end;

procedure TFrmPropAlterar.BtnCancelarClick(Sender: TObject);
begin
  Close;
end;

function TFrmPropAlterar.LocalizarAno(anoAT, indGrad: Integer): Boolean;
var a, ARow: Integer;
    valido, achou: Boolean;
begin
  valido := true;
  a := anoAT - priAno;
  if ((anoAT < priAno) or (anoAT > ultAno)) then
    valido := false
  else if Not (anoAT = detAno[a,0]) then
    begin
      a := 0;
      achou := false;
      repeat
        if (anoAT = detAno[a,0]) then
          achou := true
        else
          a := a + 1;
      until (achou or (a > High (detAno)));
      valido := achou;
    end;
  if valido then
    case indGrad of
      0: with StgCustoOpt do
        begin
          ARow := RowCount - (linAno[a,0] + 1);
          if (ARow >= 20) then
            ARow := 20;
          Col := 0;
          Row := linAno[a,0] + ARow;
          Row := Row - ARow;
        end;
    end;
end;

```

```

    end;
1: with StgCustoMaq do
begin
    ARow := RowCount - (linAno[a,1] + 1);
    if (ARow >= 20) then
        ARow := 20;
    Col := 0;
    Row := linAno[a,1] + ARow;
    Row := Row - ARow;
end;
end;
LocalizarAno := valido;
end;

procedure TFrmPropAlterar.OpLocalizarAno1Click(Sender: TObject);
var strAno: String;
begin
    try
        strAno := '';
        if InputQuery ('Série temporal - localizar ano', 'Ano (aaaa)', strAno) then
            if Not (LocalizarAno (StrToInt (strAno), 0)) then
                MessageDlg ('O ano especificado não existe.', MTInformation, [MbOK], 0);
    except
        on EConvertError do
            raise Exception.Create ('O valor fornecido é inválido');
    end;
end;

procedure TFrmPropAlterar.OpLocalizarAno2Click(Sender: TObject);
var strAno: String;
begin
    try
        strAno := '';
        if InputQuery ('Série temporal - localizar ano', 'Ano (aaaa)', strAno) then
            if Not (LocalizarAno (StrToInt (strAno), 1)) then
                MessageDlg ('O ano especificado não existe.', MTInformation, [MbOK], 0);
    except
        on EConvertError do
            raise Exception.Create ('O valor fornecido é inválido');
    end;
end;

function TFrmPropAlterar.CaixaDeEntrada(cs: Integer; var str1, str2: String): Boolean;
begin
    CaixaDeEntrada := false;
    FrmCaixaDeEntrada := TFrmCaixaDeEntrada.Create (Application);
    try
        with FrmCaixaDeEntrada do
            begin
                Caption := ChkListBoxPropCol.Items.Strings[cs];
                TxtVeloc.Text := str1;
                TxtNumCol.Text := str2;
                if (ShowModal = mrOK) then
                    begin
                        CaixaDeEntrada := true;
                        str1 := TxtVeloc.Text;

```

```

        str2 := TxtNumCol.Text;
    end;
end;
finally
    FrmCaixaDeEntrada.Free;
end;
end;

```

```

procedure TFrmPropAlterar.ChkListBoxPropColMouseDown(Sender: TObject;
    Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var APoint: TPoint;
begin
    APoint.X := X;
    APoint.Y := Y;
    indCol := ChkListBoxPropCol.ItemAtPos (APoint, true);
end;

```

```

procedure TFrmPropAlterar.ChkListBoxPropColKeyDown(Sender: TObject;
    var Key: Word; Shift: TShiftState);
begin
    indCol := ChkListBoxPropCol.ItemIndex;
    if (indCol < 0) then
        indCol := 0;
end;

```

```

{procedure TFrmPropAlterar.DistribuirValores(indVar: Integer; valor: array of Real);
var i, ind, freq: Integer;
    vAT: Real;
    ordenou: Boolean;
begin
    ind := 0;
    repeat
        ind := ind + 1;
        ordenou := true;
        for i := High (valor) downto ind do
            if (valor[i] > valor[i - 1]) then
                begin
                    vAT := valor[i - 1];
                    valor[i - 1] := valor[i];
                    valor[i] := vAT;
                    ordenou := false;
                end;
            until ordenou or (ind = High (valor));
            ind := -1;
        for i := 0 to High (valor) do
            if ((i > 0) and (valor[i] = valor[i - 1])) then
                categoria[indVar,ind,1] := categoria[indVar,ind,1] + 1
            else
                begin
                    ind := ind + 1;
                    SetLength (categoria[indVar], ind + 1);
                    categoria[indVar,ind,0] := valor[i];
                    categoria[indVar,ind,1] := 1;
                end;
            freq := 0;
        for ind := 0 to High (categoria[indVar]) do

```

