

UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ – UNIOESTE
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA AGRÍCOLA

PROGRAMA COMPUTACIONAL PARA AVALIAÇÃO DA
UNIFORMIDADE E EFICIÊNCIA DE APLICAÇÃO EM SISTEMAS DE
IRRIGAÇÃO POR PIVÔ CENTRAL SOB CONDIÇÕES DE
QUIMIGAÇÃO

PAULA CRISTINA DE SOUZA

CASCABEL - PR

2006

PAULA CRISTINA DE SOUZA

**PROGRAMA COMPUTACIONAL PARA AVALIAÇÃO DA
UNIFORMIDADE E EFICIÊNCIA DE APLICAÇÃO EM SISTEMAS DE
IRRIGAÇÃO POR PIVÔ CENTRAL SOB CONDIÇÕES DE
QUIMIGAÇÃO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Agrícola em cumprimento parcial aos requisitos para obtenção do título de Mestre em Engenharia Agrícola, área de concentração em **Engenharia de Recursos Hídricos e Meio Ambiente**.

Orientador: Prof. Dr. Márcio Antônio Vilas Boas.

CASCADEL – Paraná - Brasil

Junho – 2006

Paula Cristina de Souza

“Programa computacional para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central”

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Programa de Pós-Graduação “stricto sensu” em Engenharia Agrícola, da Universidade Estadual do Oeste do Paraná – UNIOESTE, pela comissão formada pelos professores:

Orientador: Prof. Dr. Marcio Antonio Vilas Boas
UNIOESTE/CCET – Cascavel - PR

Prof. Dr. Jorim Sousa das Virgens Filho
UEPG – Ponta Grossa - PR

Prof. Dr. Eurides Küster Macedo Júnior
UNIOESTE – Marechal Cândido Rondon - PR

Prof. Dr. Reginaldo Ferreira Santos
UNIOESTE/CCET – Cascavel - PR

Cascavel, 23 de junho de 2006.

Dedico...

À meu esposo, Eguimar, pela ajuda, pelo apoio, pela confiança, pelo amor. À meus filhos, Gabriela e Marco Antonio, e a meus pais, Maria e Manoel, pela compreensão com a abdicação da convivência em prol deste trabalho.

AGRADECIMENTOS

A DEUS por ter estado sempre a meu lado, porque DEUS é fiel.

À Universidade Estadual do Oeste do Paraná – UNIOESTE, pela oportunidade de realização do curso.

Ao Professor Dr. Márcio Antônio Vilas Boas, pela orientação, dedicação, amizade, paciência, compreensão e conhecimentos oferecidos, essenciais para a conclusão deste trabalho, sendo exemplo de ética e comprometimento.

Aos Professores Dr. Eurides K. Macedo Jr. e Dr. Reginaldo Ferreira Santos pela contribuição ao trabalho apresentado, por meio das pertinentes disciplinas ministradas, sugestões e participação na banca do exame de qualificação.

Aos demais Professores que integram o corpo docente do Curso do Mestrado da Universidade do Oeste do Paraná, pela disposição em dividir seus conhecimentos.

Aos colegas mestrandos que, mesmo com as dificuldades, mostraram-se solidários e contribuíram com o meu percurso acadêmico.

Aos funcionários do Curso de Mestrado da Universidade Estadual do Oeste do Paraná, pela colaboração.

SUMÁRIO

1 INTRODUÇÃO.....	1
2 REVISÃO DE LITERATURA.....	3
2.1 SISTEMA DE IRRIGAÇÃO POR ASPERSÃO TIPO PIVÔ CENTRAL.....	3
2.2 AVALIAÇÃO DA IRRIGAÇÃO.....	5
2.2.1 Coeficiente de Uniformidade.....	5
2.2.2 Eficiência em Potencial de Aplicação.....	6
2.3 APLICAÇÃO DE PRODUTOS QUÍMICOS VIA ÁGUA DE IRRIGAÇÃO.....	8
2.3.1 Fertirrigação.....	10
2.3.2 Herbificação.....	11
2.4 MODELAGEM E SIMULAÇÃO.....	13
2.5 PROGRAMAS COMPUTACIONAIS.....	14
3 MATERIAIS E MÉTODOS.....	16
3.1 DETERMINAÇÃO DE UNIFORMIDADE DE DISTRIBUIÇÃO.....	16
3.1.1 Uniformidade de Heermann e Hein.....	16
3.1.2 Uniformidade Proposta por Evans	17
3.1.3 Determinação da Eficiência Potencial de Aplicação.....	17
3.2 AMBIENTE DE DESENVOLVIMENTO.....	18
3.2.1 Análise de Desempenho.....	19
3.3 O PROGRAMA COMPUTACIONAL.....	19
3.3.1 Módulo de Irrigação.....	20
3.3.1.1 Coeficiente de uniformidade interpolado e coeficiente de uniformidade de Heermann e Hein.....	21
3.3.1.2 Fluxograma do cálculo de eficiência potencial de aplicação.....	22
3.3.2 Módulo de Quimificação.....	23
3.3.2.1 Coeficiente de uniformidade interpolado e coeficiente de uniformidade de Heermann e Hein.....	23
4 RESULTADOS E DISCUSSÃO.....	25
4.1 AMBIENTE DE TRABALHO DO PROGRAMA COMPUTACIONAL....	25

4.2 ANÁLISE DO PROGRAMA COMPUTACIONAL.....	39
4.2.1 Equivalências entre os Coeficientes de Uniformidade de Heermann e os Coeficientes de Uniformidade Interpolado de Evans.....	39
4.2.2 Avaliação do Coeficiente de Uniformidade Acima e Abaixo da Superfície	40
4.2.3 Avaliação do Coeficiente de Uniformidade de Nutrientes.....	45
4.2.4 Gráficos Gerados pelo Programa.....	46
5 CONCLUSÕES.....	50
REFERÊNCIAS.....	51
APÊNDICES.....	52
APÊNDICE A.....	53

LISTA DE FIGURAS

Figura 1 - Estrutura geral do programa proPIVÔ.....	20@~
Figura 2 - Fluxograma do cálculo dos coeficientes de uniformidade.....	22@~
Figura 3 - Fluxograma da eficiência em potencial de aplicação.....	23@~
Figura 4 - Fluxograma do cálculo dos coeficientes de uniformidade.....	24@~
Figura 5 - Tela inicial do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.....	26@~
Figura 6 - Submenu do pivô central.....	26@~
Figura 7 - Exemplo do formulário abrir dados, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.....	27@~
Figura 8 - Menu de visualização de dados de irrigação, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.....	28@~
Figura 9 - Menu de visualização de dados de quimigação, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.....	29@~
Figura 10 - Subitem para alteração dos dados nos módulos de irrigação e quimigação, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.....	31@~
Figura 11 - Subitem para excluir dados da tabela, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.....	32@~
Figura 12 - Exemplo do formulário Incluir dados nos módulos de irrigação e quimigação, do programa proPIVÔ, para avaliação da uniformidade,	

eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.....	34@~
Figura 13 - Exemplo do formulário para salvar arquivo, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.....	35@~
Figura 14 - Menu principal com os resultados representados, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.....	36@~
Figura 15 - Representação do gráfico de lâmina aplicada versus distância radial do centro do pivô, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.....	37@~
Figura 16 - Relatório gerado pelo programa, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.....	39@~
Figura 17 - Gráfico de probabilidade de lâminas aplicadas considerando a distribuição normal.....	44@~
Figura 18 - Gráfico de probabilidade de umidade do solo considerando a distribuição normal.....	45@~
Figura 19 - Mapa de contorno gerado pelo programa proPIVÔ e pelo programa Catch3D para dados de lâminas coletadas, respectivamente.....	47@~
Figura 20 - Mapa de contorno gerado pelo programa proPIVÔ e pelo programa Catch3D para dados de umidades medidas, respectivamente.....	47@~
Figura 21 - Mapa de contorno gerado pelo programa proPIVÔ e pelo programa Catch3D, para o ensaio do pivô X, com 4 linhas de entrada.....	49@~

LISTA DE TABELAS

Tabela 1 - Erro relativo para diferentes coeficientes de uniformidade, baseados em diferentes lâminas de aplicação.....	40@~
Tabela 2 - Comparação entre valores de CUi e CUH para dados de ensaio de lâminas coletadas.....	41@~
Tabela 3 - Comparação entre valores de CUi e CUH para dados de umidade do solo.....	41@~
Tabela 4 - Estatística descritiva dos dados de lâminas e de umidade do solo	43@~

LISTA DE SÍMBOLOS, SIGLAS E ABREVIATURAS

A	- Área total irrigada, em hectares;
a_i	- Área da coroa circular i, em hectares;
CEm	- Condutividade elétrica média, $d s m^{-1}$;
CEmi	- Menor valor de condutividade elétrica, $d s m^{-1}$;
Cf	- Concentração final da solução aplicada após a injeção, ppm;
C_i	- Concentração de íons potássio na solução concentrada, ppm.
Cl	- Concentração do íon potássio na água (prova em branco), ppm;
CU	- Coeficiente de uniformidade de distribuição da condutividade elétrica, em %;
CUC	- Coeficiente de uniformidade de Christiansen, em %;
CU_E	- Coeficiente de uniformidade interpolado de EVANS, em %
CU_H	- Coeficiente de uniformidade de HEERMANN, em %
$\bar{D}$	- Lâmina média aplicada sobre o campo inteiro, e mm;
D	- Lâmina de aplicação na posição (r,θ) , em mm;
E_{pa}	- Eficiência potencial de aplicação, em %;
$\bar{H}$	- Lâmina média coletada ponderada, mm;
n	- Número de coletores alinhados desde o ponto do pivô até a extremidade final da área irrigada.
Q	- Vazão do pivô central, m^3/h ;
Q_f	- Vazão do sistema que passa pelo cabeçal de controle, l/h;
Q_i	- Vazão da solução concentrada injetada, l/h;
r	- Raio, m;
R_i	- Distância do coletor i ao ponto do pivô, m;
S	- Espaçamento entre coletores, m;
T	- Tempo gasto por rotação, em horas;

- V - Volume total de água aplicado na área, m^3 ;
- V_i - Volume de água aplicada na coroa circular representada pelo coletor i , m^3 ;
- Y_i - Lâmina de água aplicada no coletor i , mm;
- $\bar{Y}$ - Lâminas médias de água aplicada na área, mm.

RESUMO

Considerando a importância da avaliação de sistemas de irrigação por pivô central, para determinar suas características operacionais e seu correto manejo, o presente trabalho estabeleceu como objetivo elaborar o programa proPIVÔ, para ambiente *Windows*, em linguagem *Visual Basic*, que auxilia técnicos na obtenção de valores de eficiência, em potencial, de aplicação e coeficientes de uniformidade para sistemas de irrigação por pivô central, e determina também a uniformidade de distribuição de produtos químicos aplicados via água de irrigação. Além disso, o programa constitui um recurso didático, dada a simplicidade do seu uso, as inúmeras alternativas de manipulação de dados e os gráficos e mapas que podem ser obtidos. O programa utiliza uma base de dados *Access* para armazenar as informações dos ensaios, calcula valores de coeficiente de uniformidade de Heermann e o coeficiente interpolado de Evans, gerando gráficos, mapas espaciais e relatórios.

Palavras-chave: Pivô central, irrigação, quimigação, coeficiente de uniformidade.

ABSTRACT

Considering the importance of the evaluation of irrigation systems by central pivot with objective of determining its operational characteristics and its correct handling, the present work had as objective the elaboration of the proPIVÔ program, for Windows environment, in Visual language, which helps technicians to obtain efficiency values, in potential, of application and uniformity coefficients for irrigation systems by central pivot, and also determines the uniformity of distribution of chemical products applied by irrigation water tract. Besides, this program constitutes a didactic resource, due to the simplicity of its use, the several alternatives of data manipulation and the graphs and maps which can be obtained. The program uses an Access data base to store the rehearsals/tests, it calculates values of Heermann uniformity coefficient and Evans interpolated coefficient, generating graphs, space maps and reports.

Keywords: center pivot, irrigation, chemigation, coefficient of uniformity.

1 INTRODUÇÃO

A irrigação é uma das mais importantes tecnologias empregadas para aumentar a produtividade, fornecendo uma maior eficiência e estabilidade na produção agrícola, permitindo a suplementação de água nos períodos de estiagem e a utilização contínua da área. A agricultura irrigada é a atividade agrícola que mais utiliza água e, em muitos casos, com baixo aproveitamento devido, em grande parte, a problemas de manejo.

Com a estimativa de um grande crescimento populacional, a melhoria das técnicas de irrigação é um fator essencial para manter o suprimento de alimentos em função da crescente demanda e, ao mesmo tempo, garantir a conservação do meio ambiente.

O sistema de irrigação por pivô central foi criado em 1952, nos EUA, e difundido no Brasil na década de 1970 e, desde então, tem proporcionado um significativo avanço da agricultura irrigada do país. A grande aceitação do equipamento se deve às suas características, que permitem a irrigação mecanizada de extensas áreas, mesmo de topografia irregular, facilitando a utilização de práticas de quimigação. A avaliação destes sistemas requer, entre outras informações, o conhecimento da distribuição da água aplicada. A uniformidade de distribuição da água é quantificada por coeficientes de uniformidade. Para sistemas de irrigação pivô central, o coeficiente mais utilizado é o de Christiansen, modificado por Heermann e Hein (KELLER; BLIESNER, 1990).

Após a instalação do equipamento de irrigação por pivô central convém avaliar o seu desempenho, para determinar suas características operacionais em interação com o meio agrícola no qual está operando, subsidiando eventuais correções e tomadas de decisão sobre manejo e utilização do equipamento. As

avaliações do sistema por modelos de simulação são práticas muito utilizadas por pesquisadores, que vêem nesses modelos o melhor mecanismo para sintetizar o conhecimento sobre os diferentes componentes do sistema, arquivando dados de forma sintética, e transferindo resultados de pesquisa para os usuários.

A aplicação de produtos químicos por meio de sistemas de irrigação, denominada quimigação, tem no movimento turbulento das águas um meio de manter o material químico uniformemente distribuído nas tubulações de condução, e isso faz dos sistemas pressurizados de irrigação os mais utilizados nesse processo.

Considerando esse contexto, foi estabelecido como objetivo para este projeto desenvolver um programa computacional, de caráter didático, que possibilitasse a avaliação da uniformidade e eficiência de distribuição de água em um sistema de irrigação por pivô central, além de fornecer dados de uniformidade de distribuição de produtos químicos via água de irrigação, apresentando como resultado valores de coeficientes, gráficos e relatórios.

2 REVISÃO DE LITERATURA

2.1 SISTEMA DE IRRIGAÇÃO POR ASPERSÃO TIPO PIVÔ CENTRAL

Segundo a NBR, (NBR 14244, 98) pivô central é uma máquina de irrigação automatizada constituída por uma linha de emissores que gira em torno do ponto pivô, suportada por uma ou mais torres autopropelidas. A água é suprida ao ponto pivô e, posteriormente, flui pela tubulação aérea, abastecendo os emissores. A unidade irriga uma área de formato circular.

O sistema de irrigação por aspersão tipo pivô central foi introduzido no Brasil no final dos anos 1970 e, pela grande versatilidade do equipamento, fez sucesso entre os agricultores. Atualmente é o equipamento de irrigação mais comercializado e o que mais tem contribuído para o incremento da área irrigada do país. No início eram máquinas simples acionadas por turbinas hidráulicas. A uniformidade de distribuição de água dependia do relevo e da extensão do pivô, mas com a incorporação de diversos acessórios e dispositivos tornou-se possível a utilização desse equipamento nas mais diversas situações (VILELA, 2002).

FARIAS e VIEIRA (1986) relatam que o pivô central é uma linha lateral com emissores por aspersão que opera em círculos, constituído de uma linha lateral com emissores, ancorada em uma das extremidades e suportada por torres dotadas de rodas equipadas com unidades propulsoras. Segundo esses autores, a velocidade de rotação das torres em torno do ponto central (ponto pivô) é

regulada por meio de um relé percentual situado no painel do equipamento, que comanda a velocidade de deslocamento da última torre.

A característica principal do pivô central é a aplicação de água pelo método de aspersão, mas sob taxas variáveis ao longo de sua tubulação, que se desloca no sentido circular em torno de uma torre fixa ou pivô (ALBUQUERQUE, 2003).

Segundo BERNARDO (1985), o comprimento do pivô pode variar de 200 a 800 metros, irrigando áreas de, aproximadamente, 12 a 200 hectares, respectivamente, com a distância entre as torres variando de 40 a 76 m.

Cuenca (1989), citado por VILELA (2002), comparando a irrigação por pivô central com outros sistemas de irrigação por aspersão, constata que o primeiro apresenta diversas vantagens: reduzida necessidade de mão-de-obra para a realização da irrigação; capacidade de obter maiores produtividades em razão dos menores turnos de rega adotados, mantendo o conteúdo de água no solo próximo à capacidade de campo; elevada uniformidade de distribuição de água quando os emissores e válvulas reguladores de pressão forem corretamente dimensionados, instalados e encontrar-se em perfeitas condições de operação; facilidade de automação; custo do equipamento que decresce com o aumento da área irrigada. As principais limitações do sistema, segundo Cuenca (1989), citado por VILELA (2002), são: perdas de áreas cultivadas da ordem de 20%, devido à particularidade de irrigarem áreas circulares; a intensidade de aplicação de água na extremidade final do equipamento é bastante elevada, podendo chegar a 100 mm/h, o que limita sua utilização em solos com menor capacidade de infiltração de água; a área a ser irrigada deve estar totalmente livre de obstáculos, embora já existam alternativas para contornar algumas situações; dificuldade de assistência técnica especializada em algumas regiões, principalmente para os equipamentos mais modernos e sofisticados.

2.2 AVALIAÇÃO DA IRRIGAÇÃO

2.2.1 Coeficiente de Uniformidade

Muitos coeficientes de uniformidades foram propostos para os sistemas de irrigação por aspersão. O mais antigo e amplamente utilizado é o proposto por Christiansen (1942), que utiliza como medida de dispersão o desvio médio absoluto. Apesar de muitos outros coeficientes terem sido apresentados como alternativas ao de Christiansen, nenhum apresentou vantagens significativas (FRIZZONE; DOURADO NETO, 2003).

O coeficiente de uniformidade de Christiansen (CUC) é expresso da seguinte forma:

$$CUC = \frac{100}{n} \left[\sum_{i=1}^n \left(\frac{Y_i}{\bar{Y}} \right)^2 - \frac{n-1}{n} \right] \quad (01)$$

Em que:

CUC = coeficiente de uniformidade de Christiansen, em %;

Y_i = lâminas de água coletadas em ensaios de uniformidades, no i-ésimo coletor, em mm;

$\bar{Y}$ = lâminas média de água coletada, em mm;

n = número de observações.

Para se calcular o coeficiente de uniformidade de distribuição de água em sistemas de irrigação tipo pivô central é usual ponderar as lâminas de água coletadas nos pluviômetros, pois cada coletor representa áreas progressivamente maiores, a partir do ponto do pivô, conforme Heermann e Hein, 1968, citados por

FRIZZONE e DOURADO-NETO (2003), determinando o coeficiente de uniformidade para sistemas de irrigação por pivô central pela equação:

$$CU_H = 100 \left(1 - \frac{\sum_{i=1}^n R_i |Y_i - \bar{Y}_p|}{\sum_{i=1}^n Y_i R_i} \right) \quad (02)$$

Em que:

$$\bar{Y}_p = \frac{\sum_{i=1}^n R_i Y_i}{\sum_{i=1}^n R_i} \quad (03)$$

Sendo $\bar{Y}_p$ a lâmina média de irrigação ponderada pela distância R_i .

EVANS, HANS e KROEGER (1995) salientam que a lâmina de aplicação da água no sistema de irrigação por pivô central não é uniforme no campo, ela depende do conjunto de aspersores, da topografia do campo, do movimento da máquina e de muitos outros fatores. Seria complexo desenvolver uma fórmula teórica para a distribuição da lâmina, mas um mapa de distribuição espacial da lâmina aplicada pode ser construído a partir de um número suficientemente grande de observações.

2.2.2 Eficiência em Potencial de Aplicação

A eficiência potencial de aplicação é a relação entre a quantidade de água infiltrada na parcela e a quantidade de água aplicada à parcela. Os parâmetros de eficiência expressam a qualidade da irrigação e, ao mesmo tempo, incorporam algumas conseqüências da uniformidade. Enquanto a uniformidade depende

somente do grau de dispersão da lâmina aplicada, a eficiência depende tanto da uniformidade quanto da forma como o sistema é operado. A eficiência em potencial de aplicação mede a fração de água aplicada que foi infiltrada na parcela. O seu complemento refere-se às perdas por evaporação durante a aplicação, deriva pelo vento e escoamento superficial (FRIZZONE; DOURADO NETO, 2003).

Na irrigação em pivô central, quando não ocorre escoamento superficial em razão da alta intensidade de precipitação na extremidade do pivô, a eficiência potencial de aplicação é determinada pela seguinte equação (BERNARDO, 1985):

$$E_{pa} = \frac{Q \cdot T}{10 \cdot A} \quad (04)$$

$$H_m = \frac{Q \cdot T}{10 \cdot A} \quad (05)$$

Em que:

E_{pa} = eficiência potencial de aplicação (%);

$\bar{H}$ = lâmina média coletada ponderada (mm);

Q = vazão do pivô central (m³/h);

T = tempo gasto por rotação (h);

A = área total irrigada (ha).

Segundo BERNARDO (1985), a eficiência de aplicação tem variado entre 70% e 90 %, sendo o valor de 80% possível de ser obtido sob condições normais de dimensionamento.

2.3 APLICAÇÃO DE PRODUTOS QUÍMICOS VIA ÁGUA DE IRRIGAÇÃO

Threadgill (1985), citado por VILELA (2002), comenta que o sistema de irrigação por pivô central tornou-se uma efetiva ferramenta para a aplicação de produtos químicos nas lavouras irrigadas.

A aplicação de produtos químicos na lavoura por intermédio da água de irrigação é denominada quimigação. Esse termo começou a ser usado nos Estados Unidos da América na década de 1970. O aparecimento dos sistemas de irrigação por pivô central, no fim da década de 1940, e a expansão de seu uso, na década de 1950, permitiram a aplicação de fertilizantes por meio da água de irrigação de maneira mais ampla. A expansão do uso da quimigação, incluindo vários produtos químicos, gerou novos termos como fertirrigação, herbigação, fungigação, insetigação, nematigação, etc. (VIEIRA, 1994).

VIANA et al. (2002) descrevem a quimigação como uma prática amplamente usada em pivô central, devido à sua facilidade de automação e possibilidades de aplicação eficiente da água. O comprimento da lateral do sistema é bastante variado, dependendo da necessidade do produtor, das características topográficas e das dimensões da área a ser irrigada.

O equipamento de pivô central deve estar bem ajustado, para promover uma aplicação eficiente do produto químico injetado. De acordo com VIANA et al. (2002), pois a calda estará misturada à água de irrigação, a uniformidade de aplicação do produto químico praticamente se confunde com a da aplicação da água e, portanto, é necessário que essa uniformidade seja elevada, para que se obtenha uma boa uniformidade de aplicação do produto. Segundo VIANA et al. (2002), o sistema de irrigação pressurizado vem sendo cada vez mais utilizado nesse processo, devido ao regime turbulento da água, que ajuda a manter o produto químico uniformemente distribuído nas tubulações de condução.

A uniformidade de distribuição dos produtos químicos é fator essencial para o sucesso da quimigação e, geralmente, ela é proporcional à uniformidade de distribuição da água pelo sistema de irrigação. O CU de um pulverizador tratorizado varia de 50% a 92%; na pulverização aérea o CU é de, aproximadamente, 70%. Quando devidamente calibrados e operados, os sistemas de aspersão com movimentação lateral, como o pivô central, atingem CU's de 90% ou mais. Esses elevados coeficientes de uniformidade fazem do pivô central um dos sistemas de irrigação mais adequados para a quimigação, superando, em uniformidade, as pulverizações terrestres e aéreas (VIEIRA, 1994).

De acordo com DOURADO NETO et al. (1998), em sistemas do tipo pivô central, as aplicações via água de irrigação de fungicidas, inseticidas e herbicidas pós-emergentes, que têm como alvo a parte aérea da planta tende a apresentar uma eficiência relativamente baixa, devido ao alto volume de calda aplicada.

Segundo EVANS, HANS e KROEGER (1995), o cálculo tradicional da uniformidade de distribuição com base na avaliação hidráulica de projetos de irrigação não é apropriado para avaliar a distribuição de produtos químicos por meio de pivô central. É necessário o uso de uma distribuição espacial de entrada de água e nutrientes. EVANS, HANS e KROEGER (1995) desenvolveram um algoritmo bidimensional para construção dos mapas espaciais de distribuição, baseados em número limitado de observações. O coeficiente de uniformidade é calculado a partir das lâminas observadas.

EVANS, HANS e KROEGER (1995) chegaram a um coeficiente de uniformidade interpolado, baseado no desvio médio, chamado de CU_i :

$$CU_i = 100 * \left\{ 1,0 - \frac{\frac{1}{A} \int_A |Y(r, \theta) - \bar{Y}| dA}{\bar{Y}} \right\} \quad (06)$$

Em que:

$Y(r, \theta)$ = lâmina de aplicação na posição (r, θ) ;

$\bar{Y}$ = lâmina média aplicada sobre o campo inteiro;
A = área total do campo.

2.3.1 Fertirrigação

Fertirrigação é a técnica de aplicação de fertilizantes via água de irrigação ou sistemas de irrigação. Algumas das grandes vantagens da fertirrigação, citadas por PAPADOPOULOS (1999), são: a alta produtividade, a qualidade do produto, a eficiência na recuperação do fertilizante, perdas mínimas de fertilizantes ocasionadas por lixiviação, controle da concentração de nutrientes na solução do solo, controle da forma e taxa de variação dessas formas, principalmente dos fertilizantes nitrogenados; e flexibilidade na escolha da época de distribuição do fertilizante em relação à exigência da cultura, baseada nos seus estágios de desenvolvimento fisiológico.

Conforme SOUSA et al. (2003), uma boa uniformidade de distribuição de água de irrigação é fundamental para um bom manejo da fertirrigação. Entretanto, a uniformidade de distribuição de nutrientes também pode ser afetada pela variação da sua concentração na água de irrigação, esta uniformidade será relacionada ao grau de agitação ou mistura da solução no tanque, à variação da taxa de injeção na solução e à variação da vazão no sistema de irrigação durante a fertirrigação.

Segundo FRIZZONE et al. (1998), o manejo adequado da fertirrigação requer que a injeção de fertilizante seja iniciada quando toda tubulação estiver cheia de água e os emissores em pleno funcionamento. Neste procedimento, o tempo de fertirrigação é muito importante na uniformidade de distribuição dos fertilizantes em toda a área cultivada, dividindo-se a aplicação em três etapas: a primeira para enchimento da tubulação; a segunda para a aplicação propriamente da solução na água de irrigação; e a terceira para promover a lavagem da tubulação e dos emissores, tempo que raramente deve ser inferior a 30 min.

Todavia, Sampaio et al. (1997), citados por SOUSA et al. (2003), afirmam que a estabilização da concentração de fertilizantes nas linhas de irrigação ocorre após 20 minutos da ligação do sistema. Segundo Zanini (1987), também citado por SOUSA et al. (2003), um maior tempo de fertirrigação conduzirá a uma melhor uniformidade de distribuição de fertilizantes na área.

A avaliação da uniformidade de distribuição de nutrientes em sistemas de irrigação requer a coleta de água e, posteriormente, determina-se a concentração do elemento na amostra. Devido às dificuldades na determinação de alguns elementos, estudos têm avaliado a uniformidade de distribuição de produtos químicos por condutividade elétrica, como indicador da concentração (SOUSA et al., 2003).

2.3.2 Herbificação

No Brasil, o uso de herbicidas nas principais culturas vem tomando grande impulso, principalmente em áreas em que se procura obter alta produtividade. A maioria dos herbicidas é recomendada para atuar de forma preventiva (aplicações de pré-plantio e pré-emergência), ou na fase inicial da cultura, possibilitando-se à lavoura condições de emergir e crescer sem os efeitos deletérios das plantas daninhas. Visando diminuir os custos da aplicação dos herbicidas, os produtores começaram a usar a irrigação para transporte e distribuição dessas substâncias (SILVA; KARAM; COSTA, 1994). Segundo Ogg et al. (1983), citados por SILVA, KARAM e COSTA (1994), a herbificação pode reduzir em até 50% os custos, em relação à aplicação tratorizada. Pode-se também reduzir o consumo de energia em até 90%. Onde os pivôs centrais são usados, um homem pode supervisionar o tratamento de dois ou mais pivôs, reduzindo a mão-de-obra necessária para operar os pulverizadores. Reduz também a compactação do solo e os perigos de contaminação de operadores e apresenta maior uniformidade de aplicação e aumento das atividades herbicidas.

Como desvantagem, Ogg et al. (1983), citados por SILVA, KARAM e COSTA (1994), enumeram: requerimento de maior conhecimento de manejo, requerimento de equipamentos adicionais, aumento do risco de contaminação ambiental e aumento do tempo de aplicação. Segundo COSTA, VIEIRA e VIANA (1994), os trabalhos sobre a aplicação de herbicidas via água de irrigação estão em fase de instalação, ainda não são encontrados dados definitivos publicados. Segundo VIANA (1994), os pesquisadores SILVA, KARAM e COSTA (1994) conduziram um trabalho em Sete Lagoas, MG, avaliando herbicidas em culturas de milho. COSTA, VIEIRA e VIANA (1994) salientam que, nos Estados Unidos da América, a aplicação de herbicidas via água de irrigação tem merecido muita atenção, principalmente na Geórgia, de onde saem quase todos os trabalhos sobre herbicidação.

CHISTOFFOLETI E DOURADO-NETO (2001) afirmam que a aplicação de herbicidas pós-emergentes em sistemas de irrigação por pivô central apresenta eficiência de aplicação muito baixa, mas quando a finalidade do herbicida é atingir o banco de sementes de plantas daninhas presente nos 5 centímetros superficiais do solo (região do banco de sementes que pode sofrer o processo de quebra de dormência, iniciar a germinação e ocorrer a emergência) observa-se que o regime de escoamento hidráulico turbulento faz com que o herbicida tenha a mesma distribuição da água de irrigação. Portanto, torna-se imprescindível uma boa uniformidade de aplicação no uso da herbicidação.

COSTA, VIEIRA e VIANA (1994) citam trabalhos de Banks e Dowler (1986) que estudaram a distribuição dos herbicidas em amostragem feita por meio de coletores distribuídos, diagonalmente, ao longo das curvas traçadas por três torres do pivô central e, paralelamente, com a linha de irrigação, em 12 pontos espaçados uniformemente, aplicando a lâmina à baixa e alta pressão. Nesse trabalho, foram detectados padrões discerníveis de distribuição de herbicida, atribuídos à variação de bicos e ao vento.

Outro trabalho citado por COSTA, VIEIRA e VIANA (1994), foi o de Wyman et al. (1986), em que a aplicação do herbicida por meio do pivô central, com uma barra de bicos colocada sob o pivô, a uma altura aproximada de um

metro acima do solo, cujos resultados, em três sistemas de aplicação, não detectaram diferenças.

Dowler (1984), citado por COSTA, VIEIRA e VIANA (1994), aplicou herbicida junto com fertilizante em um sistema de irrigação por pivô central, utilizando somente água como veículo e obteve resultados satisfatórios.

Segundo COSTA, VIEIRA e VIANA (1994), no Brasil inúmeros produtores já fazem uso dos sistemas de herbificação com sucesso, mas faltam o respaldo da pesquisa e o registro legal.

2.4 MODELAGEM E SIMULAÇÃO

No Brasil, o surgimento dos primeiros modelos de simulação na agricultura data do início da década de 1970. No antigo Departamento de Métodos Quantitativos–DMQ da Embrapa, de 1974 a 1977, exemplos bem sucedidos de modelos foram desenvolvidos, como o do *Anticarsia gemmatilis* na soja e o de crescimento de rebanho bovino de corte. (SILVA; BERGAMASCO; VENDITE, 2002).

As vantagens em se utilizar modelagem e simulação no âmbito da pesquisa em produção vegetal, que tem sido levantada pela equipe da Embrapa, em várias ocasiões, e citadas por SILVA, BERGAMASCO e VENDITE (2002), são as seguintes: são auxiliares importantes na geração de novas idéias e tecnologias e na sua posterior avaliação; oferecem apoio estratégico na fase de teste e adaptação de tecnologias; tornam disponíveis uma infra-estrutura centralizada de informática (banco de dados experimentais) e de modelos; permitem a análise de sensibilidade em sistemas de produção; proporcionam indicadores de prioridades de pesquisa e ajudam na escolha de melhores opções de modelagem matemática e na escolha de ferramentas computacionais mais

adequadas; possuem a capacidade de organizar/entender a complexidade de um sistema de forma didática.

Conforme DOURADO-NETO et al. (1998), modelos não são simples mecanismos para arquivar informações, produzindo estimativas. Os modelos representam o melhor mecanismo de sintetizar conhecimentos sobre diferentes componentes de um sistema, arquivando dados de forma sintética e transferindo resultados de pesquisa para os usuários.

Entre os trabalhos com uso de modelagem e simulação pode-se destacar: Modelo para simulação do desempenho de sistemas de irrigação tipo pivô central de RODRIGUES et al. (2001); Modelo bi-dimensional da lixiviação de nitrato para vários tipos de fertirrigação sob microaspersão de GARDENAS et al. (2005); Modelo aplicado ao planejamento da cafeicultura irrigada de SOUSA et al. (2003); Simulação da produtividade do milho pelo modelo CERES-Maize em função da lâmina e da uniformidade de aplicação de água de FREITAS et al. (2003); Análise de um distrito irrigado no noroeste da Espanha – Avaliação de irrigação e programação de DECHMI et al. (2003); Sistema de controle computadorizado para estudos de aplicação de água e produtos químicos, espacialmente variado, em sistemas de irrigação de movimentos contínuos de KING et al. (1999); Modelos de simulação para análise e apoio à decisão em agrossistemas de SILVA, BERGAMASCO e VENDITE (2002), entre outros.

2.5 PROGRAMAS COMPUTACIONAIS

Atualmente, existem alguns *softwares* desenvolvidos para irrigação do tipo pivô central que tratam do seu dimensionamento, simulação de desempenho e fertirrigação. Dentre eles pode-se destacar: Frutifert, Fértil, Dimpivo, Geopivo, dentre outros.

O Frutifert de AGUIAR NETTO et al. (2002), desenvolvido em ambiente *Delphi*, pelo uso da orientação a objeto. O *software* simula o manejo de irrigação em fruteiras tropicais, disponibilizando resultados de pesquisa e facilitando os cálculos de aplicação de água, por meio dos seguintes métodos: balanço de água na zona radicular, tensiometria e tanque Classe A. Permite também estabelecer fertilizações via irrigação de fruteiras tropicais, com base em dados sistematizados por universidades e institutos de pesquisa de vários estados do Brasil.

O Dimpivo, de ANDRADE JÚNIOR, AGUIAR NETO e BASTOS, (1995), dimensiona o sistema de irrigação por pivô central, utilizando a linguagem *Turbo basic*. O modelo realiza uma análise hidráulica do sistema, selecionando bocais a partir dos valores de pressão e vazão.

O Fértil, desenvolvido por PAPADOPOULOS (1999), prevê e recomenda um calendário de irrigação/fertirrigação, estimando a exigência de água pela cultura. Também trabalha com fertirrigação em árvores frutíferas e têm como parâmetros: a evaporação (EPAN), as exigências de nutrientes pela cultura, na fertilidade do solo e na eficiência da absorção dos diversos nutrientes nos diferentes sistemas de irrigação.

O Geopivo, desenvolvido por RODRIGUES et al. (2001), simula o desempenho de sistemas de irrigação tipo pivô central. Baseado no conceito de sistemas de informações geográficas, demonstrando que o modelo pode ser utilizado para representar a viabilidade espacial e temporal da pressão de operação, diâmetro médio da gota e das áreas com ou sem risco de escoamento superficial.

3 MATERIAIS E MÉTODOS

3.1 DETERMINAÇÃO DE UNIFORMIDADE DE DISTRIBUIÇÃO

3.1.1 Uniformidade de Heermann e Hein

Para o cálculo do coeficiente de uniformidade de distribuição de água foi adotado o coeficiente de Hermann e Hein, apresentados na equação 2.

Como esse coeficiente é tradicionalmente utilizado no estudo da uniformidade de distribuição de água em sistemas de irrigação por pivô central, foi adotado como parâmetro de comparação entre o coeficiente proposto por EVANS, HANS e KROEGER (1995).

O programa tem opção de análise de mais de uma linha de ensaio, neste caso, o coeficiente de Hermann e Hein apresentado. Trata-se de uma média simples dos valores encontrados para esse coeficiente nas diferentes linhas coletadas. O procedimento é semelhante ao adotado por EVANS, HANS e KROEGER (1995).

3.1.2 Uniformidade Proposta por Evans

Segundo EVANS, HANS e KROEGER (1995), para construção de uma distribuição de lâmina, é necessário o uso de um algoritmo de interpolação bidimensional. Com base nesse dado, os autores, chegaram a um coeficiente de uniformidade interpolado chamado de CU_i , conforme descrito na equação 6. Como o coeficiente de uniformidade interpolado é um parâmetro adimensional e a equação 6 desenvolvida pode ser utilizada para a construção de mapas de lâminas de aplicação de água e nutrientes, a partir de dados coletados, podendo assim ser utilizada para obtenção dos coeficientes de uniformidade das aplicações de produtos químicos e biológicos via água de irrigação, é necessário a substituição do parâmetro Y_i para os valores de concentração observados no coletor i .

3.1.3 Determinação da Eficiência Potencial de Aplicação.

Para determinar a eficiência em potencial de aplicação do sistema, foi feita uma média ponderada das lâminas coletadas e o resultado obtido representa a quantidade de água infiltrada na parcela.

Para determinação da quantidade de água aplicada à parcela foi utilizada a equação 5. A eficiência potencial de aplicação foi obtida pela relação entre a água infiltrada na parcela e a água aplicada na parcela, representada na equação 4.

3.2 AMBIENTE DE DESENVOLVIMENTO

O sistema foi desenvolvido em linguagem *Visual Basic*®, usando a base de dados *Access* para armazenar as informações dos projetos realizados, com o sistema operacional *Windows*. Para o desenvolvimento, buscou-se os algoritmos desenvolvidos por HERGERT (1996), WEBB (1996), SILER e SPOTTS (1999).

O programa é composto por dois módulos, um de irrigação em sistemas de pivô central e outro com quimigação, para o mesmo sistema, ambos desenvolvidos para fornecer os resultados do coeficiente de uniformidade. O módulo de irrigação fornece também o resultado de eficiência potencial de aplicação.

Além de relatórios impressos com os dados de eficiência e uniformidade do projeto de irrigação, o programa disponibiliza uma opção de geração de gráficos, em que são plotados os gráficos de lâmina aplicada *versus* distância radial do centro do pivô e um gráfico tridimensional da distribuição espacial da lâmina aplicada.

O programa trabalha também com a aplicação de produtos químicos distribuídos a partir da água de irrigação. Nesse caso, é necessária a entrada de dados das concentrações obtidas pelos ensaios realizados com fertirrigação e herbigação. Com esses dados são fornecidos os valores de coeficientes de uniformidade e são gerados gráficos de concentrações *versus* distância radial do centro do pivô e o gráfico tridimensional da distribuição espacial das concentrações obtidas.

3.2.1 Análise de Desempenho

A análise do programa computacional desenvolvido foi fundamentada em simulações com dados de experimentos realizados por EVANS, HANS e KROEGER (1995) e VILAS BOAS (2000), para irrigação e dados de concentração de experimentos e de PEREIRA, FRIZZONE e DOURADO NETO (2005), para quimigação.

A análise do desempenho do programa computacional foi realizada com base em comparações com resultados obtidos no programa computacional *Catch3D*, desenvolvido por ALLEN e MERKLEY (2005).

3.3 O PROGRAMA COMPUTACIONAL

O programa trabalha com dois módulos, irrigação e quimigação, e funciona a partir de um banco de dados, que carrega as variáveis de ambiente que serão utilizadas nas rotinas de cálculo e controlam as demais interfaces do programa. Para os dois módulos são permitidos os cálculos de coeficientes de uniformidade e geração de gráficos. São dadas ao usuário duas opções de gráficos, para que seja possível uma melhor visualização do ensaio. Um dos gráficos é o do perfil de distribuição, podendo o usuário analisar o perfil de cada linha de ensaio, o outro é um mapa de contorno obtido a partir do algoritmo de interpolação proposto por EVANS, HANS e KROEGER (1995). A Figura 1 mostra o fluxograma da estrutura geral do programa.

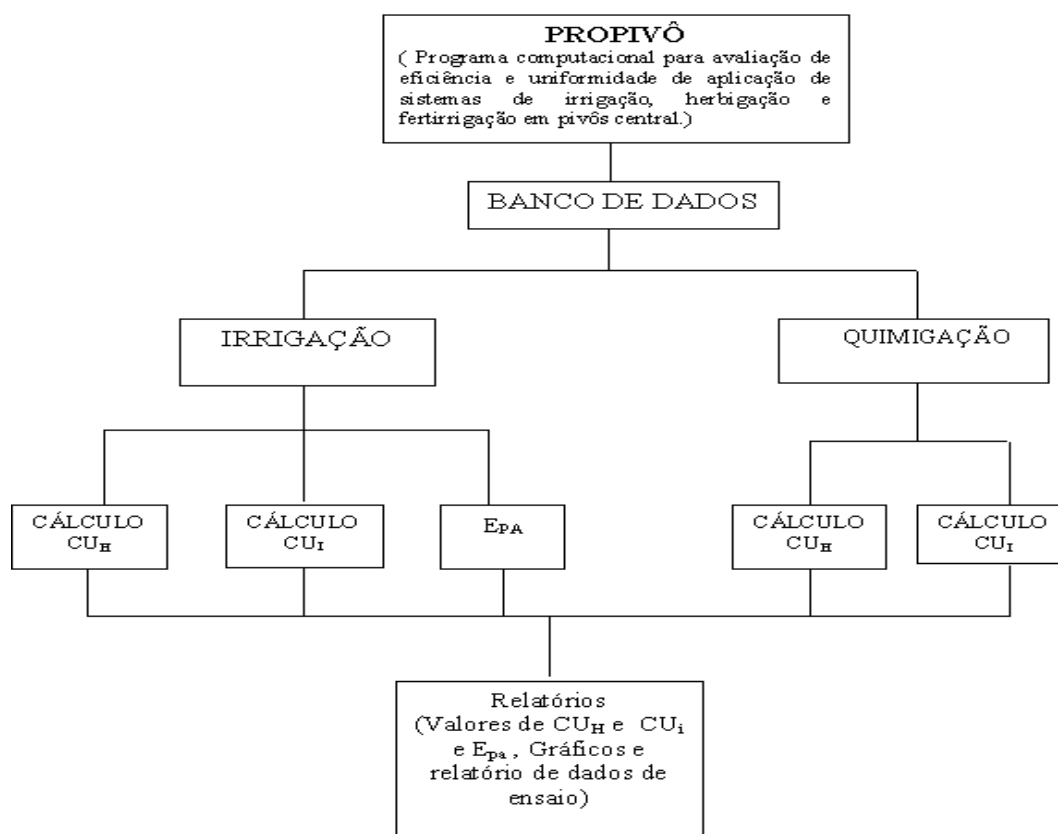


Figura 1 - Estrutura geral do programa proPIVÔ.

3.3.1 Módulo de Irrigação

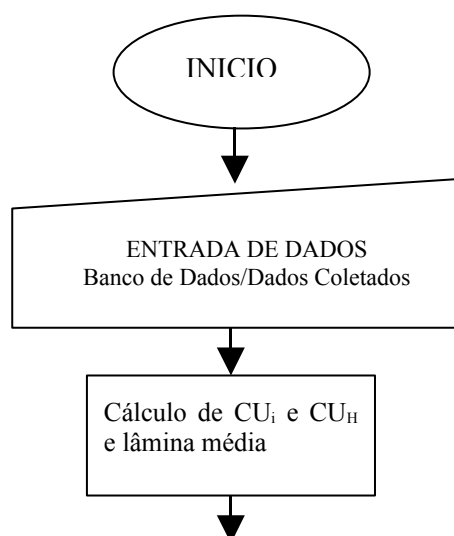
No módulo de irrigação há as seguintes opções: coeficiente de uniformidade interpolado, coeficiente de uniformidade de Heermann e Hein, lâmina média ponderada e eficiência em potencial de aplicação.