```

begin
    freq := freq + Trunc (categoria[indVar,ind,1]);
    categoria[indVar,ind,2] := (freq / (High (valor) + 2)) * 100;
end;
end;}

procedure TFrmPropAlterar.LimparPropriedade(opCab: Integer);
var ARow: Integer;
begin
    propAlterado := false;
    propSalvado := false;
    for ARow := (StgCustoOpt.RowCount - 1) downto 0 do
        StgCustoOpt.Rows[ARow].Clear;
    StgCustoOpt.RowCount := 2;
    for ARow := (StgCustoMaq.RowCount - 1) downto 1 do
        StgCustoMaq.Rows[ARow].Clear;
    StgCustoMaq.RowCount := 2;
    for ARow := (StgResultFinalEst.RowCount - 1) downto 0 do
        StgResultFinalEst.Rows[ARow].Clear;
    StgResultFinalEst.ColCount := 5;
    StgResultFinalEst.RowCount := 2;
    with StgCustoOpt do
        case opCab of
            0: begin
                ColCount := 7;
                Cells[0,0] := 'Ano/Talhão';ColWidths[0] := 148;
                Cells[1,0] := 'Início (dd/mm/aa)';ColWidths[1] := 93;
                Cells[2,0] := 'Término (dd/mm/aa)';ColWidths[2] := 104;
                Cells[3,0] := 'Atraso (dias)';ColWidths[3] := 65;
                Cells[4,0] := 'Nº de dias úteis';ColWidths[4] := 81;
                Cells[5,0] := 'Perda total (sc)';ColWidths[5] := 78;
                Cells[6,0] := 'Renda total (sc)';ColWidths[6] := 82;
            end;
            1: begin
                ColCount := 9;
                Cells[0,0] := 'Data (dd/mm/aa)';ColWidths[0] := 148;
                Cells[1,0] := 'Precipitação (mm)';ColWidths[1] := 120;
                Cells[2,0] := 'Situação';ColWidths[2] := 49;
                Cells[3,0] := 'Atraso (dias)';ColWidths[3] := 65;
                Cells[4,0] := 'Perda física diária (sc)';ColWidths[4] := 112;
                Cells[5,0] := 'Perda física acumulada (sc)';ColWidths[5] := 139;
                Cells[6,0] := 'Renda física diária (sc)';ColWidths[6] := 116;
                Cells[7,0] := 'Renda física acumulada (sc)';ColWidths[7] := 143;
                Cells[8,0] := 'Área colhida (ha)';ColWidths[8] := 87;
            end;
        end;
    end;
end;

procedure TFrmPropAlterar.RecuperarPropriedade(Sender: TObject);
var t, a, c, numTal, numAnos, aln: Integer;
    capInstal, custoDeSM: Real;
    strData: String[8];
    capColhedora: array of Real;
begin
    codProp := FrmSimul.QryProp.FieldValues['CódPropriedade'];
    FrmReg.TblReg.FindKey ([FrmSimul.QryProp.FieldValues['CódRegião']]);

```

```

LblCod.Caption := codProp;
TxtReg.Text := FrmReg.TblReg.FieldValues['Sigla'];
with FrmSimul.QrySimul do
begin
  Close;
  SQL.Clear;
  SQL.Add ('select *');
  SQL.Add ('from "SIMULAÇÃO"');
  SQL.Add ('where CódSimulação like "' + codProp + '"');
  Open;
  turnoTrab := FieldValues['TurnoDeTrabalho'];
  precCrit := FieldValues['PrecipitaçãoCrítica'];
  TxtTurno.Text := FieldValues['TurnoDeTrabalho'];
  TxtPrec.Text := FieldValues['PrecipitaçãoCrítica'];
end;
with FrmSimul.QrySimulInd do
begin
  PrecSaco := FieldValues['PreçoDoSaco'];
  Juros := FieldValues['TaxaDeJuros'];
  VidaUtil := FieldValues['VidaÚtil'];
  Eficiencia := FieldValues['Eficiência'];
  Salario := FieldValues['Salário'];
  SegAloj := FieldValues['TaxaDeSeguroAlojamento'];
  Diesel := FieldValues['PreçoDoDiesel'];
  Residual := FieldValues['ValorResidual'];
end;
with FrmSimul.QrySimulCol do
begin
  Close;
  SQL.Clear;
  SQL.Add ('select *');
  SQL.Add ('from "SIMULAÇÃO_Colhedora"');
  SQL.Add ('where CódSimulação like "' + codProp + '"');
  Open;
  First;
  numCol := RecordCount;
end;
ListarColhedoras (self);
with FrmSimul.QrySimulTal do
begin
  Close;
  SQL.Clear;
  SQL.Add ('select *');
  SQL.Add ('from "SIMULAÇÃO_Talhão"');
  SQL.Add ('where CódSimulação like "' + codProp + '"');
  Open;
  First;
  numTal := RecordCount;
  areaTotalProp := 0;
end;
LimparListaTal (self);
with FrmSimul.QryResultPropAno do
begin
  Close;
  SQL.Clear;
  SQL.Add ('select *');

```