3.3.1.1 Coeficiente de uniformidade interpolado e coeficiente de uniformidade de Heermann e Hein

O processo matemático utilizado para o cálculo do coeficiente de uniformidade interpolado é determinado pela equação de EVANS, HANS e KROEGER (1995) (Equação 6). Os valores utilizados nos cálculos destes coeficientes, que são as lâminas coletadas e as distâncias destas ao centro do pivô, são fornecidos ao programa por meio de um banco de dados, ou pela inserção dos mesmos por uma interface do programa. No banco de dados o usuário tem a opção de alterar os dados tabelados, inserir novos dados dentro de uma tabela, salvar alterações, voltar e gravar.

Os mesmos dados utilizados para o cálculo do CU_i são utilizados também para calcular o coeficiente de uniformidade de Heermann e Hein (Equação 2), que tem como finalidade fornecer parâmetros para a comparação do CU_i com o CU_H , que é um coeficiente tradicionalmente utilizado.

O fluxograma referente aos cálculos dos coeficientes de uniformidade é apresentado na Figura 2:



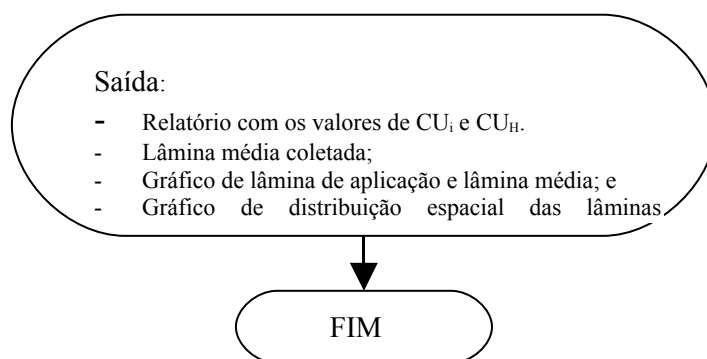
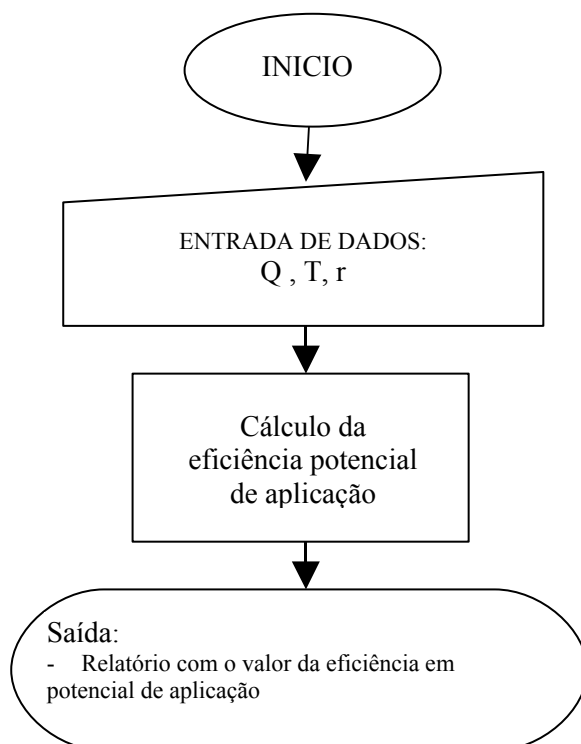


Figura 2 - Fluxograma do cálculo dos coeficientes de uniformidade.

3.3.1.2 Fluxograma do cálculo de eficiência potencial de aplicação

A opção eficiência potencial de aplicação fornece como resultado os valores dessa eficiência plotados nos relatórios do programa. Os dados de entrada fornecidos ao programa são o tempo de rotação (T), a vazão (Q) e o raio do pivô. O processo matemático utilizado para o cálculo dos valores de eficiência em potencial de aplicação é o descrito por BERNARDO (1985) (equações 4 e 5).

O fluxograma referente aos cálculos da eficiência potencial de aplicação é apresentado na Figura 3:



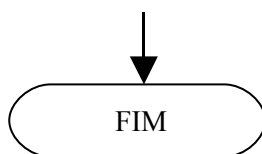


Figura 3 - Fluxograma da eficiência em potencial de aplicação.

3.3.2 Módulo de Quimigação

O módulo de quimigação possui as opções: coeficiente de uniformidade interpolado e coeficiente de uniformidade de Heermann e Hein. Os dados fornecidos ao programa para efetuar os cálculos de uniformidade são baseados na concentração de produto químico encontrada nos coletores dispostos nos ensaios.

3.3.2.1 Coeficiente de uniformidade interpolado e coeficiente de uniformidade de Heermann e Hein.

O cálculo do coeficiente de uniformidade interpolado é determinado pela equação descrita por EVANS, HANS e KROEGER (1995) (Equação 6), mas, neste caso, as lâminas coletadas serão os valores de concentração dos produtos químicos observados nos coletores dispostos no ensaio. Os valores utilizados nos cálculos desses coeficientes e as distâncias destas ao centro do pivô também são fornecidos ao programa por um banco de dados ou pela sua inserção por meio de uma interface do programa.

No banco de dados o usuário tem as mesmas opções: alterar os dados tabelados, inserir novos dados em uma tabela, salvar alterações, voltar e gravar.

Serão calculados CU_i e CU_H . O fluxograma referente aos cálculos desses coeficientes de uniformidade é apresentado na Figura 4:

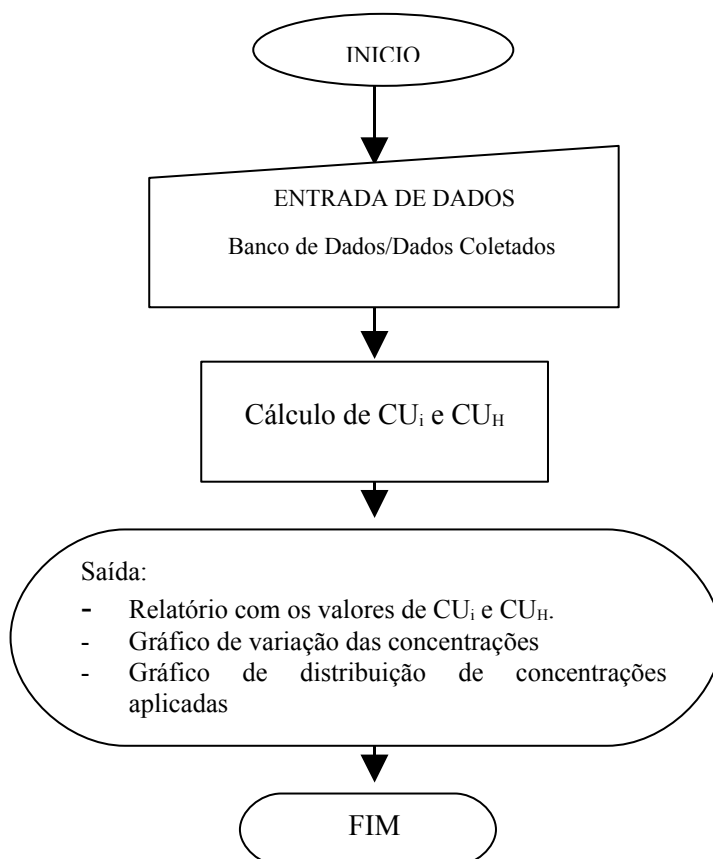


Figura 4 - Fluxograma do cálculo dos coeficientes de uniformidade.

4 RESULTADOS E DISCUSSÃO

4.1 AMBIENTE DE TRABALHO DO PROGRAMA COMPUTACIONAL

Utilizando a linguagem computacional *Visual Basic*, versão 6.0, criou-se o programa proPIVÔ.

O programa executado abre um menu principal (Figura 5), no qual são apresentadas as opções: Arquivo, Pivô Central e Ajuda. Abaixo há uma barra de ferramentas com ícones representando alguns comandos. Na primeira vez que o menu principal é apresentado os ícones que representam os comandos: Abertura de um novo arquivo, Abrir arquivo existente e Sair encontram-se ativos e os demais desativados. A partir do momento em que houver a seleção de um ensaio os itens: Gravar, Gerar gráfico de distribuição, Gerar mapa de distribuição, Gerar relatórios e Imprimir tornam-se ativos.

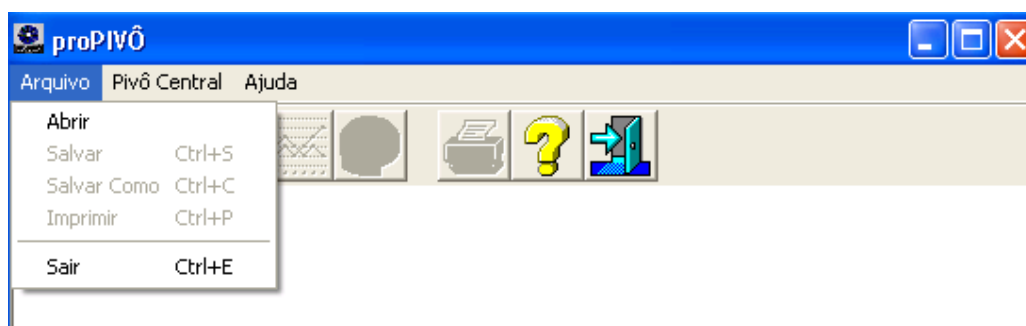


Figura 5 - Tela inicial do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.

A opção arquivo, na barra superior do projeto, apresenta um submenu com os itens: Abrir, Salvar, Salvar como, Imprimir e Sair (Figura 5). Os itens Salvar, Salvar como e Imprimir ficarão habilitados a partir do momento que o projeto esteja com os cálculos disponibilizados.

No submenu do pivô central (Figura 6) é permitido o acesso aos bancos de dados. Nesta opção é possível abrir uma tabela com dados referentes aos experimentos arquivados no banco de dados ou inserir dados de experimentos realizados. Nesse submenu, as opções gráficos e relatórios encontram-se desabilitadas, a sua habilitação ocorrerá no momento que o programa carregar os dados do experimento e calcular seus CUs.

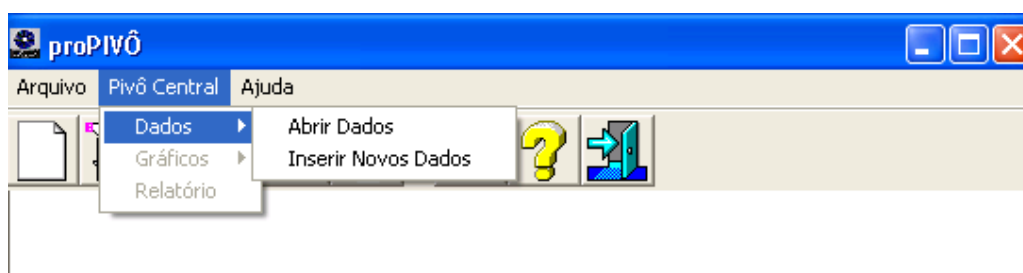


Figura 6 - Submenu do pivô central.

Quando, no submenu pivô central, a opção abrir dados é selecionada, o programa apresenta um formulário para o usuário definir o caminho do arquivo que ele deseja visualizar (Figura 7). No primeiro campo desse formulário é selecionada a unidade de disco, no segundo é selecionado o diretório e na janela à direita o arquivo com extensão *.mdb*, referente aos dados coletados nos experimentos. Na seqüência são dispostos dois botões, um para voltar à tela

inicial e o outro para abrir os dados selecionados que só é habilitado após a seleção do arquivo.

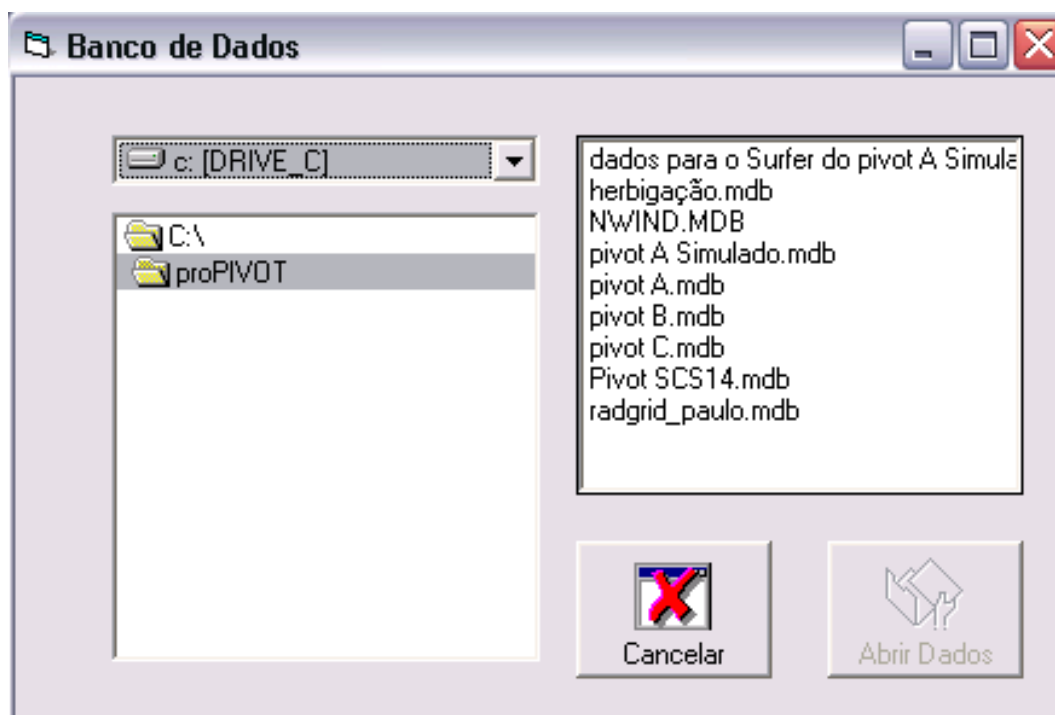


Figura 7 - Exemplo do formulário abrir dados, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.

Após a escolha do arquivo que se deseja trabalhar o programa apresenta uma tela de visualização de dados. Quando o arquivo selecionado se referir a dados de irrigação, o programa abrirá uma tela com valores de lâminas, na sua unidade (Figura 8) e, quando se tratar de dados de quimigação o programa abrirá uma tela de visualização de dados com campos destinados a informações do elemento químico aplicado e sua respectiva unidade (Figura 9). Em ambas as

telas o *layout* é similar, existe um local para que o usuário possa informar a modalidade dos dados: irrigação ou quimigação. Quando são inseridos dados referentes à irrigação o programa adequa-se a essa escolha, representando no campo de dados coletados a lâmina na sua unidade, o mesmo ocorre com os gráficos. Um segundo quadro detalha os dados de ensaio e no terceiro quadro são apresentados os dados do pivô ensaiado. No espaço destinado aos dados do pivô central ha um campo para observações. Os valores e os textos serão listados no relatório do experimento. O preenchimento desse último campo é opcional, mas é fundamental que todos os outros campos deste *frame* sejam preenchidos para que seja possível efetuar os cálculos de eficiência potencial.

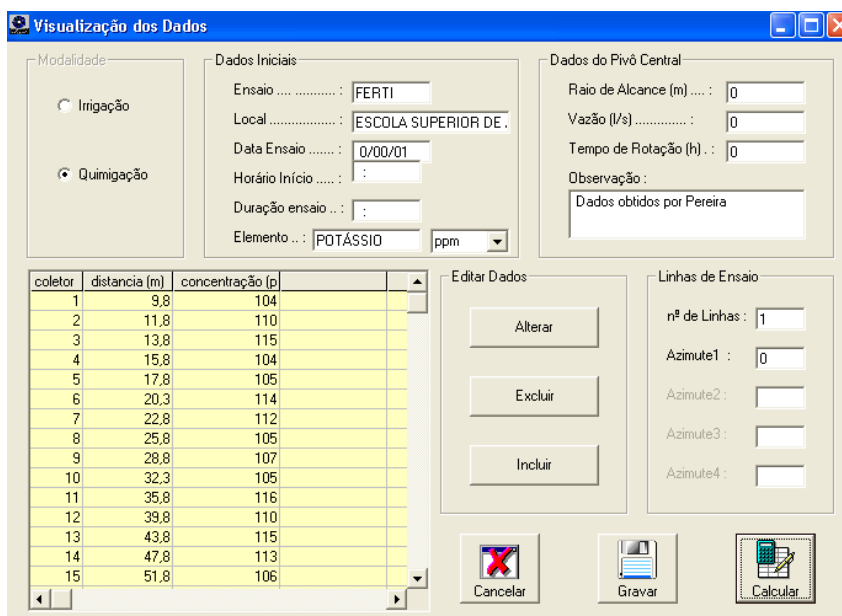
coletor	distancia (m)	lâmina1 (mm)
1	35.4	4.41
2	40	5.34
3	44.6	4.97
4	49.2	5.31
5	53.8	4.76
6	58.4	4.93
7	63	5.52
8	67.6	4.41
9	72.2	4.93
10	76.8	5.14
11	81.4	4.93
12	86	4.83
13	90.6	5.34
14	95.2	4.38
15	99.8	2.86

Figura 8 - Menu de visualização de dados de irrigação, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.

Ao lado da tabela com os dados de ensaios, encontra-se um frame de edição de dados, apresentando quatro opções de interação com os dados listados: alterar, incluir, excluir e gravar.

No canto inferior esquerdo dessa janela encontram-se três botões, um volta para a tela inicial e o outro a opção calcular, pelo qual o programa sai da tela de visualização de dados e vai para a tela de resultados.

Se o arquivo selecionado faz parte de ensaios de herbigação ou fertirrigação, aparecerão nessa tela informações sobre o tipo de produto químico aplicado. A coluna destinada a representar os dados coletados em ensaio apresentará o coeficiente medido no campo e a sua respectiva unidade.



Visualização dos Dados

Modalidade:

☐ Irrigação

☒ Quimigação

Dados Iniciais:

Ensaio : FERTI

Local : ESCOLA SUPERIOR DE.

Data Ensaio : 0/00/01

Horário Início :

Duração ensaio .. :

Elemento ... : POTÁSSIO ppm

Dados do Pivô Central:

Raio de Alcance (m) : 0

Vazão (l/s) : 0

Tempo de Rotação (h) : 0

Observação :
Dados obtidos por Pereira

coletor	distancia (m)	concentração (p)
1	9,8	104
2	11,8	110
3	13,8	115
4	15,8	104
5	17,8	105
6	20,3	114
7	22,8	112
8	25,8	105
9	28,8	107
10	32,3	105
11	35,8	116
12	39,8	110
13	43,8	115
14	47,8	113
15	51,8	106

Editar Dados:

Alterar

Excluir

Incluir

Linhas de Ensaio:

nº de Linhas : 1

Azimute1 : 0

Azimute2 :

Azimute3 :

Azimute4 :

Cancelar Gravar Calcular

Figura 9 - Menu de visualização de dados de quimigação, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.

No *frame* de edição de dados, se a opção alterar for selecionada, após o usuário selecionar a célula que deseja alterar, abri-se um formulário indicando o valor da célula selecionada e um espaço para digitação do valor alterado (Figura 10).

Esse formulário diferencia-se nos módulos de irrigação e herbificação/fertirrigação, pela unidade correspondente. São apresentadas no formulário duas opções: Cancelar, que retorna à tela de visualização dos dados e Alterar, que substitui o valor selecionado pelo valor digitado na célula de alteração.

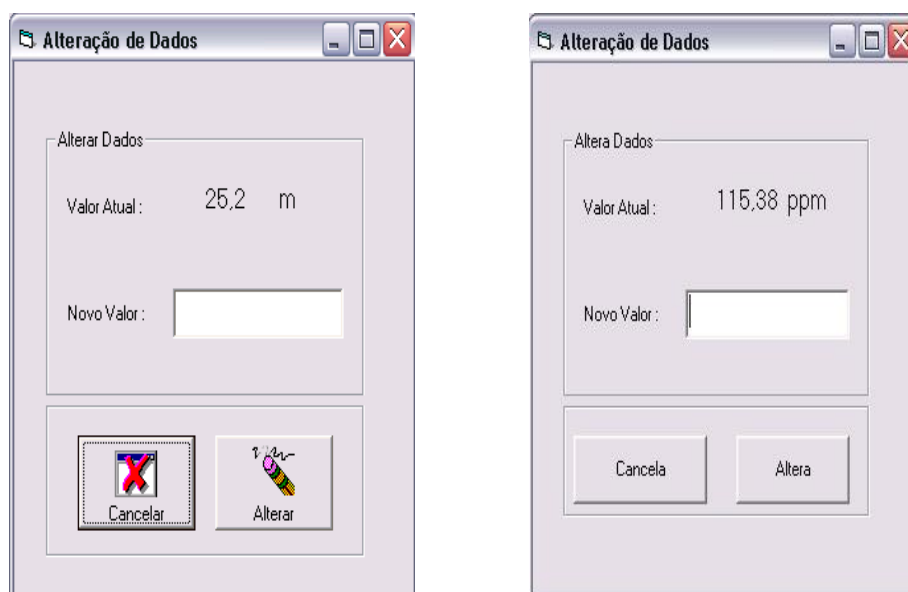


Figura 10 - Subitem para alteração dos dados nos módulos de irrigação e quimigação, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.

Quando uma linha é seleccionada e a opção excluir é acionada o programa mostra uma janela para a confirmação da linha a ser excluída (Figura 11), se o usuário clicar na tecla “não” o programa volta à tela de visualização de dados, mas se a opção for “sim” a linha seleccionada é excluída e a coluna responsável pela ordenação dos coletores é reorganizada.

Visualização dos Dados

Modalidade:

☐ Irrigação

☒ Quimigação

Dados Iniciais:

Ensaio : FERTI

Local : ESCOLA SUPERIOR DE .

Data Ensaio : 0/00/01

Horário Início : :

Duração ensaio .. : :

Elemento .. : POTÁSSIO ppm

Dados do Pivô Central:

Raio de Alcance (m) : 0

Vazão (l/s) : 0

Tempo de Rotação (h) .. : 0

Observação :
Dados obtidos por Pereira

coletor	distancia (m)	concentração (p)
1	9,8	104
2	11,8	110
3	13,8	115
4	15,8	104
5	17,8	105
6	20,3	114
7	22,8	112
8	25,8	105
9	28,8	107
10	32,3	105
11	35,8	116
12	39,8	110
13	43,8	115
14	47,8	113
15	51,8	106

Editar Dados:

Alterar

Linhas de Ensaio:

nº de Linhas : 1

Azimuth1 : 0

Azimuth2 :

Azimuth3 :

Azimuth4 :

Programa ProPivô

Confirma Exclusão da Linha?

Sim Não

Cancelar Gravar Calcular

Figura 11 - Subitem para excluir dados da tabela, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.