```

SQL.Add ('from "RESULTADO_PropAno"');
SQL.Add ('where CódSimulação like "' + codProp + '"');
Open;
First;
numAnos := RecordCount div numTal;
end;
LimparPropriedade (0);
with FrmSimul.QryResultMaqAno do
begin
Close;
SQL.Clear;
SQL.Add ('select *');
SQL.Add ('from "RESULTADO_MaqAno"');
SQL.Add ('where CódSimulação like "' + codProp + '"');
Open;
First;
end;
SetLength (linAno, numAnos);
SetLength (detAno, numAnos);
SetLength (colAno, numAnos);
SetLength (resultPropAno, numTal);
SetLength (resultMaqAno, numTal);
SetLength (resultFinalTal, numTal);
SetLength (capColhedora, numCol);
capInstal := 0;
for t := 1 to numTal do
begin
SetLength (resultPropAno[t - 1], numAnos);
SetLength (resultMaqAno[t - 1], numAnos);
RecuperarPropTal (FrmReg.TblReg.FieldValues['CódRegião'],
FrmSimul.QrySimulTal.FieldValues['CódCultivar']);
for a := 0 to (numAnos - 1) do
begin
SetLength (resultMaqAno[t - 1, a], numCol);
with FrmSimul.QryResultPropAno do
begin
if ((t - 1) = 0) then
begin
SetLength (colAno[a], numCol);
detAno[a, 0] := FieldValues['Ano'];
detAno[a, 1] := 1;
detAno[a, 2] := 0;
detAno[a, 3] := 0;
detAno[a, 4] := 0;
detAno[a, 5] := 0;
detAno[a, 6] := 0;
end;
resultPropAno[t - 1, a, 0] := FieldValues['PerdaAcumulada'];
resultPropAno[t - 1, a, 1] := FieldValues['ColheitaAcumulada'];
if (talCultivar[t, 1] <> 0) then
begin
strData := StgPropTal.Cells[7, t];
System.Insert (Copy (FloatToStr (detAno[a, 0]), 3, 4), strData, 7);
resultPropAno[t - 1, a, 2] := StrToDate (FormatDateTime ('dd/mm/' + Copy (FloatToStr
(detAno[a, 0]), 3, 4), FieldValues['DataDeInícioDaColheita'])) - StrToDate (strData);

```

```

    aln := Trunc ((StrToDate (StgPropTal.Cells[7,t]) + resultPropAno[t - 1,a,2]) - (StrToDate
(StgPropTal.Cells[6,t]) + 3));
    if (aln < 0) then
        aln := 0;
    end;
    resultPropAno[t - 1,a,3] := aln;
    resultPropAno[t - 1,a,4] := FieldValues['PrazoDeColheita'];
    resultPropAno[t - 1,a,5] := FieldValues['NúmeroDeDiasÚteis'];
    Next;
end;
for c := 0 to (numCol - 1) do
    with FrmSimul.QryResultMaqAno do
        begin
            resultMaqAno[t - 1,a,c,0] := FieldValues['CustoProdutivo'];
            resultMaqAno[t - 1,a,c,1] := FieldValues['CustoUnitário'];
            resultMaqAno[t - 1,a,c,2] := FieldValues['ÁreaColhida'];
            if ((t - 1) = 0) then
                begin
                    colAno[a,c] := 0;
                    if (a = 0) then
                        begin
                            codColhedora[indColhedora[c],2] := FieldValues['NúmeroDeColhedoras'];
                            capColhedora[c] := (codColhedora[indColhedora[c],1] * codColhedora[indColhedora[c],2] *
codColhedora[indColhedora[c],4] * (Eficiencia / 1000)) * turnoTrab;
                            capInstal := capInstal + capColhedora[c];
                        end;
                    end;
                    Next;
                end;
            end;
            FrmSimul.QrySimulTal.Next;
        end;
    StgPropTal.RowCount := StgPropTal.RowCount - 1;
    for a := 0 to (numAnos - 1) do
        begin
            with StgCustoOpt do
                begin
                    linAno[a,0] := RowCount - 1;
                    Cells[0,RowCount - 1] := 'ANO - ' + FloatToStr (detAno[a,0]);
                    RowCount := RowCount + 2;
                    for t := 1 to numTal do
                        begin
                            Cells[0,RowCount - 1] := 'Talhão ' + letra[t];
                            if (talCultivar[t,1] <> 0) then
                                begin
                                    Cells[1,RowCount - 1] := FormatDateTime ('dd/mm/' + Copy (FloatToStr (detAno[a,0]), 3, 2),
StrToDate (StgPropTal.Cells[7,t]) + resultPropAno[t - 1,a,2]);
                                    Cells[2,RowCount - 1] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[1,RowCount - 1]) +
resultPropAno[t - 1,a,4]);
                                end
                            else
                                begin
                                    Cells[1,RowCount - 1] := 'Não encontrado';
                                    Cells[2,RowCount - 1] := 'Não encontrado';
                                end;
                            Cells[3,RowCount - 1] := FloatToStr (resultPropAno[t - 1,a,3] + resultPropAno[t - 1,a,4]);
                        end
                    end;
                end;
            end;
        end;
    end;
end;

```