Na opção incluir o usuário poderá incluir um valor de lâmina ou concentração em qualquer ponto da tabela (Figura 12). Quando o botão é acionado o programa mostra uma janela para que o usuário informe a posição que foi feita a coleta de dados e o seu resultado. Se selecionar incluir o programa apresenta a nova tabela com o valor da inclusão, posicionado na distância

informada. Se o comando for cancelado pelo botão cancela, será exibida novamente a tela de visualização dos dados.

The figure displays two versions of a software dialog box titled "Incluir Dados".

The left dialog box contains two input fields: "Posição (m)" and "Lâmina (mm) :.....". Below these fields are two buttons: "Cancelar" (with a red 'X' icon) and "Incluir" (with a green checkmark icon).

The right dialog box contains two input fields: "Posição (m)" and "Concentração (ppm):". Below these fields are two buttons: "Cancelar" and "Incluir".

Figura 12 - Exemplo do formulário Incluir dados nos módulos de irrigação e quimigação, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.

Qualquer alteração na tabela de dados pode ser gravada. A opção gravar permite ao usuário salvar a tabela com suas alterações no banco de dados do programa ou em qualquer unidade de disco, para isso o programa abre uma janela, semelhante a da opção abrir dados, na qual é permitida a escolha do caminho que se deseja salvar as alterações (Figura 13).

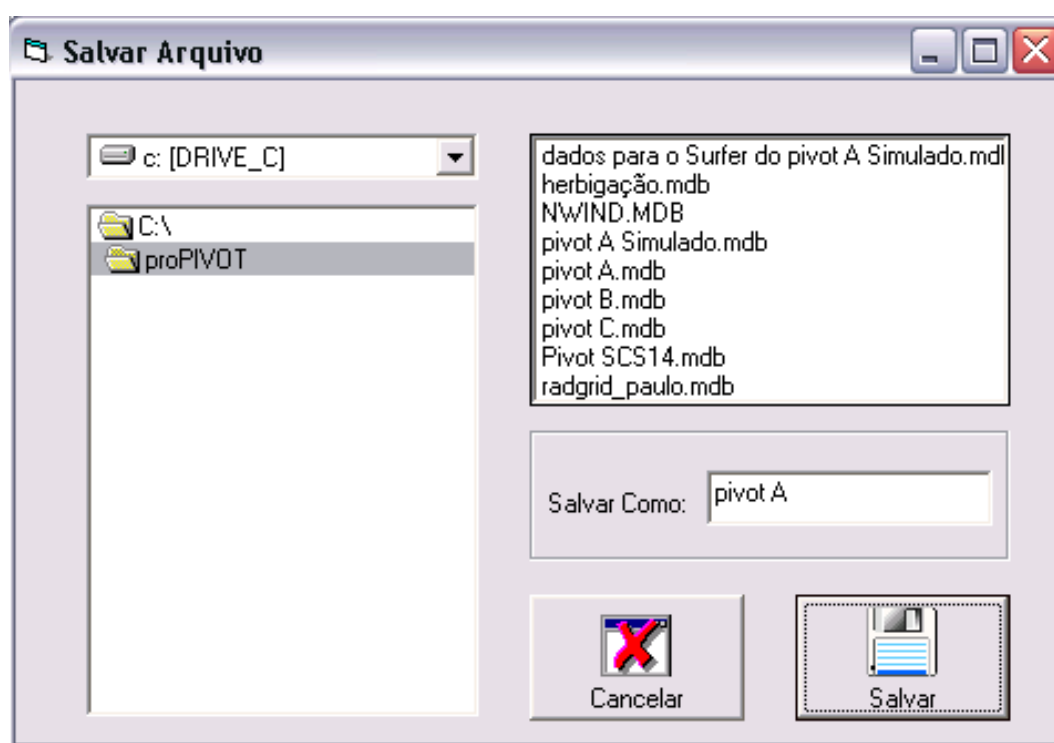


Figura 13 - Exemplo do formulário para salvar arquivo, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.

Após a definição dos dados do ensaio no menu visualizar dados, o programa retorna à tela principal, agora expandida, na qual são informados os valores dos coeficiente de uniformidade, a lâmina média ponderada e, no caso de irrigação, a eficiência, em potencial, de aplicação (Figura 14).

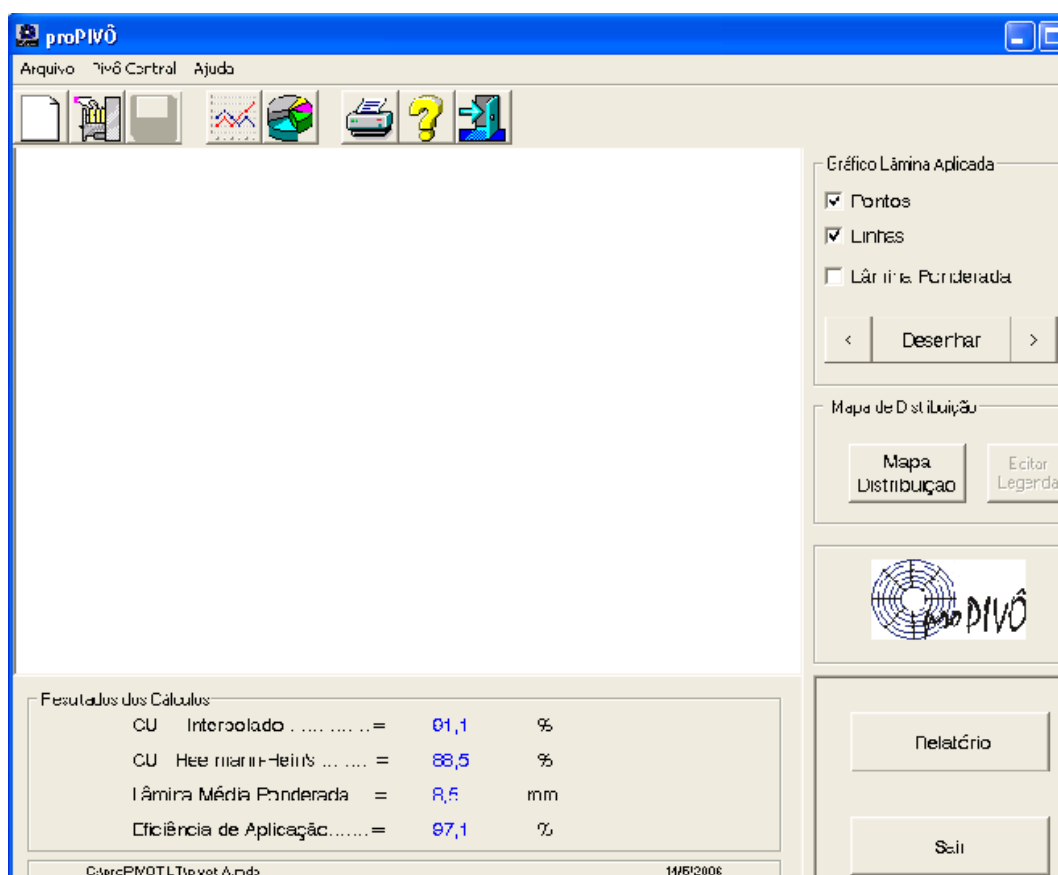


Figura 14 - Menu principal com os resultados representados, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.

Nesse menu é dada a opção de desenhar o gráfico de lâmina aplicada. No gráfico há três modos de visualização dos resultados: o primeiro é representar apenas os pontos referentes às lâminas coletadas, o segundo é representar as linhas, dando o perfil de distribuição de água e a terceira representação traça a lâmina média ponderada, apresentando uma estimativa dos pontos que estão

subirrigados e os pontos que superirrigados (Figura 15). Esse resultado também pode ser conferido pelo ícone disponível nessa tela, que representa o gráfico bidimensional.

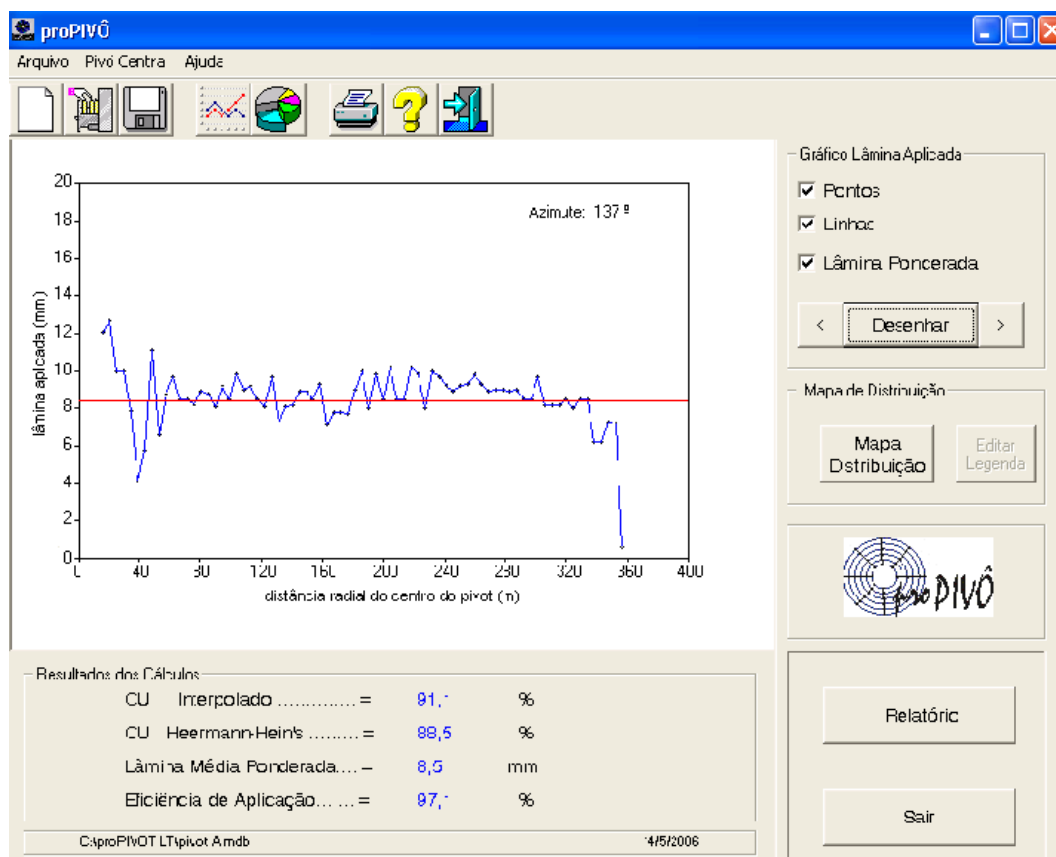


Figura 15 - Representação do gráfico de lâmina aplicada versus distância radial do centro do pivô, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.

O usuário pode imprimir o relatório referente aos dados ensaiados (Figura 16). Nesta etapa é permitido a escolha dos itens que devem ser contemplados no relatório: dados iniciais e dados do pivô, cálculo dos resultados, mapa de distribuição, gráfico de lâmina aplicada, gráfico de distribuição espacial e dados do ensaio. O relatório será gerado com os itens selecionados no campo “Selecione itens do relatório”.

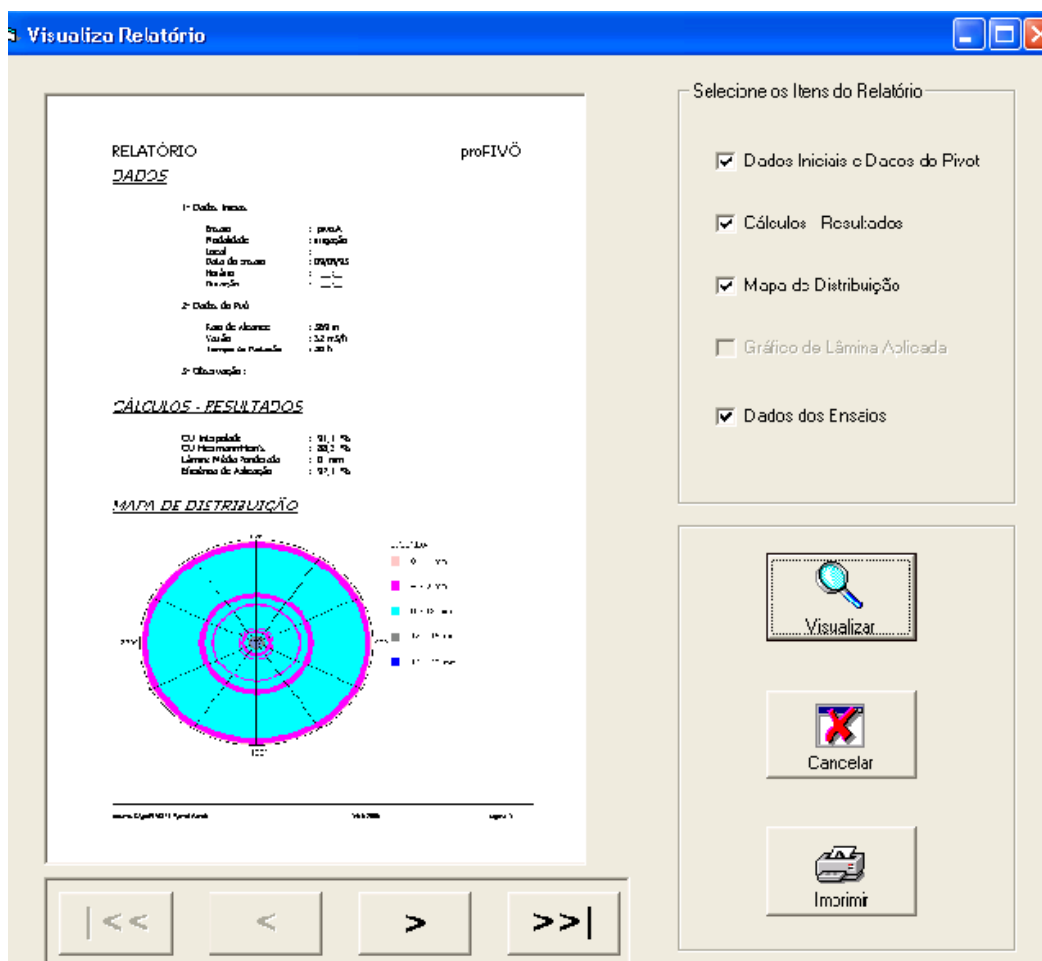


Figura 16 - Relatório gerado pelo programa, do programa proPIVÔ, para avaliação da uniformidade, eficiência de aplicação e quimigação em sistemas de irrigação por pivô central.

O usuário poderá entrar com novos dados de ensaio e inserí-los no banco de dados. Para isso, é disponibilizada uma tela com espaços destinados à entrada de dados do ensaio, do equipamento e dos dados coletados. Para tornar a entrada dos dados mais funcional, o usuário pode informar ao programa que seus dados estão eqüidistantes e o valor do incremento, a partir daí o programa calcula as distâncias automaticamente, sendo necessário ao usuário digitar apenas os valores de lâmina e concentração. No caso de herbigação e fertirrigação é necessário informar o tipo de produto químico aplicado e sua respectiva unidade.

4.2 ANÁLISE DO PROGRAMA COMPUTACIONAL

4.2.1 Equivalências entre os Coeficientes de Uniformidade de Heermann e os Coeficientes de Uniformidade Interpolado de Evans.

Para comparação dos dois coeficientes optou-se pelo *teste t*, que é o teste mais usado para comparação de médias. Foram alterados alguns dados de lâmina na tabela referente a um ensaio de pivô, no banco de dados do programa e observado os valores dos dois coeficientes analisados. Com esses dados (Tabela 1), aplicando o *teste t*, observou que os dois coeficientes tabelados não têm, em média, a mesma característica. Foram aplicados também na amostra os

testes de Tukey e Duncan, ambos obtiveram a mesma conclusão do *teste t*, ou seja, os coeficientes não apresentam as mesmas características.

Foi calculado também o erro relativo dos dois coeficientes, considerando-se o coeficiente de uniformidade de Heermann como o valor mais provável da grandeza e os dados de coeficientes de uniformidade de Evans como valor aproximado, os resultados encontram-se na Tabela 1. O erro relativo médio encontrado foi de $e_r = 2,42\%$. Os valores de CU_i são ligeiramente maiores que os valores encontrados para CU_H . Segundo EVANS, HANS e KROEGER (1995), uma explicação para esse fato pode ser a maneira com que os dados foram interpolados pelo algoritmo proposto, preservando algumas pequenas variações.

Tabela 1 - Erro relativo para diferentes coeficientes de uniformidade, baseados em diferentes lâminas de aplicação

CU_i (%)	CU_H (%)	Erro Relativo (%)
90,7	88,5	2.49
91,0	88,8	2.48
90,9	88,6	2.6
90,5	88,1	2.72
90,8	88,6	2.48
91,8	90,5	1.44
90,4	88,2	2.49
90,7	88,7	2.72
90,5	88,3	2.49
90,3	88,3	2.27

4.2.2 Avaliação do Coeficiente de Uniformidade Acima e Abaixo da Superfície

VILAS BOAS (2000) realizou um experimento em um pivô central, no qual, com auxílio de uma TDR (Reflectomia com domínio de tempo), obteve uma leitura direta do conteúdo de umidade do solo, monitorando a distribuição e absorção de água pela cultura e, por meio de coletores, obteve as lâminas de água aplicadas acima da superfície. Com os dados desse experimento foi determinada a distribuição de água acima e abaixo da superfície.

Os valores do coeficiente de distribuição para as duas situações foram calculados por meio do programa proPIVÔ, alterando-se os espaçamentos entre os coletores e calculando-se também o erro relativo dos dois coeficientes, considerando o coeficiente de uniformidade de Heermann como o valor mais provável da grandeza e os dados de coeficientes de uniformidade de Evans como valor aproximado. Esse procedimento é semelhante ao adotado por EVANS, HANS e KROEGER (1995), para determinar a robustez do seu coeficiente, os resultados encontram-se nas Tabelas 2 e 3.

Tabela 2 - Comparação entre valores de CU_i e CU_H para dados de ensaio de lâminas coletadas

TESTE	ESPAÇAMENTO (m)	E_i (%)	E_H (%)	CU_i	CU_H	LAMINA MÉDIA
1	5,0	2,13	0,00	90,8	88,9	5,7
2	10,0	0,11	1,80	89,0	87,3	5,7
3	15,0	4,61	1,80	93,0	90,5	5,6
4	20,0	1,80	0,67	90,5	88,3	5,7
		2,16 ¹	1,07 ²			

Notas: 1) Média dos erros relativos ao CU_i

2) Média dos erros relativos ao CU_H

Tabela 3 - Comparação entre valores de CU_i e CU_H para dados de umidade do solo

TESTE	ESPAÇAMENTO (m)	E_i (%)	E_H (%)	CU_i	CU_H	UMIDADE MÉDIA
1	5,0	1,07	0	94,7	93,7	0,2
2	10,0	0,96	0,11	94,6	93,8	0,2

3	15,0	1,39	0,21	95,0	93,9	0,2
4	20,0	2,24	0,64	95,8	94,3	0,2
		1,42 ¹	0,24 ²			

Notas: 1) Média dos erros relativos ao CU_i
2) Média dos erros relativos ao CU_H

O resumo da estatísticas dos dados do ensaio realizado por VILLAS BOAS (2000) para as lâminas coletadas e a umidade do solo, estão apresentados na Tabela 4.

Tabela 4 - Estatística descritiva dos dados de lâminas e de umidade do solo

TESTE	LÂMINA (mm)	θ (m ³ m ³)
N	100	100
Média	5,57	0,23
Mínimo	3,85	0,18
Máximo	8,17	0,28
Mediana	5,51	0,22
Desvio Padrão	0,91	0,02
Quartil 1	4,98	0,21
Quartil 3	6,06	0,24
C V (%)	16,31	8,27
Assimetria	0,20	1,50
Curtose	0,31	0,41
P value	0,06	0,10

O maior valor de coeficiente de variação (CV), ocorreu para os dados coletados de lâminas aplicadas (acima da superfície), demonstrando uma maior variação na área amostrada, em ambos os casos obteve-se baixa variabilidade. O intervalo interquartilico, ou seja, a diferença entre quartil 1 e quartil 3, foi de 1,08 e 0,03, para os dados de lâmina coletada e umidade do solo, respectivamente, demonstrando uma maior uniformidade nos dados coletados de umidade do solo (abaixo da superfície). O desvio padrão apresenta um valor menor para os dados amostrais de umidade, verificando-se uma menor dispersão destes dados. Ambos apresentam assimetria positiva e curva platicúrtica, ou seja, a curva de distribuição é mais achatada em relação a uma curva de distribuição normal.

Os dados apresentados nas tabelas 2, 3 e 4, demonstram que os valores de uniformidade de distribuição no interior do solo foram maiores que na superfície, verificando-se assim uma redistribuição de água. Verifica-se também que a lâmina média e a umidade média apresentadas pelo programa, para os dados de ensaio com espaçamentos de 5 metros, são próximos aos valores obtidos, com os mesmos parâmetros, pela estatística descritiva.

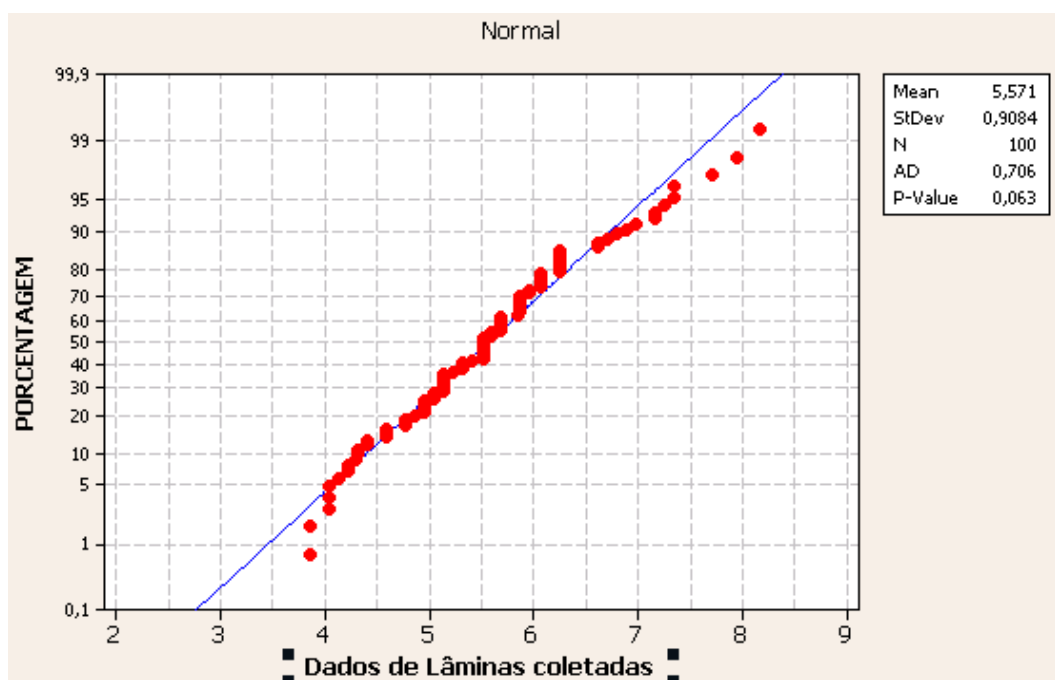


Figura 17 - Gráfico de probabilidade de lâminas aplicadas considerando a distribuição normal.

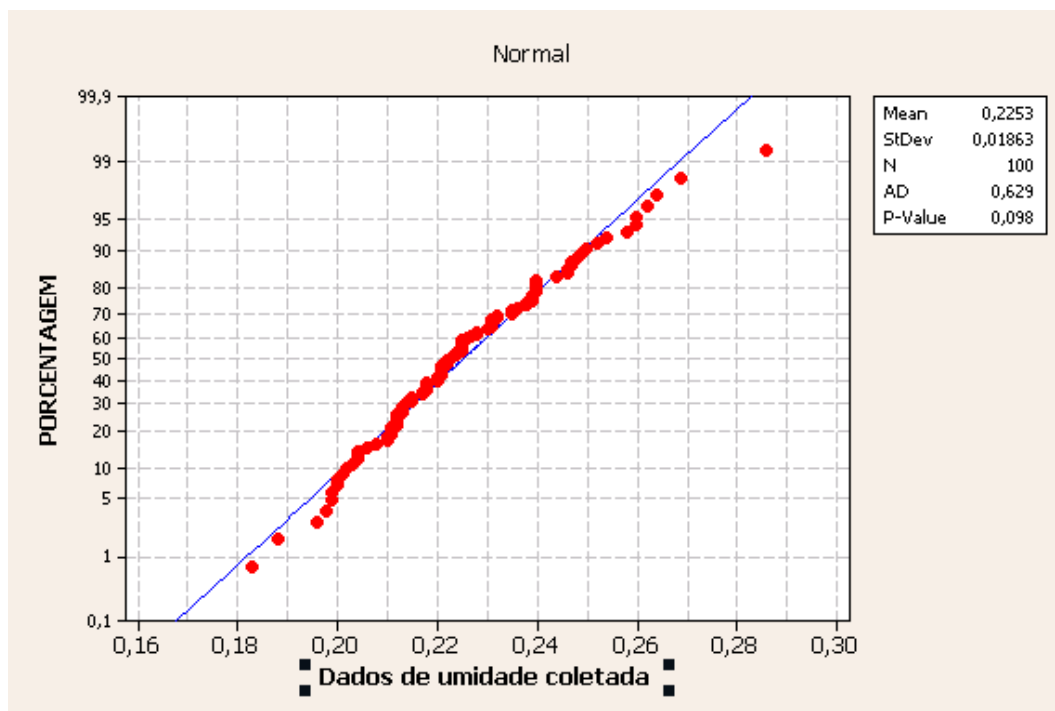


Figura 18 - Gráfico de probabilidade de umidade do solo considerando a distribuição normal.

Pelos dados da Tabela 4, para o valor da estatística de Anderson-Darling, e das figuras 17 e 18, pode-se observar que a distribuição normal é a mais adequada para os dados do estudo.

4.2.3 Avaliação do Coeficiente de Uniformidade de Nutrientes

Para análise dos resultados do programa em relação à aplicação de produtos químicos, utilizaram-se dados de campo de PEREIRA, FRIZZONE e DOURADO NETO (2005), em que os autores estudam a aplicação de potássio em sistemas de irrigação por pivô central.

Os dados de campo dos autores acima citados foram armazenados no banco de dados do programa.

O coeficiente de uniformidade de Evans, CU_i , para a distribuição de potássio foi de 96,6% e a concentração média de 110,8ppm de potássio, que superou a concentração prevista determinada por PEREIRA, FRIZZONE e DOURADO NETO (2005) como 105ppm de potássio, tendo um erro relativo de 5,5%, considerando o valor determinado por PEREIRA, FRIZZONE e DOURADO NETO (2005) como valor mais provável da grandeza.

4.2.4 Gráficos Gerados pelo Programa

O programa apresenta como resultado, gráficos de distribuição linear e mapas de contorno. Para análise destes gráficos foram utilizados dados de ensaios contidos no banco de dados do programa e gerado seus gráficos.

Os mapas de contorno obtidos com o programa foram comparados com os mapas de contorno gerados pelo programa *Catch3D*, para pivô central com os mesmos dados de ensaios. O programa proPIVÔ apresenta cinco níveis de legenda e o programa *Catch3D* apresenta 9 níveis de legenda. Para uma linha de ensaio o programa proPIVÔ apresenta como resultado arcos concentricos como o programa *Catch3D*, mas as faixas apresentadas pelo proPIVÔ são maiores, devido ao menor número de faixas, o que permite uma melhor visualização dos pontos de aplicação.

Os mapas de contorno do ensaio do pivô A, são apresentados na Figura 19, em que se observa que nos mapas gerados pelos dois programas para o ensaio do pivô A, as áreas de maior concentração apresentadas pelo mapa de contorno

do programa proPIVÔ, coincidem com as coroas de aplicação geradas pelo mapa de contorno do programa *Catch3D*.

O mesmo processo foi feito para os ensaios de VILAS BOAS (2000), para dados de lâminas coletadas e umidade do solo.

Os resultados podem ser verificados nas figuras 19 e 20.

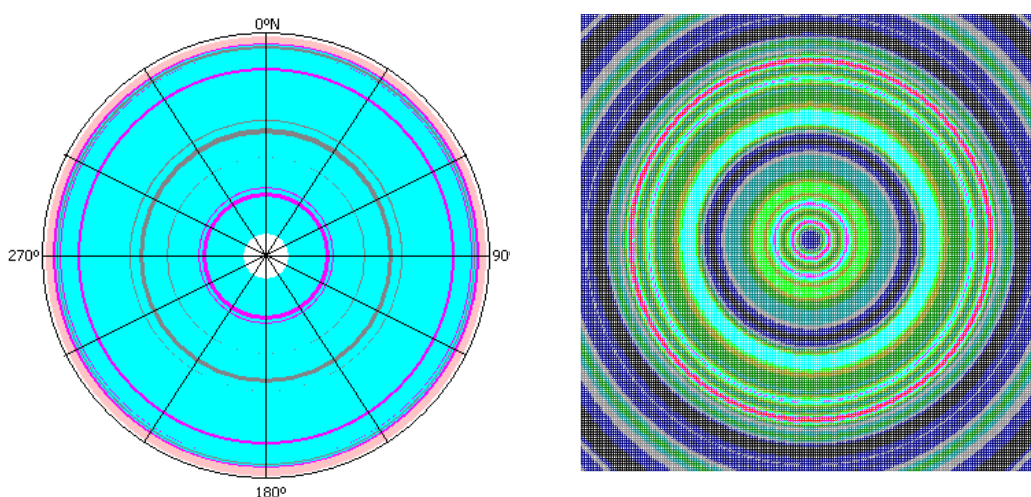
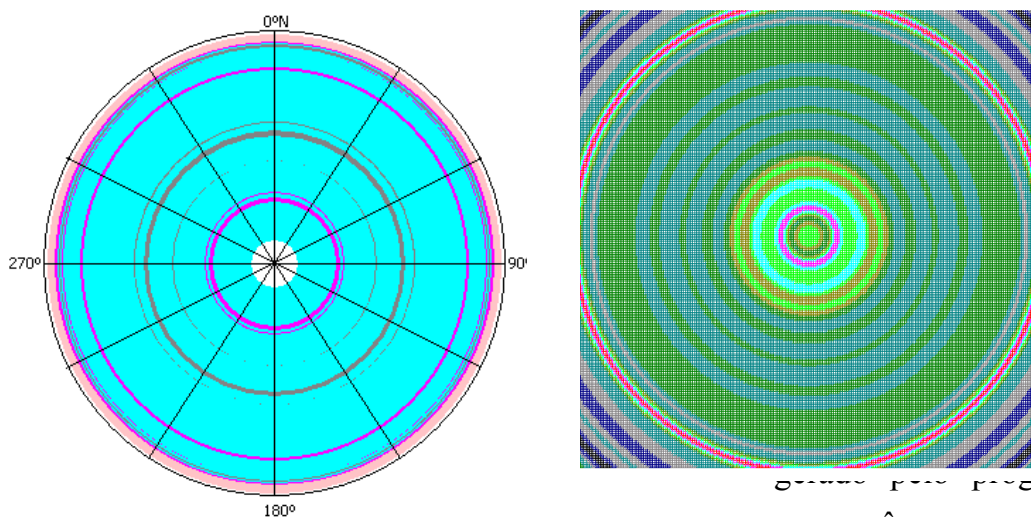


Figura 19 - Mapa de contorno gerado pelo programa proPIVÔ e pelo programa *Catch3D* para dados de lâminas coletadas, respectivamente.



no
programa
proPIVÔ e pelo

programa *Catch3D* para dados de umidades medidas, respectivamente.

O programa permite a entrada de dados para mais de uma linha de ensaio, devendo ser informados os valores de azimuth, ou seja, o ângulo absoluto em relação ao norte. Os gráficos para esses casos também são gerados por interpolação, utilizando-se o sistema de coordenadas polares descrito por EVANS, HANS e KROEGER (1995).

Baseado no artigo de EVANS, HANS e KROEGER (1995), foram selecionados os dados de ensaio do Pivô X (experimento realizado pelo autor acima citado) e o resultado obtido com o mapa de contorno desse pivô foi comparado com os resultados obtidos com o mapa de contorno do mesmo ensaio gerado pelo programa *Catch3D*. Os dois resultados são semelhantes aos obtidos pelos autores.

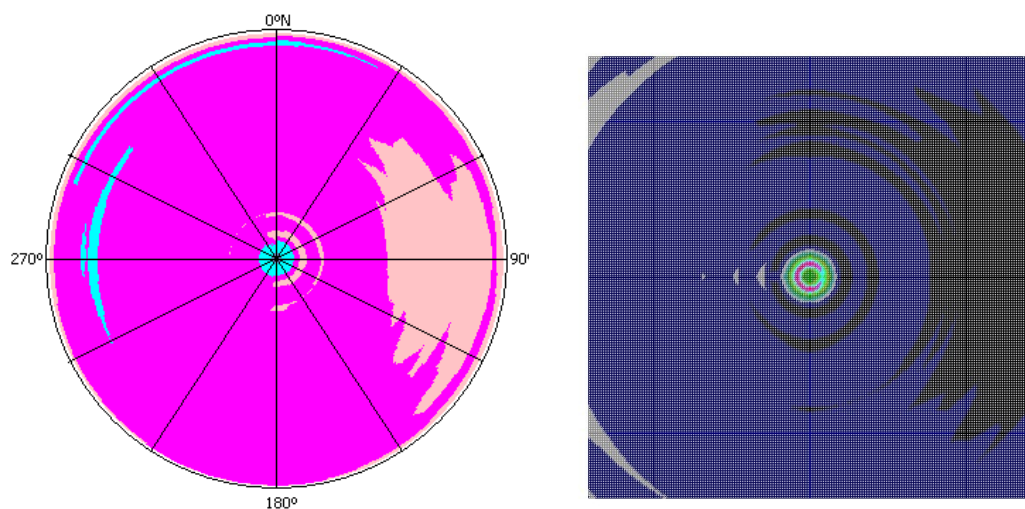


Figura 21 - Mapa de contorno gerado pelo programa proPIVÔ e pelo programa *Catch3D*, para o ensaio do pivô X, com 4 linhas de entrada.

5 CONCLUSÕES

Com base nos dados obtidos pela pesquisa, foi possível concluir que:

- O programa computacional proPIVÔ é capaz de avaliar, de maneira eficiente, a uniformidade e eficiência de distribuição de água em um sistema de irrigação por pivô central, fornecendo de maneira eficiente, mapas de distribuição de aplicação de água e produtos químicos.
- A facilidade em manipular os níveis de legenda, no caso de mapas de contorno, a análise de uniformidade e eficiência de distribuição de água e produtos químicos, além da visualização dos resultados em forma de relatório, tornam o proPIVÔ uma ferramenta de fácil interpretação e didaticamente útil, apresentando os resultados esperados.

REFERÊNCIAS

AGUIAR NETTO, A. O.; COELHO, E. F.; SOUZA, V. F.; BORGES, A. L.; JESUS, T. O. Programa computacional para irrigação e fertirrigação de fruteiras tropicais - FRUTFERT. In: CONGRESSO BRASILEIRO DE ENGENHARIA AGRÍCOLA – CONBEA, 31, 2002. Salvador – BA. **Anais**. Sociedade Brasileira de Engenharia Agrícola, 1 CD-ROM.

ALBUQUERQUE, Emilio P. **Planilha eletrônica para a programação de irrigação em pivôs centrais**. Sete Lagoas: Embrapa milho e sorgo, 2003. Circular técnica 25. 9 p.

ALLEN, R. G.; MERKLEY, G. P. **CATCH-3D - Sprinkler irrigation data analysis e software**. Biological & Irrigation Engineering Department, Utah State University, and the Research & Extension Center of the University of Idaho. Disponível em: <http://www.engineering.usu.edu/bie/software/catch3d.php>. Acesso em: junho de 2005.

ANDRADE JÚNIOR, A. S.; AGUIAR NETO, A. O.; BASTOS, E. A. Dimpivo. Programa computacional para dimensionamento de sistemas de irrigação pivô central. AGROSOFT: FEIRA E CONGRESSO DE INFORMÁTICA APLICADA À AGROPECUÁRIA E À AGROINDÚSTRIA, 1995. Juiz de Fora. **Anais**. 1995. 1 CD-ROM.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – NBR 14244 - Equipamentos de Irrigação mecanizada – Pivô Central e lateral móvel providos de emissores fixos ou rotativos - Determinação da uniformidade de distribuição de água. Dez.98 - 11 páginas.

BERNARDO, S. **Manual de irrigação**. 6 ed. Viçosa: Universidade Federal de Viçosa, 1985.

COSTA, E. F.; VIEIRA, R. F.; VIANA, P. A. **Quimigação**. São Paulo: Embrapa-SPI, 1994.

CHRISTOFFOLETI, P. J.; DOURADO-NETO, D. Manejo integrado de plantas daninhas sob pivô central. In: ZAMBOLIM, L. (Ed) **Manejo integrado fitossanidade: cultivo protegido, pivô central e plantio direto**. Viçosa, MG, 2001. cap.16, p. 625-682.

DECHMI, F.; PLAYÁN, E.; FACI, J. M.; TEJERO, M.; BERCERO, A. Análisis of an irrigation district in northeastern Spain II. Irrigation evaluation, simulation and scheduling. **Agricultural water management**, v. 61, p. 93-109, 2003.

DOURADO NETO, D.; TERUEL, D.A.; REICHARDT, K.; NIELSEN, R. D.; FRIZZONE, J.A.; BACCHI, O. O .S. Princípios de modelagem e simulação: III modelagem do sistema radicular e outros processos subterrâneos, limitações dos modelos e o futuro da modelagem em agricultura. **Scientia Agrícola**. São Paulo, v. 55, n. especial, p. 58-61, 1998.

EVANS, R. G.; HAN, S.; KROEGER, M. W. Spatial distribution and uniformity evaluations for chemigation with center pivots. **Transactions of the ASAE**, v. 38, n. 01, p.85-92, 1995.

FARIAS, M. A.; VIEIRA, J. Irrigação por aspersão: sistemas mais usados no Brasil. **Informe Agropecuário**, São Paulo, v. 12, n. 139, jun.1986.

FREITAS, P. S. L.; MANTOVANI, E. C.; SEDIYAMA, G. C.; COSTA, L.C.; Simulação da produtividade do milho pelo modelo CERES-Maize em função da lâmina e da uniformidade de aplicação de água. **Revista Brasileira de Engenharia Agrícola e Ambiental**, Campina Grande - PB, v. 8, n. 2/3, p. 225-232, 2003.

FRIZZONE, J. A.; DOURADO-NETO, D. **Avaliação de sistemas de irrigação**. Piracicaba–SP: Sbea, 2003.

FRIZZONE, J. A.; PAZ, V. P. S.; MAIA, P. C. S.; CRUCIANIL, D. E. Fertirrigação nitrogenada na cultura do feijoeiro (*phaseolus vulgaris* L.) por sistema de irrigação por aspersão. **Revista Brasileira de Engenharia Agrícola Ambiental**, Campina Grande, v. 2, p. 63-67, 1998.

GARDENAS, A. I.; HOPMANS, J. K.; HANSON, B. R.; SIMUNEK, J. Two-dimensional modeling of nitrate leaching for various fertigation scenarios under micro. **Agricultural Water Management**, California, n. 74, p. 219-242, 2005.

HERGERT, D. **Visual basic 4**. São Paulo: Berkeley Brasil, 1996.

KELLER, J.; BLIESNER, R.D. Sprinkle and Trickle irrigation. Van Nostrand Reinhold, Ney York, 1990, 652p.

KING, B.A.; McCANN, I.R.; EBERLEIN, C.V., STARK, J.C. **Computer Control system for spatially varied water and chemical application studies with continuous-move irrigation systems**. Computers and electronics in agriculture, 1999.

PAPADOPOULOS, I. **Fertirrigation: present situation and future prospects**. In: FOLEGATTI, Marcos V. ed. *Fertirrigação: citrus, flores e hortaliças*. Editora Agropecuária, Guaíba-RS (1999), p.11-154.

PEREIRA, A.S.; FRIZZONE, J.A.; DOURADO-NETO, D. Uniformidade de distribuição de água e potássio aplicados por um sistema acoplado a um pivô central. **XV CONIRD – Congresso nacional de irrigação e drenagem. Teresina-PI, 2005.**

RODRIGUES, L. N.; PRUSKI, F. F.; SILVA, D. D.; MARTINEZ, M. A. GEOPIVO: Modelo para simulação do desempenho de sistemas de irrigação tipo pivô central. **Revista Brasileira de Engenharia Agrícola e Ambiental**, v.5, n.3, p.397-402, 2001.

SILER, B.; SPOTTS, J. **Visual basic 6**. 4 ed. Rio de Janeiro: Campus, 1999.

SILVA, F.C da; BERGAMASCO, A.F.; VENDITE, L.L. Modelos de simulação para análise e apoio à decisão em agrossistemas. **Revista Biociências, Universidade de Taubaté**. Taubaté-SP, v-8, n-1, 2002.

SILVA, J. B.; KARAM, D.; COSTA, E. F. **Herbificação**. Brasília-DF, 1994.

SOUSA, V.F. de; FOLEGATTI, M.V.; FRIZZONE, J.A.; CORRÊA, R.A.de L., ALENCAR, C.M. **Distribuição de fertilizantes em um sistema de fertirrigação por gotejamento**. Campina Grande: Bras. Eng. Agrícola e Ambiental, 2003.

VIANA, Paulo A.; BRITO, Ricardo A. L.; PINTO, Nicésio F. J.A.; PITTA, Gilson V.E.; KARAM, Décio. **Quimigação na cultura do milho**. Sete Lagos-MG: Embrapa, 2002.

VIEIRA, R.F. **Introdução à quimigação**. Brasília: Embrapa-SPI, 1994.

VILAS BOAS, Márcio A. Avaliação do desempenho de sistemas de irrigação por pivô central utilizando técnicas de controle de qualidade. XXIX Congresso Brasileiro de Engenharia Agrícola, 2000. Fortaleza. **Anais**. Sociedade Brasileira de Engenharia Agrícola, 1 CD-ROM.

VILELA, Luiz A.A. **Metodologia para dimensionamento de um sistema de pulverização acoplável a pivô central**. Piracicaba, 149f. Dissertação (Doutorado em Agronomia) Universidade de São Paulo, 2002.

WEBB, A. **Usando Visual Basic 4**. Rio de Janeiro: Campus, 1996.