```

Cells[4,RowCount - 1] := FloatToStr (resultPropAno[t - 1,a,5]);
Cells[5,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,a,0]]);
Cells[6,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,a,1]]);
detAno[a,2] := detAno[a,2] + resultPropAno[t - 1,a,5] * turnoTrab;
detAno[a,3] := detAno[a,3] + resultPropAno[t - 1,a,0];
detAno[a,4] := detAno[a,4] + resultPropAno[t - 1,a,1];
RowCount := RowCount + 1;
end;
RowCount := RowCount + 5;
Cells[0,RowCount - 6] := 'RESULTADOS';
Cells[0,RowCount - 5] := 'Perda total (US$):';Cells[1,RowCount - 5] := Format ('%n', [detAno[a,3] *
PrecSaco]);
Cells[0,RowCount - 4] := 'Renda total (US$):';Cells[1,RowCount - 4] := Format ('%n', [detAno[a,4] *
PrecSaco]);
Cells[0,RowCount - 3] := 'Tempo de utilização anual (h):';Cells[1,RowCount - 3] := Format ('%n',
[detAno[a,2]]);
end;
with StgCustoMq do
begin
linAno[a,1] := RowCount - 1;
Cells[0,RowCount - 1] := 'ANO - ' + FloatToStr (detAno[a,0]);
RowCount := RowCount + 2;
for c := 0 to (numCol - 1) do
begin
RowCount := RowCount + 5;
Cells[0,RowCount - 7] := 'Marca: ' + modColhedora[indColhedora[c],0];
Cells[0,RowCount - 6] := 'Modelo: ' + modColhedora[indColhedora[c],1];
Cells[0,RowCount - 5] := 'Plataforma: ' + Format ('%n m', [codColhedora[indColhedora[c],3]]);
Cells[0,RowCount - 4] := 'Velocidade: ' + Format ('%n km/h', [codColhedora[indColhedora[c],1]]);
Cells[0,RowCount - 3] := 'Número de colhedoras: ' + FloatToStr
(codColhedora[indColhedora[c],2]);
Cells[0,RowCount - 2] := 'Capacidade: ' + Format ('%n ha/dia', [capColhedora[c]]);
Cells[0,RowCount - 1] := 'RESULTADOS';
custoDeSM := 0;
for t := 1 to numTal do
begin
RowCount := RowCount + 1;
Cells[0,RowCount - 1] := 'Talhão ' + letra[t];
Cells[1,RowCount - 1] := Format ('%n', [resultMqAno[t - 1,a,c,2]]);
Cells[2,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,a,0] * PrecSaco * (capColhedora[c] /
capInstal)]);
Cells[3,RowCount - 1] := Format ('%n', [resultPropAno[t - 1,a,1] * PrecSaco * (capColhedora[c] /
capInstal)]);
Cells[4,RowCount - 1] := Format ('%n', [resultMqAno[t - 1,a,c,0]]);
Cells[5,RowCount - 1] := Format ('%n', [resultMqAno[t - 1,a,c,1]]);
custoDeSM := custoDeSM + resultMqAno[t - 1,a,c,0];
colAno[a,c] := colAno[a,c] + resultMqAno[t - 1,a,c,1];
end;
RowCount := RowCount + 5;
detAno[a,5] := detAno[a,5] + custoDeSM;
colAno[a,c] := colAno[a,c] / numTal;
detAno[a,6] := detAno[a,6] + colAno[a,c];
Cells[0,RowCount - 5] := 'SERVIÇO MECANIZADO (US$):';Cells[1,RowCount - 5] := Format
('%n', [custoDeSM]);
Cells[0,RowCount - 4] := 'CUSTO M. COLHEDORA (US$/sc):';Cells[1,RowCount - 4] := Format
('%n', [colAno[a,c]]);

```

```

end;
detAno[a,5] := PrecSaco * (detAno[a,4] - detAno[a,3]) - detAno[a,5];
detAno[a,6] := detAno[a,6] / numCol;
Cells[0,RowCount - 2] := 'REND A LÍQUIDA (US$)';Cells[1,RowCount - 2] := Format ('%n',
[detAno[a,5]]);
Cells[0,RowCount - 1] := 'CUSTO M. FROTA (US$)';Cells[1,RowCount - 1] := Format ('%n',
[detAno[a,6]]);
RowCount := RowCount + 2;
end;
end;
StgCustoOpt.RowCount := StgCustoOpt.RowCount - 2;
StgCustoMaq.RowCount := StgCustoMaq.RowCount - 2;
priAno := Trunc (detAno[0,0]);
ultAno := Trunc (detAno[numAnos - 1,0]);
ListarAnos (FrmReg.TblReg.FieldValues['CódRegião']);
PgCtrlPropAlterar.ActivePageIndex := 0;
ResultadoFinal (self);
end;