APÊNDICES

APÊNDICE A

- **FrmAbout**

Option Explicit

```

' Reg Key Security Options...
Const READ_CONTROL = &H20000
Const KEY_QUERY_VALUE = &H1
Const KEY_SET_VALUE = &H2
Const KEY_CREATE_SUB_KEY = &H4
Const KEY_ENUMERATE_SUB_KEYS = &H8
Const KEY_NOTIFY = &H10
Const KEY_CREATE_LINK = &H20
Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE
+ _
    KEY_CREATE_SUB_KEY +
KEY_ENUMERATE_SUB_KEYS + _
    KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL

' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1                ' Unicode nul terminated string
Const REG_DWORD = 4             ' 32-bit number

Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools
Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared
Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias
"RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal
ulOptions As Long, ByVal samDesired As Long, ByRef phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias
"RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String, ByVal
lpReserved As Long, ByRef lpType As Long, ByVal lpData As String, ByRef lpcbData
As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long)
As Long

Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

```

```

Private Sub Form_Load()
    Me.Caption = "About " & App.Title
    lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." &
App.Revision
    lblTitle.Caption = App.Title
    lblDescription.Caption = App.Comments

End Sub

Public Sub StartSysInfo()
    On Error GoTo SysInfoErr

    Dim rc As Long
    Dim SysInfoPath As String

    ' Try To Get System Info Program Path\Name From Registry...
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,
gREGVALSYSINFO, SysInfoPath) Then
        ' Try To Get System Info Program Path Only From Registry...
        ElseIf GetKeyValue(HKEY_LOCAL_MACHINE,
gREGKEYSYSINFOLOC, gREGVALSYSINFOLOC, SysInfoPath) Then
            ' Validate Existence Of Known 32 Bit File Version
            If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
                SysInfoPath = SysInfoPath & "\MSINFO32.EXE"

                ' Error - File Can Not Be Found...
            Else
                GoTo SysInfoErr
            End If
        ' Error - Registry Entry Can Not Be Found...
        Else
            GoTo SysInfoErr
        End If

        Call Shell(SysInfoPath, vbNormalFocus)

    Exit Sub
SysInfoErr:
    MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub

Public Function GetKeyValue(KeyRoot As Long, KeyName As String,
SubKeyRef As String, ByRef KeyVal As String) As Boolean
    Dim i As Long                ' Loop Counter
    Dim rc As Long                ' Return Code
    Dim hKey As Long              ' Handle To An Open Registry
    Dim hDepth As Long            '
Key

```

```

Key          Dim KeyValType As Long          ' Data Type Of A Registry
Registry Key Value
Variable     Dim tmpVal As String            ' Tempory Storage For A
            Dim KeyValSize As Long          ' Size Of Registry Key
            '-----
            ' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
            '-----
            rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) '
Open Registry Key

            If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError      ' Handle
Error...

            tmpVal = String$(1024, 0)          ' Allocate Variable Space
            KeyValSize = 1024                  ' Mark Variable Size

            '-----
            ' Retrieve Registry Key Value...
            '-----
            rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
                                KeyValType, tmpVal, KeyValSize) ' Get/Create Key Value

            If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError      ' Handle
Errors

            If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then      ' Win95 Adds Null
Terminated String...
                tmpVal = Left(tmpVal, KeyValSize - 1)          ' Null Found, Extract
From String
            Else
                ' WinNT Does NOT Null Terminate
String...
                tmpVal = Left(tmpVal, KeyValSize)              ' Null Not Found, Extract
String Only
            End If
            '-----
            ' Determine Key Value Type For Conversion...
            '-----
            Select Case KeyValType
            Case REG_SZ
                KeyVal = tmpVal
                ' Search Data Types...
                ' String Registry Key Data Type
                ' Copy String Value
            Case REG_DWORD
                ' Double Word Registry Key
Data Type
                For i = Len(tmpVal) To 1 Step -1
                    ' Convert Each Bit
                    KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1))) ' Build Value Char.
By Char.
                Next
                KeyVal = Format$("&h" + KeyVal)                  ' Convert Double Word
To String

```

End Select

```
GetKeyValue = True           ' Return Success
rc = RegCloseKey(hKey)       ' Close Registry Key
Exit Function                ' Exit
```

```
GetKeyError: ' Cleanup After An Error Has Occured...
KeyVal = ""   ' Set Return Val To Empty String
GetKeyValue = False   ' Return Failure
rc = RegCloseKey(hKey) ' Close Registry Key
End Function
```

Private Sub picIcon_Click()

End Sub

Private Sub lblDisclaimer_Click()

End Sub

- **FrmAlterar**

```
Private Sub cmdAlterar_Click()
    NovoValor = txtNovoValor.Text
    txtNovoValor.Text = ""
    Unload frmAlterar
    frmLerBD1.Enabled = True
    frmLerBD1.Show
```

End Sub

```
Private Sub cmdCancelar_Click()
    NovoValor = ""
    Unload frmAlterar
    frmLerBD1.Enabled = True
    frmLerBD1.Show
```

End Sub

Private Sub Form_Load()

```
    lbl1.Caption = ValorAtual
    lbl2.Caption = unidade
```

End Sub

```
Function SóNúmeros(Key As Integer) As String
    'selecione os caracteres que desejar
    Const Números$ = ".,0123456789"
    SóNúmeros = Key
```

```

If Key <> 8 Then 'conferindo se é backspace
    'vê se é um caracter permitido
    If InStr(Números$, Chr(Key)) = 0 Then
        SóNúmeros = 0
    End If
End If
End Function

```

```

Private Sub txtNovoValor_KeyPress(KeyAscii As Integer)
    KeyAscii = SóNúmeros(KeyAscii)

```

```

End Sub

```

- **FrmAlteraLegenda**

```

Option Explicit
Dim k As Integer
Dim limites As String

```

```

Private Sub Check1_Click(Index As Integer)

```

```

    If Check1(Index).Value = 0 Then
        cmd1(Index).Enabled = False
        Check1(Index).Caption = ""
        numfaixa = numfaixa - 1
        If numfaixa < 0 Then numfaixa = 0
    End If
    If Check1(Index).Value = 1 Then
        cmd1(Index).Enabled = True
        If LIsup(Index + 1) < LInf(Index + 1) Then LIsup(Index + 1) =
LInf(Index + 1) + 10
        limites = LInf(Index + 1) & " - " & LIsup(Index + 1)
        Check1(Index).Caption = limites
        numfaixa = numfaixa + 1
        If numfaixa > 5 Then numfaixa = 5
    End If

```

```

End Sub

```

```

Private Sub cmd1_Click(Index As Integer)

```

```

    Text1.Text = ""
    Text2.Text = ""
    k = Index + 1
    Picture1.Visible = True
    Label2.Caption = LInf(Index + 1)
    Label5.Caption = LIsup(Index + 1)

```

End Sub

```
Private Sub cmdCancela_Click()
    Picture1.Visible = False
End Sub
```

```
Private Sub cmdConfirma_Click()

If Text1.Text = "" Then Text1.Text = LIinf(k)
If Text2.Text = "" Then Text2.Text = LIsup(k)
```

```
LIinf(k) = CSng(Text1.Text)
LIsup(k) = CSng(Text2.Text)
```

```
LIsup(k - 1) = LIinf(k)
LIinf(k + 1) = LIsup(k)
alteraLegenda
Picture1.Visible = False
```

End Sub

```
Private Sub cmdSair_Click()

    'verifica as faixas
    LIinf(0) = 0
    For k = 1 To numfaixa
        If LIinf(k) <> LIsup(k - 1) Then LIinf(k) = LIsup(k - 1)
    Next k
    If LIsup(numfaixa) < dlamax Then LIsup(numfaixa) = dlamax

    alteraLegenda
    FaixaAlterada = True
    Unload Me
End Sub
```

```
Private Sub Form_Load()
    Picture1.Visible = False
    alteraLegenda
End Sub
```

```
Private Sub alteraLegenda()
For k = 1 To numfaixa
    cmd1(k - 1).Visible = True
    Check1(k - 1).Visible = True
    Check1(k - 1).Value = 1
    If LIsup(k) < LIinf(k) Then LIsup(k) = LIinf(k) + 10
    limites = LIinf(k) & " - " & LIsup(k)
```

```

    Check1(k - 1).Caption = limites
Next k
If numfaixa < 5 Then
    For k = 5 To numfaixa + 1 Step -1
        cmd1(k - 1).Enabled = False
        Check1(k - 1).Value = 0
    Next k
End If

```

```

End Sub
Function SóNúmeros(Key As Integer) As String
    'selecione os caracteres que desejar
    Const Números$ = ".,0123456789"
    SóNúmeros = Key
    If Key <> 8 Then 'conferindo se é backspace
        'vê se é um caracter permitido
        If InStr(Números$, Chr(Key)) = 0 Then
            SóNúmeros = 0
        End If
    End If
End Function

```

```

Private Sub text1_KeyPress(KeyAscii As Integer)
    KeyAscii = SóNúmeros(KeyAscii)

```

```

End Sub
Private Sub text2_KeyPress(KeyAscii As Integer)
    KeyAscii = SóNúmeros(KeyAscii)

```

```

End Sub

```

- **FrmEscolhaBD1**

```

Private Sub cmdAbrirDados_Click()
    strAccessDb = Dir1.path & "\" & File1.FileName
    arquivo = File1.FileName
    'Unload frmEscolhaBD1

```

```

    frmLerBD1.Show
    temporarioDb = "C:\TEMP_PIVOT.MDB"
    ArquivoNovo = True
    geradoMapa = False
    gerado2D(1) = False
    gerado2D(2) = False
    gerado2D(3) = False
    gerado2D(4) = False
End Sub

```

```

Private Sub cmdCancelar_Click()

```

```
Unload Me
End Sub
```

```
Private Sub Dir1_Change()
    File1.path = Dir1.path
End Sub
```

```
Private Sub Drive1_Change()
    Dir1.path = Drive1.Drive
End Sub
```

```
Private Sub File1_Click()
    cmdAbrirDados.Enabled = True
End Sub
```

```
Private Sub File1_DblClick()
    strAccessDb = Dir1.path & "\" & File1.FileName
    temporarioDb = "C:\TEMP_PIVOT.MDB"
    arquivo = File1.FileName
    Unload frmEscolhaBD1
    frmLerBD1.Show
    ArquivoNovo = True
    geradoMapa = False
    gerado2D(1) = False: gerado2D(2) = False: gerado2D(3) = False:
gerado2D(4) = False
    gerado3D = False
```

```
End Sub
```

```
Private Sub Form_Load()
    Dir1.path = App.path
End Sub
```

- **FrmInclui**

```
Private Sub cmdCancelar_Click()
    frmLerBD1.Enabled = True
    Unload frmInclui
    frmLerBD1.Show
End Sub
```

```
Private Sub cmdIncluir_Click()
    iDistancia = txtPosição.Text
    ilamina = txtLamina.Text
    Unload frmInclui
```

```
End Sub
```

```
Function SóNúmeros(Key As Integer) As String
```

```

'selecione os caracteres que desejar
Const Números$ = ".,0123456789"
SóNúmeros = Key
If Key <> 8 Then 'conferindo se é backspace
    'vê se é um caracter permitido
    If InStr(Números$, Chr(Key)) = 0 Then
        SóNúmeros = 0
    End If
End If
End Function

```

```

Private Sub Form_Load()
If modalidade = "herbigação" Then
    lbl3.Caption = "Concentração" & " (" & unElemento & "):"

End If
End Sub

```

```

Private Sub txtPosição_KeyPress(KeyAscii As Integer)
    KeyAscii = SóNúmeros(KeyAscii)

End Sub

```

```

Private Sub txtLamina_KeyPress(KeyAscii As Integer)
    KeyAscii = SóNúmeros(KeyAscii)

End Sub

```

- **FrmInserirDados**

```

Option Explicit
Dim i, j As Integer
Dim numero As Single

Private Sub cboElemento_click()
    MSFlexGrid1.TextMatrix(0, 2) = " concentração" & "(" &
cboElemento.Text & ")"

End Sub

Private Sub cmdCancelar_Click()
    Unload frmInserirDados
    frmPrincipal1.Show

End Sub

Private Sub cmdAtiva_Click(Index As Integer)
    frmInserir.Enabled = True

```

```

txtLamina1.SetFocus
txtDistancia.Text = txtValorInicial.Text
End Sub

Private Sub cmdGravar_Click()

' verifica os azimutes

If Combo1.Text = "1 linha" Then
    If txtAz1.Text = "" Then txtAz1.Text = "0"
    txtAz2.Text = "0": txtAz3.Text = "0": txtAz4.Text = "0"

End If
If Combo1.Text = "2 linhas" Then
    If txtAz1.Text = "" Or txtAz2.Text = "" Then
        MsgBox ("Faltou o valor do Azimute !")
        Exit Sub
    End If
    If CSng(txtAz2.Text) <= CSng(txtAz1.Text) Then
        MsgBox ("O Azimute2 tem que ser maior que o azimute1 !")
        Exit Sub
    End If
    txtAz3.Text = "0": txtAz4.Text = "0"
End If

If Combo1.Text = "3 linhas" Then
    If txtAz1.Text = "" Or txtAz2.Text = "" Or txtAz3.Text = "" Then
        MsgBox ("Faltou o valor do Azimute !")
        Exit Sub
    End If
    If CSng(txtAz3.Text) <= CSng(txtAz2.Text) Or CSng(txtAz2.Text) <=
CSng(txtAz1.Text) Then
        MsgBox ("Os Azimutes tem que estar em ordem crescente !")
        Exit Sub
    End If
    txtAz4.Text = "0"
End If

If Combo1.Text = "4 linhas" Then
    If txtAz1.Text = "" Or txtAz2.Text = "" Or txtAz3.Text = "" Or txtAz4.Text
= "" Then
        MsgBox ("Faltou o valor do Azimute !")
        Exit Sub
    End If
    If CSng(txtAz4.Text) <= CSng(txtAz3.Text) Or CSng(txtAz3.Text) <=
CSng(txtAz2.Text) Or CSng(txtAz2.Text) <= CSng(txtAz1.Text) Then
        MsgBox ("Os Azimutes tem que estar em ordem crescente !")
        Exit Sub
    End If
End If

```

```
formulario = "inserir"
```

```
PreencheTabela
frmSalvar.Show
frmInserirDados.Hide
DadosCarregados = True
cmdSair.Enabled = True
End Sub
```

```
Private Sub cmdInformaValores_Click()
frmInserir.Enabled = True
txtLamina1.SetFocus
txtDistancia.Text = txtValorInicial.Text
End Sub
```

```
Private Sub cmdInserir_Click()
```

```
frmDistancia.Enabled = False
i = i + 1
If i = 1 Then
```

```
    MSFlexGrid1.TextMatrix(1, 0) = i
    MSFlexGrid1.TextMatrix(1, 1) = txtDistancia.Text
    MSFlexGrid1.TextMatrix(1, 2) = txtLamina1.Text
    MSFlexGrid1.TextMatrix(1, 3) = txtLamina2.Text
    MSFlexGrid1.TextMatrix(1, 4) = txtLamina3.Text
    MSFlexGrid1.TextMatrix(1, 5) = txtLamina4.Text
```

```
Else
    MSFlexGrid1.AddItem i & Chr(9) & txtDistancia.Text & Chr(9) &
txtLamina1.Text & Chr(9) &
    & txtLamina2.Text & Chr(9) & txtLamina3.Text & Chr(9) &
txtLamina4.Text
End If
If OptSim.Value = False Then
    numero = Val(txtDistancia.Text) + Val(txtIncremento.Text)
Else
    numero = CSng(txtValorInicial.Text) + i * (CSng(txtIncremento.Text))
End If
    txtDistancia.Text = numero
    txtLamina1.Text = ""
    txtLamina2.Text = ""
    txtLamina3.Text = ""
    txtLamina4.Text = ""
    If OptSim.Value = False Then
        txtDistancia = ""
        txtDistancia.SetFocus
    Else
        txtLamina1.SetFocus
```

End If

End Sub

Private Sub cmdSair_Click()
Dim i As Integer

' descarrega o formulário
Unload frmInserirDados
DadosCarregados = True
' chama o formulário principal
frmPrincipal1.Height = 9360
frmPrincipal1.Width = 11415
frmPrincipal1.Show

DadosCarregados = True

If optIrrigacao.Value = True Then
 modalidade = "irrigação"
Else
 modalidade = "herbificação"
End If

End Sub

Private Sub Combo1_Click()

If Combo1.Text = "1 linha" Then
 txtLamina2.Enabled = False
 txtLamina3.Enabled = False
 txtLamina4.Enabled = False
 lblLamina2.Enabled = False
 lblLamina3.Enabled = False
 lblLamina4.Enabled = False
 lblAz2.Enabled = False
 lblAz3.Enabled = False
 lblAz4.Enabled = False
 txtAz2.Enabled = False
 txtAz3.Enabled = False
 txtAz4.Enabled = False
End If

If Combo1.Text = "2 linhas" Then
 txtLamina2.Enabled = True
 txtLamina3.Enabled = False
 txtLamina4.Enabled = False
 lblLamina2.Enabled = True

```

        lblLamina3.Enabled = False
        lblLamina4.Enabled = False
        lblAz2.Enabled = True
        lblAz3.Enabled = False
        lblAz4.Enabled = False
        txtAz2.Enabled = True
        txtAz3.Enabled = False
        txtAz4.Enabled = False
    End If

    If Combo1.Text = "3 linhas" Then
        txtLamina2.Enabled = True
        txtLamina3.Enabled = True
        txtLamina4.Enabled = False
        lblLamina2.Enabled = True
        lblLamina3.Enabled = True
        lblLamina4.Enabled = False
        lblAz2.Enabled = True
        lblAz3.Enabled = True
        lblAz4.Enabled = False
        txtAz2.Enabled = True
        txtAz3.Enabled = True
        txtAz4.Enabled = False
    End If

    If Combo1.Text = "4 linhas" Then
        txtLamina2.Enabled = True
        txtLamina3.Enabled = True
        txtLamina4.Enabled = True
        lblLamina2.Enabled = True
        lblLamina3.Enabled = True
        lblLamina4.Enabled = True
        lblAz2.Enabled = True
        lblAz3.Enabled = True
        lblAz4.Enabled = True
        txtAz2.Enabled = True
        txtAz3.Enabled = True
        txtAz4.Enabled = True
    End If
End Sub

```

```

Private Sub Form_Load()
    i = 0
    txtValorInicial.Visible = False
    txtIncremento.Visible = False
    Label2.Enabled = False
    Label3.Enabled = False
    txtLamina2.Enabled = False
    txtLamina3.Enabled = False
    txtLamina4.Enabled = False

```

```

lblLamina2.Enabled = False
lblLamina3.Enabled = False
lblLamina4.Enabled = False
cmdInformaValores.Enabled = False
    lblAz2.Enabled = False
    lblAz3.Enabled = False
    lblAz4.Enabled = False
    txtAz2.Enabled = False
    txtAz3.Enabled = False
    txtAz4.Enabled = False
    FormataTabela
End Sub

```

```

Private Sub Label15_Click()

```

```

End Sub

```

```

Private Sub optHerbigacao_Click()

```

```

    lblElemento.Visible = True
    txtElemento.Visible = True
    cboElemento.Visible = True

```

```

    MSFlexGrid1.TextMatrix(0, 2) = " concentração" & "(" & cboElemento.Text
& ")"

```

```

End Sub

```

```

Private Sub optIrrigacao_Click()

```

```

    MSFlexGrid1.TextMatrix(0, 2) = " lâmina (mm)"
    'lblLamina.Caption = "Lâmina (mm):"

```

```

    lblElemento.Visible = False
    txtElemento.Visible = False
    cboElemento.Visible = False

```

```

End Sub

```

```

Private Sub OptNão_Click()

```

```

    txtValorInicial.Visible = False
    txtIncremento.Visible = False
    txtDistancia.Locked = False
    txtDistancia.BackColor = &H80000005
    cmdInformaValores = False
    frmInserir.Enabled = True

```

```

End Sub

```

```

Private Sub OptSim_Click()

```

```

    frmInserir.Enabled = False
    txtValorInicial.Visible = True

```

```

txtIncremento.Visible = True
Label2.Enabled = True
Label3.Enabled = True
txtValorInicial.SetFocus
txtDistancia.Locked = True
txtDistancia.BackColor = &H8000000F
cmdInformaValores.Enabled = True

End Sub

Private Sub FormataTabela()

Dim largura_coluna() As Single
Dim largura_campo As Single

' define linhas fixas igual a uma e não usa colunas fixas
MSFlexGrid1.Rows = 10
MSFlexGrid1.FixedRows = 1
MSFlexGrid1.FixedCols = 0

' define o numero de linhas e colunas e cria uma matrix
MSFlexGrid1.Rows = 1
MSFlexGrid1.Cols = 6
' exibe os cabeçalhos das colunas

MSFlexGrid1.TextMatrix(0, 0) = " coletor"
MSFlexGrid1.TextMatrix(0, 1) = " distancia (m)"
If optHerbigacao.Value = True Then
    MSFlexGrid1.TextMatrix(0, 2) = " concentração1(ppm)"
    MSFlexGrid1.TextMatrix(0, 3) = " concentração2(ppm)"
    MSFlexGrid1.TextMatrix(0, 4) = " concentração3(ppm)"
    MSFlexGrid1.TextMatrix(0, 5) = " concentração4(ppm)"
Else
    MSFlexGrid1.TextMatrix(0, 2) = " lâmina1(mm)"
    MSFlexGrid1.TextMatrix(0, 3) = " lâmina2(mm)"
    MSFlexGrid1.TextMatrix(0, 4) = " lâmina3(mm)"
    MSFlexGrid1.TextMatrix(0, 5) = " lâmina4(mm)"
End If
' exibe o valor de cada linha
linha = 1

MSFlexGrid1.Rows = MSFlexGrid1.Rows + 1

' define a largura das colunas do grid
MSFlexGrid1.ColWidth(0) = 650
MSFlexGrid1.ColWidth(1) = 1100
MSFlexGrid1.ColWidth(2) = 1100
End Sub

```

```

'Private Sub txtLamina_KeyPress(KeyAscii As Integer)

    'If KeyAscii = vbKeyReturn Then
        'j = j + 1
        ' If j = 1 Then

            ' MSFlexGrid1.TextMatrix(1, 0) = j
            ' MSFlexGrid1.TextMatrix(1, 1) = txtDistancia.Text
            ' MSFlexGrid1.TextMatrix(1, 2) = txtLamina.Text

        ' Else
            ' MSFlexGrid1.AddItem j & Chr(9) & txtDistancia.Text & Chr(9) &
txtLamina.Text
        ' End If
        ' numero = Val(txtDistancia.Text) + Val(txtIncremento.Text)
        ' numero = CSng(txtValorInicial.Text) + j * (CSng(txtIncremento.Text))
        ' txtDistancia.Text = numero
        ' txtLamina.Text = ""
        ' txtLamina.SetFocus
        ' End If

'End Sub

Private Sub txtLamina_click()
    If OptSim.Value = True Then txtDistancia.Text = txtValorInicial.Text

End Sub
Private Function PreencheTabela()
    Dim BD As Database
    Dim TBTabela As Recordset
    Dim TBDados As Recordset
    Dim linha_i As Integer
    Dim nlinha As Integer

    Set BD = OpenDatabase(temporarioDb)
    Set TBTabela = BD.OpenRecordset("Tabela", dbOpenTable)
    Set TBDados = BD.OpenRecordset("Dados", dbOpenTable)

    nlinha = MSFlexGrid1.Rows

    For linha_i = 1 To nlinha - 1

        TBTabela.AddNew
        TBTabela("coletor") = MSFlexGrid1.TextMatrix(linha_i, 0)
        TBTabela("distancia") = MSFlexGrid1.TextMatrix(linha_i, 1)
    
```

```

        If MSFlexGrid1.TextMatrix(linha_i, 2) = "" Then
MSFlexGrid1.TextMatrix(linha_i, 2) = "0"
        TBTabela("lamina1") = MSFlexGrid1.TextMatrix(linha_i, 2)
        If MSFlexGrid1.TextMatrix(linha_i, 3) = "" Then
MSFlexGrid1.TextMatrix(linha_i, 3) = "0"
        TBTabela("lamina2") = MSFlexGrid1.TextMatrix(linha_i, 3)
        If MSFlexGrid1.TextMatrix(linha_i, 4) = "" Then
MSFlexGrid1.TextMatrix(linha_i, 4) = "0"
        TBTabela("lamina3") = MSFlexGrid1.TextMatrix(linha_i, 4)
        If MSFlexGrid1.TextMatrix(linha_i, 5) = "" Then
MSFlexGrid1.TextMatrix(linha_i, 5) = "0"
        TBTabela("lamina4") = MSFlexGrid1.TextMatrix(linha_i, 5)

```

```

TBTabela.Update

```

```

Next linha_i

```

```

TBDados.AddNew

```

```

If optHerbigacao = True Then
    TBDados("modalidade") = "herbificação"
Else
    TBDados("modalidade") = "irrigação"
End If
If Text1.Text = "" Then
    Text1.Text = " "
End If
If Text2.Text = "" Then
    Text2.Text = " "
End If
If Text6.Text = "" Then
    Text6.Text = " "
End If
If Text7.Text = "" Then
    Text7.Text = " "
End If
If Text8.Text = "" Then
    Text8.Text = " "
End If
If Text9.Text = "" Then
    Text9.Text = " "
End If
If mskData.Text = "" Then
    mskData.Text = "    "
End If
If mskHorario.Text = "" Then
    mskHorario.Text = "    "
End If
If mskDuracao.Text = "" Then
    mskDuracao.Text = "    "

```

```

End If
If txtElemento.Text = "" Then
    txtElemento.Text = " "
End If

TBDados("ensaio") = Text1.Text
TBDados("local") = Text2.Text
TBDados("dataensaio") = mskData.Text
TBDados("horainicio") = mskHorario.Text
TBDados("duracao") = mskDuracao.Text
TBDados("raio") = Text6.Text
TBDados("vazao") = Text7.Text
TBDados("trot") = Text8.Text
TBDados("obs") = Text9.Text
TBDados("elemento") = txtElemento.Text
TBDados("unelemento") = cboElemento.Text
TBDados("nlinhas") = CSng(Left$(Combo1.Text, 1))
TBDados("azimute1") = CSng(txtAz1.Text)
TBDados("azimute2") = CSng(txtAz2.Text)
TBDados("azimute3") = CSng(txtAz3.Text)
TBDados("azimute4") = CSng(txtAz4.Text)

TBDados.Update
TBDados.Close

TBTabela.Close
BD.Close

End Function

Function SóNúmeros(Key As Integer) As String
    'selecione os caracteres que desejar
    Const Números$ = ".,0123456789"
    SóNúmeros = Key
    If Key <> 8 Then 'conferindo se é backspace
        'vê se é um caracter permitido
        If InStr(Números$, Chr(Key)) = 0 Then
            SóNúmeros = 0
        End If
    End If
End Function

Private Sub Text6_KeyPress(KeyAscii As Integer)
    KeyAscii = SóNúmeros(KeyAscii)
End Sub

Private Sub Text7_KeyPress(KeyAscii As Integer)

```

```
KeyAscii = SóNúmeros(KeyAscii)
```

```
End Sub
```

```
Private Sub Text8_KeyPress(KeyAscii As Integer)
```

```
KeyAscii = SóNúmeros(KeyAscii)
```

```
End Sub
```

```
Private Sub txtValorInicial_KeyPress(KeyAscii As Integer)
```

```
KeyAscii = SóNúmeros(KeyAscii)
```

```
End Sub
```

```
Private Sub txtIncremento_KeyPress(KeyAscii As Integer)
```

```
KeyAscii = SóNúmeros(KeyAscii)
```

```
End Sub
```

- **FrmLerBD1**

```
Option Explicit
```

```
Dim BD As Database
```

```
Dim TBDados As Recordset
```

```
Dim TBTabela As Recordset
```

```
Dim i As Integer
```

```
Dim distancia(1000) As Single
```

```
Dim lamina(1000, 1000) As Single
```

```
Dim nlinha, ncolunas As Integer
```

```
Dim u As Integer
```

```
Dim alinha As Integer
```

```
Dim registro As Integer
```

```
Dim txtAzimute, txtAz As String
```

```
Dim ordem1, ordem2, ordem3, ordem4 As Integer
```

```
Private Sub cmdAlterar_Click()
```

```
If MSFlexGrid1.Col <> 0 Then
```

```
ValorAtual = MSFlexGrid1.Text
```

```
If optIrigacao.Value = True Then
```

```
If MSFlexGrid1.Col = 1 Then
```

```
unidade = "m"
```

```
Else
```

```

        unidade = "mm"
    End If
Else
    If MSFlexGrid1.Col = 1 Then
        unidade = "m"
    Else
        unidade = cboElemento.Text
    End If

End If

    alinha = MSFlexGrid1.Row
    coluna = MSFlexGrid1.Col
    frmLerBD1.Enabled = False
    frmAltera.Show
Else
End If

End Sub

Private Sub cmdExcluir_Click()
If MSFlexGrid1.Row <> 0 Then

    If MsgBox("Confirma Exclusão da Linha?", vbYesNo) = vbYes Then

        nlinha = MSFlexGrid1.Row

        ExcluiLinha

    Else

        End If
    End If
End Sub

Private Sub cmdGravar_Click()
If txtNlinhas > 4 Then
    MsgBox ("O nº de linhas não pode ser maior que 4!")
    Exit Sub
End If

'valores do azimuth

azimute(1) = Val(txtAz1.Text): azimute(2) = Val(txtAz2.Text)
azimute(3) = Val(txtAz3.Text): azimute(4) = Val(txtAz4.Text)

' ordenação crescente de azimuth

```

```

        If azimuth(1) > 360 Or azimuth(2) > 360 Or azimuth(3) > 360 Or azimuth(4)
> 360 Then
            MsgBox ("O azimuth tem que ser menor que 360°")
            Exit Sub
        End If

        If nlinhas = 2 Then
            If azimuth(1) < azimuth(2) Then
                Else

                MsgBox ("Os azimutes tem que estar em ordem crescente")
                Exit Sub
            End If

            End If
        If nlinhas = 3 Then
            If azimuth(1) < azimuth(2) < azimuth(3) Then
                Else

                MsgBox ("Os azimutes tem que estar em ordem crescente")
                Exit Sub
            End If

            End If
        If nlinhas = 4 Then
            If azimuth(1) < azimuth(2) And azimuth(2) < azimuth(3) And azimuth(3)
< azimuth(4) Then
                Else

                MsgBox ("Os azimutes tem que estar em ordem crescente")
                Exit Sub
            End If

            End If

        temporarioDb = "C:\TEMP_PIVOT.MDB"

        CriarTabela (Dados)
        PreencheTabela
        frmLerBD1.Enabled = False
        frmSalvar.Show

    End Sub

    Private Sub cmdIncluir_Click()
        unElemento = cboElemento

```

```

frmLerBD1.Enabled = False

frmInclui.Show
End Sub

Private Sub cmdSair_Click()
    Dim i As Integer

    If optIrigacao.Value = True Then
        modalidade = "irrigação"
    Else
        modalidade = "herbificação"
    End If

    Elemento = txtElemento.Text
    unElemento = cboElemento

    'dados para calculo eficiencia
    If Text6.Text <> Null And Text7.Text <> Null And Text8.Text <> Null
Then
        raio_Ef = CSng(Text6.Text)
        q_Ef = CSng(Text7.Text)
        tr_Ef = CSng(Text8.Text)
    End If

    ' descarrega o formulário
    Unload frmLerBD1
    DadosCarregados = True
    FaixaAlterada = False

    ' chama o formulário principal

    frmPrincipal1.Height = 9360
    frmPrincipal1.Width = 11415
    frmPrincipal1.Show

End Sub

Private Sub cmdVoltar_Click()
    Unload frmLerBD1
    frmPrincipal1.Show
End Sub

Private Sub form_Activate()
    If NovoValor <> "" Then
        MSFlexGrid1.TextMatrix(alinha, coluna) = NovoValor
    End If
    If iDistancia <> 0 Then
        OrdenaMatrix
    End If

```

End If

End Sub

Private Sub Form_GotFocus()

 Unload frmAlterar

End Sub

Private Sub Form_Load()

 dlamax = 0: dlamin = 0

 FaixaAlterada = False

 salvaGrafico = False

 raio_Ef = 0: q_Ef = 0: tr_Ef = 0

 laminaponderada = 0: Epa = 0: CUi = 0: CUh = 0

 geradoMapa = False

 gerado2D(1) = False

 gerado2D(2) = False

 gerado2D(3) = False

 gerado2D(4) = False

 u = 0

 iDistancia = 0

 ilamina = 0

 Call encheGrid("strAccess.mdb", "Select * from Tabela ")

 frmPrincipal1.Hide

 fraTipo.Enabled = False

End Sub

Private Function PreencheTabela()

 Dim linha_i As Integer

 Set BD = OpenDatabase(temporarioDb)

 Set TBTabela = BD.OpenRecordset("Tabela", dbOpenTable)

 Set TBDados = BD.OpenRecordset("Dados", dbOpenTable)

 nlinha = MSFlexGrid1.Rows

 If u <> 0 Then

 nlinha = nlinha - u

End If

For linha_i = 1 To nlinha - 1

TBTabela.AddNew

```

TBTabela("coletor") = MSFlexGrid1.TextMatrix(linha_i, 0)
TBTabela("distancia") = MSFlexGrid1.TextMatrix(linha_i, 1)
TBTabela("lamina1") = MSFlexGrid1.TextMatrix(linha_i, 2)
If nlinhas = 4 Then
    TBTabela("lamina2") = MSFlexGrid1.TextMatrix(linha_i, 3)
    TBTabela("lamina3") = MSFlexGrid1.TextMatrix(linha_i, 4)
    TBTabela("lamina4") = MSFlexGrid1.TextMatrix(linha_i, 5)
End If
If nlinhas = 3 Then
    TBTabela("lamina2") = MSFlexGrid1.TextMatrix(linha_i, 3)
    TBTabela("lamina3") = MSFlexGrid1.TextMatrix(linha_i, 4)
End If
If nlinhas = 2 Then
    TBTabela("lamina2") = MSFlexGrid1.TextMatrix(linha_i, 3)
End If
TBTabela.Update

```

Next linha_i

TBDados.AddNew

```

If Text1.Text = "" Then
    Text1.Text = " "
End If
If Text2.Text = "" Then
    Text2.Text = " "
End If
If Text6.Text = "" Then
    Text6.Text = " "
End If
If Text7.Text = "" Then
    Text7.Text = " "
End If
If Text8.Text = "" Then
    Text8.Text = " "
End If
If Text9.Text = "" Then
    Text9.Text = " "
End If
If mskData.Text = "" Then
    mskData.Text = "    "

```

```

End If
If mskHorario.Text = "" Then
    mskHorario.Text = "  "
End If
If mskDuracao.Text = "" Then
    mskDuracao.Text = "    "
End If
If txtElemento.Text = "" Then
    txtElemento.Text = " "
End If
If cboElemento.Text = "" Then
    cboElemento.Text = " "
End If

TBDados("ensaio") = Text1.Text
TBDados("local") = Text2.Text
TBDados("dataensaio") = mskData.Text
TBDados("horainicio") = mskHorario.Text
TBDados("duracao") = mskDuracao.Text
TBDados("raio") = Text6.Text
TBDados("vazao") = Text7.Text
TBDados("trot") = Text8.Text
TBDados("obs") = Text9.Text
TBDados("elemento") = txtElemento
TBDados("unelemento") = cboElemento.Text
TBDados("nlinhas") = nlinhas

    For i = 1 To nlinhas
        TBDados.Fields(12 + i).Value = azimuth(i)
    Next i
TBDados.Update
TBDados.Close

TBTabela.Close
BD.Close

End Function

Private Function ExcluiLinha()

```

```

    u = u + 1
    Do While nlinha < linha - 1

        MSFlexGrid1.TextMatrix(nlinha, 1) = MSFlexGrid1.TextMatrix(nlinha + 1,
1)
        MSFlexGrid1.TextMatrix(nlinha, 2) = MSFlexGrid1.TextMatrix(nlinha +
1, 2)
        nlinha = nlinha + 1
    
```

```

Loop
    linha = linha - 1
    MSFlexGrid1.TextMatrix(linha, 0) = " "
    MSFlexGrid1.TextMatrix(linha, 1) = " "
    MSFlexGrid1.TextMatrix(linha, 2) = " "

MSFlexGrid1.Rows = MSFlexGrid1.Rows - 1
MSFlexGrid1.Refresh

End Function

Private Sub OrdenaMatrix()
    Dim linha_i, linha_u As Integer
    Dim i As Integer
    Dim Di, Li, Dj, Lj As Single

    ' incluindo mais um item

    linha_u = MSFlexGrid1.Rows

    MSFlexGrid1.AddItem linha_u & Chr(9) & iDistancia & Chr(9) & ilamina

    ' ordenando a matrix

    linha_u = MSFlexGrid1.Rows - 1
    linha_i = 1

    Do While MSFlexGrid1.TextMatrix(linha_i, 1) < iDistancia
        linha_i = linha_i + 1
    Loop

    Di = iDistancia
    Li = ilamina

    For i = linha_i To linha_u

        Dj = MSFlexGrid1.TextMatrix(i, 1)
        Lj = MSFlexGrid1.TextMatrix(i, 2)

        MSFlexGrid1.TextMatrix(i, 1) = Di
        MSFlexGrid1.TextMatrix(i, 2) = Li

        Di = Dj
        Li = Lj

    Next i

```

```
MSFlexGrid1.Refresh
```

```
End Sub
```

```
Function SóNúmeros(Key As Integer) As String
    'selecione os caracteres que desejar
    Const Números$ = ",.0123456789"
    SóNúmeros = Key
    If Key <> 8 Then 'conferindo se é backspace
        'vê se é um caracter permitido
        If InStr(Números$, Chr(Key)) = 0 Then
            SóNúmeros = 0
        End If
    End If
End Function
```

```
Private Sub Form_Terminate()
```

```
    frmPrincipal1.Enabled = True
    frmPrincipal1.Show
```

```
End Sub
```

```
Private Sub Text6_KeyPress(KeyAscii As Integer)
    KeyAscii = SóNúmeros(KeyAscii)
```

```
End Sub
```

```
Private Sub Text7_KeyPress(KeyAscii As Integer)
    KeyAscii = SóNúmeros(KeyAscii)
```

```
End Sub
```

```
Private Sub Text8_KeyPress(KeyAscii As Integer)
    KeyAscii = SóNúmeros(KeyAscii)
```

```
End Sub
```

```
Private Function encheGrid(Dados As String, sql As String)
    Dim k As Integer
```

```
Dados = strAccessDb
```

```
Set BD = OpenDatabase(strAccessDb)
Set TBTabela = BD.OpenRecordset("Tabela", dbOpenTable)
Set TBDados = BD.OpenRecordset("Dados", dbOpenTable)
```

```
TBDados.MoveFirst
```

```

        If TBDados("modalidade") <> "" Then modalidade =
TBDados("modalidade")
        If TBDados("ensaio") <> "" Then Text1.Text = TBDados("ensaio")
        If TBDados("local") <> "" Then Text2.Text = TBDados("local")
        If TBDados("dataensaio") <> "" Then mskData.Text =
TBDados("dataensaio")
        If TBDados("duracao") <> "" Then mskDuracao.Text =
TBDados("duracao")
        If TBDados("horainicio") <> "" Then mskHorario.Text =
TBDados("horainicio")
        If TBDados("raio") <> "" Then Text6.Text = TBDados("raio")
        If TBDados("vazao") <> "" Then Text7.Text = TBDados("vazao")
        If TBDados("trot") <> "" Then Text8.Text = TBDados("trot")
        If TBDados("obs") <> "" Then Text9.Text = TBDados("obs")
        If TBDados("elemento") <> "" Then txtElemento.Text =
TBDados("elemento")
        If TBDados("unelemento") <> "" Then cboElemento.Text =
TBDados("unelemento")

        nlinhas = TBDados.Fields(12).Value
        For coluna = 1 To nlinhas
            azimuth(coluna) = TBDados.Fields(12 + coluna).Value
        Next coluna

        txtNlinhas.Text = nlinhas
        If nlinhas = 1 Then
            txtAz1.Text = azimuth(1)
            lblAz2.Enabled = False: lblAz3.Enabled = False: lblAz4.Enabled = False
            txtAz2.Enabled = False: txtAz3.Enabled = False: txtAz4.Enabled = False
        End If
        If nlinhas = 2 Then
            txtAz1.Text = azimuth(1)
            txtAz2.Text = azimuth(2)
            lblAz3.Enabled = False: lblAz4.Enabled = False
            txtAz3.Enabled = False: txtAz4.Enabled = False
        End If
        If nlinhas = 3 Then
            txtAz1.Text = azimuth(1)
            txtAz2.Text = azimuth(2)
            txtAz3.Text = azimuth(3)
            lblAz4.Enabled = False
            txtAz4.Enabled = False
        End If
        If nlinhas = 4 Then
            txtAz1.Text = azimuth(1)
            txtAz2.Text = azimuth(2)
            txtAz3.Text = azimuth(3)
            txtAz4.Text = azimuth(4)
        End If
        ' define linhas fixas igual a uma e não usa colunas fixas

```

```

MSFlexGrid1.Rows = 2
MSFlexGrid1.FixedRows = 1
MSFlexGrid1.FixedCols = 0

' define o numero de linhas e colunas e cria uma matrix com o total de registros
a exibir
MSFlexGrid1.Rows = 1
MSFlexGrid1.Cols = TBTabela.Fields.Count

ReDim largura_coluna(0 To TBTabela.Fields.Count - 1)

ncolunas = 1 + nlinhas

MSFlexGrid1.TextMatrix(0, 0) = "coletor"
MSFlexGrid1.TextMatrix(0, 1) = " distancia (m)"

For coluna = 1 To ncolunas - 1
    MSFlexGrid1.TextMatrix(0, coluna + 1) = " lâmina" & coluna & " (mm)"
Next coluna

linha = 1
Do While Not TBTabela.EOF

    MSFlexGrid1.Rows = MSFlexGrid1.Rows + 1

    For coluna = 0 To nlinhas + 1
        MSFlexGrid1.TextMatrix(linha, coluna) = TBTabela.Fields(coluna).Value

    Next coluna

    TBTabela.MoveNext
    linha = linha + 1
Loop

' fecha o recordset e a conexao
TBDados.Close
TBTabela.Close
BD.Close

' define a largura das colunas do grid
MSFlexGrid1.ColWidth(0) = 650
MSFlexGrid1.ColWidth(1) = 1100

MSFlexGrid1.ColWidth(2) = 1300
MSFlexGrid1.ColWidth(3) = 1300
MSFlexGrid1.ColWidth(4) = 1300

```

```
MSFlexGrid1.ColWidth(5) = 1300
```

```
' verifica se os dados são de irrigação ou herbificação
If modalidade = "herbificação" Then
    optHerbigacao.Value = True
    MSFlexGrid1.TextMatrix(0, 2) = " concentração (ppm)"
    lblElemento.Visible = True
    txtElemento.Visible = True
    cboElemento.Visible = True
```

```
Else
    optIrrigacao.Value = True
```

```
End If
```

```
End Function
```

- **FrmPrincipal1**

```
Vref4 = glamina(1, j - 1)
    angulo = Abs(azimuteH(4) - azimuteH(1))
```

```
End If
```

```
Vint1 = Vref3 - (Vref3 - Vref1) * (angulo - (i - 1)) / angulo
Vint3 = Vref3 - (Vref3 - Vref1) * (angulo - i) / angulo
Vint2 = Vref4 - (Vref4 - Vref2) * (angulo - (i - 1)) / angulo
Vint4 = Vref4 - (Vref4 - Vref2) * (angulo - (i)) / angulo
```

```
laminaInterp(360 * j - (i - 1)) = ((Vint3 + Vint4) / 2 + (Vint1 + Vint2) / 2)
/ 2
```

```
raioInterp(360 * j - (i - 1)) = (distancia(j) + distancia(j - 1)) / 2
anguloInterp(360 * j - (i - 1)) = i - 0.5
areaInterp(360 * j - (i - 1)) = 3.14159 * ((distancia(j) ^ 2 - distancia(j - 1)
^ 2) / 360)
```

```
areaTotal = areaTotal + areaInterp(360 * j - (i - 1))
somaLamina = somaLamina + laminaInterp(360 * j - (i - 1)) *
areaInterp(360 * j - (i - 1))
Next i
Next j
```

```
laminaponderada = somaLamina / areaTotal
```

```
For j = nlinhas To 1 Step -1
    For i = 1 To 360
```

```

        soma2 = soma2 + Abs((laminaInterp(360 * j - (i - 1)) - laminaponderada)) *
areaInterp(360 * j - (i - 1))
    Next i
Next j

```

```
CUint = 100 * (1 - soma2 / areaTotal / laminaponderada)
```

```
' calculo do CUh
```

```

For j = 1 To npernas
    For i = 1 To nlinhas
        s1 = s1 + distancia(i)
        S2 = S2 + glamina(j, i) * distancia(i)
    Next i
    lampond(j) = S2 / s1
    For i = 1 To nlinhas
        S3 = S3 + distancia(i) * Abs(glamina(j, i) - (S2 / s1))
    Next i
    s4 = s4 + (1 - S3 / S2) * 100
Next j
CUh = s4 / npernas

```

```
End Select
```

```

CUi = Int(10 * CUint + 0.5) / 10
laminaponderada = Int(10 * laminaponderada + 0.5) / 10
CUh = Int(10 * CUh + 0.5) / 10
Label6.Caption = CUi
Label9.Caption = laminaponderada
Label8.Caption = CUh

```

```
If modalidade = "irrigação" Then
```

```
If q_Ef = 0 Or raio_Ef = 0 Or tr_Ef = 0 Then
```

```

lblEficiencia.Visible = False
Label11.Visible = False
Label10.Visible = False

```

```

Else
lblEficiencia.Visible = True
Label11.Visible = True
Label10.Visible = True

```

```

Epa = Int(1000 * (laminaponderada * 10 * 3.141592653 * (raio_Ef ^ 2)) /
(10000 * q_Ef * 3.6 * tr_Ef)) / 10
Label11.Caption = Epa