procedure TFrmPropAlterar.RecuperarPropTal(codReg, codCul: Integer);
var ACol, aln: Integer;
begin
FrmCultivar.TblCul.FindKey ([codCul]);
with FrmCultivar.QryRegCul do
begin
Close;
SQL.Clear;
SQL.Add ('select *');
SQL.Add ('from "REGIÃO_CULTIVAR"');
SQL.Add ('where CódRegião = ' + IntToStr (codReg));
SQL.Add ('and CódCultivar = ' + IntToStr (codCul));
Open;
FindFirst;
end;
with StgPropTal do
begin
Cells[0,RowCount - 1] := letra[RowCount - 1];
Cells[1,RowCount - 1] := FrmCultivar.TblCul.FieldValues['NomeCultivar'];
Cells[2,RowCount - 1] := FrmCultivar.TblCul.FieldValues['GrupoDeMaturação'];
Cells[3,RowCount - 1] := FrmSimul.QrySimulTal.FieldValues['ÁreaDeLavoura'];
areaTotalProp := areaTotalProp + FrmSimul.QrySimulTal.FieldValues['ÁreaDeLavoura'];
talCultivar[RowCount - 1,0] := 0;
if Not (FrmCultivar.QryRegCul.Found) then
begin
for ACol := 4 to 8 do
Cells[ACol,RowCount - 1] := 'Não encontrado';
MessageDlg ('A função de perdas à qual o talhão ' + letra[RowCount - 1] + ' está vinculado foi
removida.',
MTWarning, [MbOK], 0);
talCultivar[RowCount - 1,1] := 0;
end
else
begin
Cells[4,RowCount - 1] := FrmCultivar.QryRegCul.FieldValues['CicloDeMaturação'];
Cells[5,RowCount - 1] := FormatDateTime ('dd/mm/' + Copy (IntToStr (StrToInt (FormatDateTime
('yyyy', Date)) - 1), 3, 2),

```

```

FrmSimul.QrySimulTal.FieldValues['DataDeSemeadura']);
Cells[6,RowCount - 1] := FormatDateTime ('dd/mm/yy', StrToDate (Cells[5,RowCount - 1]) +
StrToInt (Cells[4,RowCount - 1]));
Cells[7,RowCount - 1] := FormatDateTime ('dd/mm/' + FormatDateTime ('yy', Date),
FrmSimul.QrySimulTal.FieldValues['DataDeInícioDaColheita']);
aln := Trunc (StrToDate (Cells[7,RowCount - 1]) - (StrToDate (Cells[6,RowCount - 1]) + 3));
if (aln < 1) then
    aln := 0;
Cells[8,RowCount - 1] := IntToStr (aln);
talCultivar[RowCount - 1,1] := codCul;
talCultivar[RowCount - 1,2] := FrmCultivar.QryRegCul.FieldValues['CoeficienteDePerdA'];
talCultivar[RowCount - 1,3] := FrmCultivar.QryRegCul.FieldValues['CoeficienteDePerdB'];
talCultivar[RowCount - 1,4] := FrmCultivar.QryRegCul.FieldValues['CoeficienteDePerdC'];
talCultivar[RowCount - 1,5] := FrmCultivar.QryRegCul.FieldValues['CoeficienteDeProdC'];
end;
RowCount := RowCount + 1;
end;
end;
end;

```

```

procedure TFrmPropAlterar.ListarAnos(codReg: Integer);
begin
    CboPriAno.Items.Clear;
    CboUltAno.Items.Clear;
    with FrmSimul.QryPrec do
    begin
        Close;
        SQL.Clear;
        SQL.Add ('select distinct extract (year from Data) as Ano');
        SQL.Add ('from "REGIÃO_Precipitação"');
        SQL.Add ('where CódRegião = ' + IntToStr (codReg));
        Open;
        First;
    end;
    repeat
        CboPriAno.Items.Add (FrmSimul.QryPrec.FieldValues['Ano']);
        FrmSimul.QryPrec.Next;
    until FrmSimul.QryPrec.Eof;
    CboUltAno.Items := CboPriAno.Items;
    CboPriAno.ItemIndex := priAno - StrToInt (CboPriAno.Items[0]);
    CboUltAno.ItemIndex := ultAno - StrToInt (CboUltAno.Items[0]);
    CboPriAno.Text := IntToStr (priAno);
    CboUltAno.Text := IntToStr (ultAno);
end;

```

```

procedure TFrmPropAlterar.CalcularMedias(opResult, indVar: Integer; valor: array of Real);
var i: Integer;
    min, max, media, desvio: Real;
begin
    min := valor[0];
    max := valor[0];
    media := 0;
    for i := 0 to High (valor) do
    begin
        if (min > valor[i]) then
            min := valor[i]
        else if (max < valor[i]) then

```

```

    max := valor[i];
    media := media + valor[i];
end;
media := media / (High (valor) + 1);
desvio := 0;
for i := 0 to High (valor) do
    desvio := desvio + Sqr (valor[i] - media);
desvio := Sqrt (desvio / High (valor));
case opResult of
    0: begin
        resultFinalVar[indVar,0] := min;
        resultFinalVar[indVar,1] := max;
        resultFinalVar[indVar,2] := media;
        resultFinalVar[indVar,3] := desvio;
    end;
    1, 2: begin
        resultFinalTal[indVar,opResult - 1,0] := min;
        resultFinalTal[indVar,opResult - 1,1] := max;
        resultFinalTal[indVar,opResult - 1,2] := media;
        resultFinalTal[indVar,opResult - 1,3] := desvio;
    end;
end;
end;

procedure TFrmPropAlterar.OpExportarClick(Sender: TObject);
var t, ind, i: Integer;
    achou: Boolean;
begin
    if Not (StgResultFinalEst.RowCount > 2) then
        raise Exception.Create ('Não há nenhuma informação para exportar');
    try
        FrmExpExcel := TFrmExpExcel.Create (Application);
        with FrmExpExcel do
            begin
                SetLength (indCultivar, 1);
                SetLength (indColhedora, numCol);
                opSimul := 0;
                codSimul := codProp;
                areaTotalSimul := areaTotalProp;
                stgVirtual := StgResultFinalEst;
                ind := 0;
                for t := 1 to (StgPropTal.RowCount - 1) do
                    if (t > 1) then
                        begin
                            achou := false;
                            i := 0;
                            repeat
                                if (talCultivar[t,1] = indCultivar[i,0]) then
                                    achou := true
                                else
                                    i := i + 1;
                            until (achou or (i > ind));
                            if achou then
                                begin
                                    indCultivar[i,1] := indCultivar[i,1] + StrToFloat (StgPropTal.Cells[3,t]);
                                    indCultivar[i,2] := indCultivar[i,2] + 1;
                                end;
                            end;
                        end;
                end;
            end;
        end;
    except
        ;
    end;
end;

```

```

    end
  else
  begin
    ind := ind + 1;
    SetLength (indCultivar, ind + 1);
    indCultivar[ind,0] := talCultivar[t,1];
    indCultivar[ind,1] := StrToFloat (StgPropTal.Cells[3,t]);
    indCultivar[ind,2] := 1;
  end;
end
else
begin
  indCultivar[0,0] := talCultivar[1,1];
  indCultivar[0,1] := StrToFloat (StgPropTal.Cells[3,1]);
  indCultivar[0,2] := 1;
end;
caplInstal := 0;
for i := 0 to High (indColhedora) do
begin
  indColhedora[i,0] := codColhedora[FrmPropAlterar.indColhedora[i],4];
  indColhedora[i,1] := codColhedora[FrmPropAlterar.indColhedora[i],3];
  indColhedora[i,2] := codColhedora[FrmPropAlterar.indColhedora[i],2];
  caplInstal := caplInstal + (codColhedora[FrmPropAlterar.indColhedora[i],1] *
codColhedora[FrmPropAlterar.indColhedora[i],2] * codColhedora[FrmPropAlterar.indColhedora[i],4] *
(Eficiencia / 1000)) * turnoTrab;
end;
  ShowModal;
end;
finally
  FrmExpExcel.Free;
end;
end;

end.

```

F - Módulo 10: Módulo Configuração dos Indicadores (Arquivo UIndicador.pas)

G - Código Fonte 18: UIndicador.pas <<código>>

```

unit UIndicador;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls, ComCtrls;

type
  TFrmIndicador = class(TForm)
    Lbl1: TLabel;
    GrpBoxSac: TGroupBox;
    Lbl2: TLabel;
    TxtPrecSaco: TEdit;

```

```

GrpBoxMq: TGroupBox;
Lbl3: TLabel;
Lbl4: TLabel;
Lbl5: TLabel;
Lbl6: TLabel;
Lbl7: TLabel;
TxtJuros: TEdit;
TxtVidaUtil: TEdit;
TxtEficiencia: TEdit;
TxtSalario: TEdit;
TxtSegAloj: TEdit;
Lbl8: TLabel;
TxtDiesel: TEdit;
StatusBarInd: TStatusBar;
Pnl1: TPanel;
BtnConfirmar: TBitBtn;
BtnCancelar: TBitBtn;
Lbl9: TLabel;
TxtResidual: TEdit;
procedure FormShow(Sender: TObject);
procedure BtnConfirmarClick(Sender: TObject);
procedure BtnCancelarClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FrmIndicador: TFrmIndicador;

implementation

uses USimulacao;

{$R *.DFM}

procedure TFrmIndicador.FormShow(Sender: TObject);
begin
  Screen.Cursor := crHourglass;
  try
    TxtPrecSaco.Text := FloatToStr (PrecSaco);
    TxtJuros.Text := FloatToStr (Juros);
    TxtVidaUtil.Text := FloatToStr (VidaUtil);
    TxtEficiencia.Text := FloatToStr (Eficiencia);
    TxtSalario.Text := FloatToStr (Salario);
    TxtSegAloj.Text := FloatToStr (SegAloj);
    TxtDiesel.Text := FloatToStr (Diesel);
    TxtResidual.Text := FloatToStr (Residual);
  finally
    Screen.Cursor := crDefault;
  end;
end;

procedure TFrmIndicador.BtnConfirmarClick(Sender: TObject);
var i: array[0..7] of Real;

```

```

begin
  Screen.Cursor := crHourglass;
  try try
    i[0] := StrToFloat (TxtPrecSaco.Text);
    i[1] := StrToFloat (TxtJuros.Text);
    i[2] := StrToFloat (TxtVidaUtil.Text);
    i[3] := StrToFloat (TxtEficiencia.Text);
    i[4] := StrToFloat (TxtSalario.Text);
    i[5] := StrToFloat (TxtSegAloj.Text);
    i[6] := StrToFloat (TxtDiesel.Text);
    i[7] := StrToFloat (TxtResidual.Text);
    if (i[3] > 100) then
      raise Exception.Create ('A eficiência operacional é inválida');
    PrecSaco := i[0];
    Juros := i[1];
    VidaUtil := i[2];
    Eficiencia := i[3];
    Salario := i[4];
    SegAloj := i[5];
    Diesel := i[6];
    Residual := i[7];
    Close;
  finally
    Screen.Cursor := crDefault;
  end;
except
  on EConvertError do
    raise Exception.Create ('O valor do campo é inválido');
  end;
end;

procedure TFrmIndicador.BtnCancelarClick(Sender: TObject);
begin
  Close;
end;

end.

```

F - Módulo 11: Módulo de Entrada de Velocidade e Número de Máquinas (Arquivo UCaixaDeEntrada.pas)

G - Código Fonte 19: UCaixaDeEntrada.pas <<código>>

```

unit UCaixaDeEntrada;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;

type
  TFrmCaixaDeEntrada = class(TForm)

```

```

    Lbl1: TLabel;
    TxtVeloc: TEdit;
    Lbl2: TLabel;
    TxtNumCol: TEdit;
    CmdConfirmar: TButton;
    CmdCancelar: TButton;
    procedure FormKeyPress(Sender: TObject; var Key: Char);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    FrmCaixaDeEntrada: TFrmCaixaDeEntrada;

implementation

{$R *.DFM}

procedure TFrmCaixaDeEntrada.FormKeyPress(Sender: TObject; var Key: Char);
begin
    if (Key = #27) then
    begin
        Key := #0;
        Close;
    end;
end;

end.

```

F - Módulo 12: Módulo Exportação Para Planilha do MS Excel® (Arquivo UExpExcel.pas)

G - Código Fonte 20: UExpExcel.pas <<código>>

```

unit UExpExcel;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    StdCtrls, Buttons, ExtCtrls, OleServer, Excel97, Grids, Clipbrd;

type
    TFrmExpExcel = class(TForm)
        Lbl1: TLabel;
        RdGrpOpExp: TRadioGroup;
        Pnl1: TPanel;
        BtnConfirmar: TBitBtn;
        BtnCancelar: TBitBtn;
        OpenDialog1: TOpenDialog;
        ExcelApp1: TExcelApplication;
        ExcelWksht1: TExcelWorksheet;
        ExcelWkbk1: TExcelWorkbook;
    end;

```

```

    procedure BtnConfirmarClick(Sender: TObject);
    procedure BtnCancelarClick(Sender: TObject);
private
    { Private declarations }
    Icid: Integer;
    procedure NovaPlanilha (Sender: TObject);
public
    { Public declarations }
    opSimul: Integer;
    areaTotalSimul, capInstal: Real;
    codSimul: String;
    indCultivar, indColhedora: array[0..2] of Real;
    stgVirtual: TStringGrid;
end;

var
    FrmExpExcel: TFrmExpExcel;

implementation

uses USimulacao, UCultivar;

{$R *.DFM}

procedure TFrmExpExcel.BtnConfirmarClick(Sender: TObject);
begin
    Screen.Cursor := crHourglass;
    Icid := GetUserDefaultLCID;
    try
        case RdGrpOpExp.ItemIndex of
            0: ExcelWkbk1.ConnectTo (ExcelApp1.Workbooks.Add (EmptyParam, Icid));
            1: if OpenFileDialog1.Execute then
                ExcelWkbk1.ConnectTo (ExcelApp1.Workbooks.Open (OpenDialog1.FileName, EmptyParam,
                    EmptyParam, EmptyParam, EmptyParam, EmptyParam, EmptyParam, EmptyParam, EmptyParam, EmptyParam, EmptyParam, Icid))
                else
                    Abort;
        end;
        ExcelApp1.Caption := Caption;
        ExcelApp1.DisplayAlerts[Icid] := false;
        ExcelApp1.ScreenUpdating[Icid] := true;
        NovaPlanilha (self);
        ExcelApp1.Connect;
        ExcelApp1.Visible[Icid] := true;
        Close;
    finally
        Screen.Cursor := crDefault;
    end;
end;

procedure TFrmExpExcel.NovaPlanilha(Sender: TObject);
var I, ACol, ARow: Integer;
begin

```

```

ExcelWksht1.ConnectTo (ExcelWkbk1.Worksheets.Add (EmptyParam, EmptyParam, 1,
EmptyParam, lcid) as _Worksheet);
with ExcelWksht1 do
begin
  Activate(lcid);
  with PageSetup do
  begin
    Orientation := 2;
    CenterHorizontally := true;
    CenterVertically := true;
  end;
  Name := codSimul;
  Range['A1','A1'].Value := codSimul;
  Range['A2','A2'].Value := 'ÁREA DE LAVOURA (ha)'; Range['B2','B2'].Value := areaTotalSimul;
  Range['A3','A3'].Value := 'CULTIVARES:';
  for ARow := 0 to High (indCultivar) do
  begin
    FrmCultivar.TblCul.FindKey ([indCultivar[ARow,0]]);
    Range['B' + IntToStr (3 + ARow), 'B' + IntToStr (3 + ARow)].Value :=
FrmCultivar.TblCul.FieldValues['NomeCultivar'] + Format (' x %g', [indCultivar[ARow,2]]);
    Range['C' + IntToStr (3 + ARow), 'C' + IntToStr (3 + ARow)].Value := Format ('(%g ha)',
[indCultivar[ARow,1]]);
  end;
  l := 5 + High (indCultivar);
  if (opSimul = 0) then
  begin
    Range['A' + IntToStr (l), 'A' + IntToStr (l)].Value := 'CAPACIDADE (ha/dia)'; Range['B' + IntToStr
(l), 'B' + IntToStr (l)].Value := caplInstal;
    Range['C' + IntToStr (l + 1), 'C' + IntToStr (l + 1)].Value := 'Nº de colhedoras';
  end
  else
    Range['A' + IntToStr (l), 'A' + IntToStr (l)].Value := 'COLHEDORA';
    Range['A' + IntToStr (l + 1), 'A' + IntToStr (l + 1)].Value := 'Plataforma (m)';
    Range['B' + IntToStr (l + 1), 'B' + IntToStr (l + 1)].Value := 'Potência (kW)';
    for ARow := 0 to High (indColhedora) do
      for ACol := 0 to (2 - opSimul) do
        Range[letra[ACol + 1] + IntToStr (ARow + (l + 2)), letra[ACol + 1] + IntToStr (ARow + (l +
2))].Value := indColhedora[ARow, ACol];
        Range['D2','D2'].Value := 'INDICADORES:';
        Range['E2','E2'].Value := 'Preço da saca (US$/sc)'; Range['F2','F2'].Value := PrecSaco;
        Range['E3','E3'].Value := 'Taxa de juros (% aa)'; Range['F3','F3'].Value := Juros;
        Range['E4','E4'].Value := 'Vida útil (anos)'; Range['F4','F4'].Value := VidaUtil;
        Range['E5','E5'].Value := 'Eficiência (%)'; Range['F5','F5'].Value := Eficiencia;
        Range['E6','E6'].Value := 'Salário (US$/mês)'; Range['F6','F6'].Value := Salario;
        Range['E7','E7'].Value := 'Taxa de seg. e aloj. (% aa)'; Range['F7','F7'].Value := SegAloj;
        Range['E8','E8'].Value := 'Preço do diesel (US$/L)'; Range['F8','F8'].Value := Diesel;
        Range['E9','E9'].Value := 'Residual (% aa)'; Range['F9','F9'].Value := Residual;
        l := l + (High (indColhedora) + 2);
        if (l < 9) then
          l := 9;
        Range['A' + IntToStr (l + 2), 'A' + IntToStr (l + 2)].Value := 'RESULTADO FINAL';
        for ACol := 0 to (stgVirtual.ColCount - 1) do
          for ARow := 0 to (stgVirtual.RowCount - 1) do
            begin
              Clipboard.AsText := stgVirtual.Cells[ACol, ARow];

```

```

        Range[letra[ACol + 1] + IntToStr (ARow + (I + 3)),letra[ACol + 1] + IntToStr (ARow + (I +
3))].PasteSpecial (0,0,0,0);
        Clipboard.Clear;
    end;
    ARow := I + (stgVirtual.RowCount + 2);
    ACol := stgVirtual.ColCount;
    if (ACol < 6) then
        ACol := 6;
    Range['A1',letra[ACol] + IntToStr (ARow)].AutoFormat (xlRangeAutoFormatClassic3, true, true, true,
true, true, true);
    with Range['A1',letra[ACol] + '1'] do
        begin
            Interior.Color := clNone;
            Font.Color := clBlack;
        end;
    with Range['A2',letra[ACol] + IntToStr (ARow)] do
        begin
            HorizontalAlignment := 2;
            Interior.Color := clNone;
            Font.Color := clBlack;
            with Borders.Item[3] do
                begin
                    LineStyle := psDot;
                    Color := clSilver;
                end;
            end;
        ExcelWksht1.Range['A1','A1'].Select;
    end;
end;

procedure TFrmExpExcel.BtnCancelarClick(Sender: TObject);
begin
    Close;
end;

end.

```