```

```

    End If
Else
    Label11.Visible = False
    Label10.Visible = False
    lblEficiencia.Visible = False
End If
End Sub

Private Sub DesenhaEixos()

    Dim escalaX As Single
    Dim escalaY As Single
    Dim i As Integer
    Dim j As Integer

    If modalidade = "irrigação" Then
        DrawRotatedText Picture2, "lâmina aplicada (mm)", _
            180, 3200, _
            "MS Sans Serif", 8, 90, 400, False, False, False
    Else
        DrawRotatedText Picture2, "concentração de " & Elemento & " (" &
unElemento & ")", _
            180, 4000, _
            "MS Sans Serif", 8, 90, 400, False, False, False
    End If
    DrawRotatedText Picture2, "distância radial do centro do pivot (m)", _
        2700, 4800, _
        "MS Sans Serif", 8, 0, 400, False, False, False

    ' desenha os eixos x e y

    'Picture1.ForeColor = vbBlue

    Picture2.Line (700, 4500)-(7200, 4500)
    Picture2.Line -(7200, 500)
    Picture2.Line -(700, 500)
    Picture2.Line -(700, 4500)

    ' escala dos eixos

    escalaX = 650
    escalaY = -400
    espacamento = distancia(nlinhas) - distancia(nlinhas - 1)
    limiteX = distancia(1) + nlinhas * espacamento + espacamento / 2
    fatorX = Int(limiteX / 100) + 1

    For i = 700 To 7200 Step escalaX
        Picture2.Line (i, 4500)-(i, 4550)
    
```

```

Picture2.ForeColor = &H8000000F
If Int(j / 100) = 0 Then
    Picture2.PSet (i - 120, 4550)
Else
    Picture2.PSet (i - 160, 4550)
End If
If j = 0 Then
    Picture2.PSet (i - 100, 4550)
End If
Picture2.ForeColor = vbBlack

Picture2.Print Str(j)
j = j + 10 * fatorX
Next i

j = 0
limiteY = glamina(1, 1)

' maior valor da lâmina
For i = 2 To nlinhas
    If glamina(1, i) > limiteY Then
        limiteY = glamina(1, i)
    End If
Next i
fatorY = Int(limiteY / 10 + 1)

For i = 4500 To 500 Step escalaY
    Picture2.Line (700, i)-(650, i)
    Picture2.ForeColor = &H8000000F
    If Int(j / 10) = 0 Then
        Picture2.PSet (500, i - 120)
    Else
        Picture2.PSet (400, i - 120)
    End If
    If j = 0 Then
        Picture2.PSet (500, i - 100)
    End If
    Picture2.ForeColor = vbBlack
    Picture2.Print Str(j)
    j = j + fatorY
Next i

End Sub

Private Sub PlotaPontos()
Dim i As Single

```

```

Dim plotX, plotY As Single
Dim Xi, Yi As Single

Xi = 700 + distancia(1) * 650 / (10 * fatorX)

Yi = (4500 - glamina(leg, 1) * 400 / fatorY)

For i = 1 To nlinhas

    plotX = (700 + distancia(i) * 650 / (10 * fatorX))
    plotY = (4500 - glamina(leg, i) * 400 / fatorY)

    If chkPontos.Value = 1 Then
        Picture2.Circle (plotX, plotY), 15, vbBlack

    End If
    If chkLinhas.Value = 1 Then
        DrawWidth = 2
        Picture2.Line (Xi, Yi)-(plotX, plotY), vbBlue
    End If
    Xi = plotX
    Yi = plotY

Next i
If chkPonderada.Value = 1 Then
    plotY = (4500 - lampond(leg) * 400 / fatorY)
    Picture2.Line (700, plotY)-(7200, plotY), vbRed

End If
Picture2.PSet (5500, 700)
Picture2.Print "Azimute: "; azimuthK(leg); ""
Toolbar1.Buttons.Item("salvar").Enabled = True
End Sub

Private Sub mnuLaminaAplicada_Click()

    carregaDados

    Picture2.Picture = Nothing
    DesenhaEixos
    leg = 1
    nome$ = "graf2D" & CStr(leg)
    PlotaPontos
    Picture2.Picture = Picture2.Image
    SavePicture Picture2.Picture, App.path & "\" & nome$ & ".bmp"
    gerado2D(leg) = True
    Toolbar1.Buttons.Item("salvar").Enabled = True
    mnuSalvar.Enabled = True
    mnuSalvarComo.Enabled = True

```

End Sub

```
Private Sub mnuRelatorio_Click()
    Dim fso As New FileSystemObject
```

```
    Call fso.CopyFile(strAccessDb, "C:\VPivot.mdb")
    frmPrincipal1.Enabled = False
    frmVisualizaRelatorio.Show
```

End Sub

```
Private Sub mnuSair_Click()
```

```
    Unload Me
```

End Sub

```
Private Sub mnuSalvar_Click()
```

```
    Picture2.Picture = Picture2.Image
```

```
    SavePicture Picture2.Picture, App.path & "\grafico.bmp"
```

```
    arquivo = "grafico.bmp"
```

End Sub

```
Private Sub mnuSalvarComo_Click()
```

```
Picture2.Picture = Picture2.Image
```

```
SavePicture Picture2.Picture, App.path & "\grafico.bmp"
```

```
arquivo = "grafico.bmp"
```

```
frmSalvaGrafico.Show
```

End Sub

```
Private Sub mnuSobre_Click()
```

```
frmAbout.Show vbModal
```

End Sub

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
```

```
Select Case Button.Key
```

```
Case "novo"
```

```
frmInserirDados.Show
```

```
Case "abrir"
```

```
frmEscolhaBD1.Show
```

```
Case "salvar"
```

```
Picture2.Picture = Picture2.Image
```

```
SavePicture Picture2.Picture, App.path & "\grafico.bmp"
```

```
arquivo = "grafico.bmp"
```

```
frmSalvaGrafico.Show
```

Case "grafico2D"

carregaDados

Picture2.Picture = Nothing

DesenhaEixos

leg = 1

nome\$ = "graf2D" & CStr(leg)

PlotaPontos

Picture2.Picture = Picture2.Image

SavePicture Picture2.Picture, App.path & "\" & nome\$ & ".bmp"

gerado2D(leg) = True

Toolbar1.Buttons.Item("salvar").Enabled = True

mnuSalvar.Enabled = True

mnuSalvarComo.Enabled = True

Case "Mapa"

geradoMapa = True

FazMapa

Picture2.Picture = Picture2.Image

SavePicture Picture2.Picture, App.path & "\" & nome\$ & ".bmp"

Toolbar1.Buttons.Item("salvar").Enabled = True

mnuSalvar.Enabled = True

mnuSalvarComo.Enabled = True

cmdEditaLegenda.Enabled = True

Case "imprimir"

Dim fso As New FileSystemObject

Call fso.CopyFile(strAccessDb, "C:\VPivot.mdb")

frmPrincipal1.Enabled = False

frmVisualizaRelatorio.Show

Case "ajuda"

App.HelpFile = App.path & "\AJUDA.HLP"

Call WinHelp(frmPrincipal1.hWnd, App.HelpFile, HELP_FINDER, ByVal

0&)

Case "sair"

Unload Me

End Select

End Sub

Private Sub cmdEditaLegenda_Click()

frmAlteraLegenda.Show

End Sub

```
Private Sub cmdMapa_Click()
```

```
geradoMapa = True
FazMapa
Picture2.Picture = Picture2.Image
SavePicture Picture2.Picture, App.path & "\graf1D.bmp"
Toolbar1.Buttons.Item("salvar").Enabled = True
mnuSalvar.Enabled = True
mnuSalvarComo.Enabled = True
cmdEditaLegenda.Enabled = True
End Sub
```

```
Private Function procuraFaixa()
```

```
For j = 1 To numfaixa
    If ilamina = 0 Then
        ifaixa = 1
        Exit Function
    End If
    If ilamina > LIsup(numfaixa) Then
        ifaixa = numfaixa
        'LIsup(numfaixa) = gdiam(i)
        Exit Function
    End If

    If ilamina > LIinf(j) And ilamina <= LIsup(j) Then
        ifaixa = j
        Exit Function
    End If
Next j

End Function
```

```
Private Sub desenhaArco()
```

```
If a = 1 Then
    Picture2.ForeColor = &HC0C0FF
    Picture2.FillStyle = vbSolid
    Picture2.FillColor = &HC0C0FF

End If
If a = 2 Then
    Picture2.ForeColor = &HFF00FF
    Picture2.FillStyle = vbSolid
    Picture2.FillColor = &HFF00FF
```

```

End If
If a = 3 Then
    Picture2.ForeColor = &HFFFF00
    Picture2.FillStyle = vbSolid
    Picture2.FillColor = &HFFFF00

End If
If a = 4 Then
    Picture2.ForeColor = &H808080
    Picture2.FillStyle = vbSolid
    Picture2.FillColor = &H808080

End If
If a = 5 Then
    Picture2.ForeColor = &HFF0000
    Picture2.FillStyle = vbSolid
    Picture2.FillColor = &HFF0000

End If

If inicio <= 0.5 * pi Then
    inicio = 0.5 * pi - inicio
Else
    inicio = 2.5 * pi - inicio
End If
If inicio = 0 Then inicio = 0.001
If inicio > 2 * pi Then inicio = 6.28
If final <= 0.5 * pi Then
    final = 0.5 * pi - final
Else
    final = 2.5 * pi - final
End If
If final = 0 Then final = 0.001
If final > 2 * pi Then final = 6.28

Picture2.Circle (3000, 2650), 2400 * Rx / gr(u), , -final, -inicio

End Sub

Private Sub legenda()
    Picture2.ForeColor = vbBlack
    Picture2.FillStyle = vbSolid
    Picture2.CurrentX = 5800
    Picture2.CurrentY = 200
    Picture2.FontUnderline = True

    Picture2.Font = "Tahoma"

```

```

Picture2.FontSize = 10
Picture2.Print "LEGENDA"
Picture2.FontSize = 8
Picture2.FontUnderline = False

For j = 1 To numfaixa
    If j = 1 Then
        Picture2.ForeColor = &HC0C0FF
        Picture2.FillStyle = vbSolid
        Picture2.FillColor = &HC0C0FF
        End If
    If j = 2 Then
        Picture2.ForeColor = &HFF00FF
        Picture2.FillStyle = vbSolid
        Picture2.FillColor = &HFF00FF
        End If
    If j = 3 Then
        Picture2.ForeColor = &HFFFF00
        Picture2.FillStyle = vbSolid
        Picture2.FillColor = &HFFFF00
        End If
    If j = 4 Then
        Picture2.ForeColor = &H808080
        Picture2.FillStyle = vbSolid
        Picture2.FillColor = &H808080
        End If
    If j = 5 Then
        Picture2.ForeColor = &HFF0000
        Picture2.FillStyle = vbSolid
        Picture2.FillColor = &HFF0000
        End If

    Picture2.Line (5800, 600 * j)-(6000, 200 + 600 * j), , BF

    Picture2.ForeColor = vbBlack
    Picture2.CurrentX = 6200
    Picture2.CurrentY = 600 * j
    ud = " mm"
    If modalidade = "herbigação" Then ud = unElemento
    Picture2.Print LInf(j); " - "; LIsup(j); ud

Next j

End Sub

```

Private Sub FazMapa()

```
Set BancodeDados = OpenDatabase(strAccessDb)
Set TBTabela = BancodeDados.OpenRecordset("Tabela", dbOpenTable)
Set TBDados = BancodeDados.OpenRecordset("Dados", dbOpenTable)
```

```
nlinhas = TBDados("nlinhas")
```

```
For i = 1 To nlinhas
    azimuteH(i) = TBDados.Fields(12 + i).Value
Next i
    azimuteH(nlinhas + 1) = 360
```

```
TBTabela.MoveFirst
i = 1
dlamin = 1000
dlamax = 0
Do While Not TBTabela.EOF
```

```
    gr(i) = TBTabela("distancia")
```

```
    gdlam(1, i) = TBTabela("lamina1")
    If gdlam(1, i) > dlamax Then dlamax = gdlam(1, i)
    If gdlam(1, i) < dlammin Then dlammin = gdlam(1, i)
```

```
    If nlinhas = 2 Then
```

```
        gdlam(2, i) = TBTabela("lamina2")
        If gdlam(2, i) > dlamax Then dlamax = gdlam(2, i)
        If gdlam(2, i) < dlammin Then dlammin = gdlam(2, i)
```

```
    End If
```

```
    If nlinhas = 3 Then
```

```
        gdlam(2, i) = TBTabela("lamina2")
        If gdlam(2, i) > dlamax Then dlamax = gdlam(2, i)
        If gdlam(2, i) < dlammin Then dlammin = gdlam(2, i)
        gdlam(3, i) = TBTabela("lamina3")
        If gdlam(3, i) > dlamax Then dlamax = gdlam(3, i)
        If gdlam(3, i) < dlammin Then dlammin = gdlam(3, i)
```

```
    End If
```

```
    If nlinhas = 4 Then
```

```
        gdlam(2, i) = TBTabela("lamina2")
        If gdlam(2, i) > dlamax Then dlamax = gdlam(2, i)
        If gdlam(2, i) < dlammin Then dlammin = gdlam(2, i)
        gdlam(3, i) = TBTabela("lamina3")
        If gdlam(3, i) > dlamax Then dlamax = gdlam(3, i)
        If gdlam(3, i) < dlammin Then dlammin = gdlam(3, i)
        gdlam(4, i) = TBTabela("lamina4")
        If gdlam(4, i) > dlamax Then dlamax = gdlam(4, i)
        If gdlam(4, i) < dlammin Then dlammin = gdlam(4, i)
```

```

    End If
    i = i + 1
    TBTabela.MoveNext
Loop

    u = i - 1

TBDados.Close
TBTabela.Close
BancodeDados.Close

' verifica a faixa

For i = 1 To u

    For k = 1 To nlinhas

        ilamina = gdlam(k, i)
        Call procuraFaixa
        faixa(k, i) = ifaixa
    Next k

Next i
For i = 1 To u
    faixa(nlinhas + 1, i) = faixa(1, i)
    gdlam(nlinhas + 1, i) = gdlam(1, i)
Next i

' traça o círculo externo
Picture2.Picture = Nothing
Picture2.ForeColor = vbBlack
Picture2.FillColor = vbWhite
Picture2.FillStyle = vbTransparent
Picture2.Circle (3000, 2650), 2440

'#####PULA traça círculo

If nlinhas = 1 Then
    FazMapa2
Else

    intervalo = (gr(u) - gr(u - 1)) / 5

    For i = u To 1 Step -1
        For m1 = 1 To 5

            dr(m1) = gr(i) - m1 * intervalo

            If nlinhas = 1 Then azimuthH(1) = 0

```

For k1 = 1 To nlinhas

```

    dl(k1, m1) = gdlam(k1, i) - m1 * intervalo * (gdlam(k1, i - 1) - gdlam(k1,
i)) / (gr(i - 1) - gr(i))
    ilamina = dl(k1, m1)
    procuraFaixa
    fx(k1, m1) = ifaixa

```

Next k1

Next m1

For m = 1 To 5

For k = 1 To nlinhas

```

    a = faixa(k, i)
    b = faixa(k + 1, i)
    Rx = gr(i)
    qtfaixa = a - b

```

If qtfaixa = 0 Then

```

    inicio = azimuteH(k) * pi / 180
    If inicio = 0 Then inicio = 0.001
    final = azimuteH(k + 1) * pi / 180
    If final = 0 Then final = 0.001
    desenhaArco

```

End If

If qtfaixa = 1 Then

```

    azimute1 = azimuteH(k)
    azimute2 = azimuteH(k + 1)
    c = gdlam(k, i)
    e = gdlam(k + 1, i)
    d = LInf(a)
    azimutex = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
    inicio = azimute1 * pi / 180
    final = azimutex * pi / 180
    If inicio = 0 Then inicio = 0.001
    If final = 0 Then final = 0.001
    desenhaArco

```

a = a - 1

d = e

azimute1 = azimutex

azimutex = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)

inicio = azimute1 * pi / 180

```

    final = azimuthx * pi / 180
    desenhaArco
End If

If qtfaixa = 2 Then
    azimuth1 = azimuthH(k)
    azimuth2 = azimuthH(k + 1)
    c = gdlam(k, i)
    e = gdlam(k + 1, i)
    d = Llinf(a)
    azimuthx = azimuth1 + (d - c) * (azimuth2 - azimuth1) / (e - c)
    inicio = azimuth1 * pi / 180
    final = azimuthx * pi / 180
    If inicio = 0 Then inicio = 0.001
    If final = 0 Then final = 0.001
    desenhaArco

    a = a - 1
    d = Llinf(a)
    azimuth1 = azimuthx
    azimuthx = azimuth1 + (d - c) * (azimuth2 - azimuth1) / (e - c)
    inicio = azimuth1 * pi / 180
    final = azimuthx * pi / 180
    If inicio = 0 Then inicio = 0.001
    If final = 0 Then final = 0.001
    desenhaArco

    a = a - 1
    d = e
    azimuth1 = azimuthx
    azimuthx = azimuth1 + (d - c) * (azimuth2 - azimuth1) / (e - c)
    inicio = azimuth1 * pi / 180
    final = azimuthx * pi / 180
    If inicio = 0 Then inicio = 0.001
    If final = 0 Then final = 0.001
    desenhaArco
End If

If qtfaixa = 3 Then
    azimuth1 = azimuthH(k)
    azimuth2 = azimuthH(k + 1)
    c = gdlam(k, i)
    e = gdlam(k + 1, i)
    d = Llinf(a)
    azimuthx = azimuth1 + (d - c) * (azimuth2 - azimuth1) / (e - c)
    inicio = azimuth1 * pi / 180
    final = azimuthx * pi / 180
    If inicio = 0 Then inicio = 0.001
    If final = 0 Then final = 0.001
    desenhaArco

```

```

a = a - 1
d = Llinf(a)
azimute1 = azimutx
azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimutx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

```

a = a - 1
d = Llinf(a)
azimute1 = azimutx
azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimutx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

```

a = a - 1
d = e
azimute1 = azimutx
azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimutx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

End If

```

If qtfaixa = 4 Then
azimute1 = azimuteH(k)
azimute2 = azimuteH(k + 1)
c = gdlam(k, i)
e = gdlam(k + 1, i)
d = Llinf(a)
azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimutx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

```

a = a - 1
d = Llinf(a)
azimute1 = azimutx

```

```

azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimutx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

```

a = a - 1
d = Llinf(a)
azimute1 = azimutx
azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimutx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

```

a = a - 1
d = Llinf(a)
azimute1 = azimutx
azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimutx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

```

a = a - 1
d = e
azimute1 = azimutx
azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimutx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco
End If

```

'##### valores negativos

```

If qtfaixa = -1 Then
azimute1 = azimuteH(k)
azimute2 = azimuteH(k + 1)
c = gdlam(k, i)
e = gdlam(k + 1, i)
d = LIsup(a)
azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimutx * pi / 180

```

If inicio = 0 Then inicio = 0.001

If final = 0 Then final = 0.001

desenhaArco

a = a + 1

d = e

azimute1 = azimutx

azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)

inicio = azimute1 * pi / 180

final = azimutx * pi / 180

desenhaArco

End If

If qtfaixa = -2 Then

azimute1 = azimuteH(k)

azimute2 = azimuteH(k + 1)

c = gdlam(k, i)

e = gdlam(k + 1, i)

d = LIsup(a)

azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)

inicio = azimute1 * pi / 180

final = azimutx * pi / 180

If inicio = 0 Then inicio = 0.001

If final = 0 Then final = 0.001

desenhaArco

a = a + 1

d = LIsup(a)

azimute1 = azimutx

azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)

inicio = azimute1 * pi / 180

final = azimutx * pi / 180

If inicio = 0 Then inicio = 0.001

If final = 0 Then final = 0.001

desenhaArco

a = a + 1

d = e

azimute1 = azimutx

azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)

inicio = azimute1 * pi / 180

final = azimutx * pi / 180

If inicio = 0 Then inicio = 0.001

If final = 0 Then final = 0.001

desenhaArco

End If

If qtfaixa = -3 Then

azimute1 = azimuteH(k)

azimute2 = azimuteH(k + 1)

```

c = gdlam(k, i)
e = gdlam(k + 1, i)
d = LIsup(a)
azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimutx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

```

a = a + 1
d = LIsup(a)
azimute1 = azimutx
azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimutx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

```

a = a + 1
d = LIsup(a)
azimute1 = azimutx
azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimutx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

```

a = a + 1
d = e
azimute1 = azimutx
azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimutx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

End If

```

If qtfaixa = -4 Then
azimute1 = azimuteH(k)
azimute2 = azimuteH(k + 1)
c = gdlam(k, i)
e = gdlam(k + 1, i)
d = LIsup(a)
azimutx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180

```

```

final = azimuthx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

```

a = a + 1
d = LIsup(a)
azimute1 = azimuthx
azimuthx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimuthx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

```

a = a + 1
d = LIsup(a)
azimute1 = azimuthx
azimuthx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimuthx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

```

a = a + 1
d = LIsup(a)
azimute1 = azimuthx
azimuthx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimuthx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco

```

```

a = a + 1
d = e
azimute1 = azimuthx
azimuthx = azimute1 + (d - c) * (azimute2 - azimute1) / (e - c)
inicio = azimute1 * pi / 180
final = azimuthx * pi / 180
If inicio = 0 Then inicio = 0.001
If final = 0 Then final = 0.001
desenhaArco
End If

```

Next k

If i <> 1 Then

gr(i) = dr(m)

For k = 1 To nlinhas

gdlam(k, i) = dl(k, m)

faixa(k, i) = fx(k, m)

Next k

End If

Next m

Next i

Picture2.ForeColor = &H80000003

Picture2.FillStyle = vbSolid

Picture2.FillColor = &H80000003

Rx = gr(1)

Picture2.Circle (3000, 2650), 2400 * Rx / gr(u)

Picture2.ForeColor = vbBlack

Picture2.FillStyle = vbTransparent

Picture2.Line (800, 3750)-(5200, 1550)

Picture2.Line (1670, 4700)-(4330, 600)

Picture2.Line (800, 1550)-(5200, 3750)

Picture2.Line (1670, 600)-(4330, 4700)

Picture2.Line (550, 2650)-(5450, 2650)

Picture2.Line (3000, 200)-(3000, 5100)

Picture2.Font = "Tahoma"

Picture2.CurrentX = 5490

Picture2.CurrentY = 2550

Picture2.Print "90°"

Picture2.CurrentX = 200

Picture2.CurrentY = 2550

Picture2.Print "270°"

Picture2.CurrentX = 2900

Picture2.CurrentY = 20

Picture2.Print "0°N"

Picture2.CurrentX = 2900

Picture2.CurrentY = 5150

Picture2.Print "180°"

```

        legenda
    End If

```

```

End Sub

```

```

Private Sub fazfaixa()

```

```

    Dim incremento As Single
    LInf(1) = (Int(dlamin / 10)) * 10
    LSup(numfaixa) = Int(10 + dlamax)
    incremento = Int((LSup(numfaixa) - LInf(1)) / 5 + 0.5)
    LSup(1) = LInf(1) + incremento
        For j = 2 To numfaixa
            LInf(j) = LSup(j - 1)
            LSup(j) = LInf(j) + incremento
        Next j

```

```

End Sub

```

```

Private Sub FazMapa2()

```

```

    If dlamax > LSup(numfaixa) Then
        fazfaixa
        For i = 1 To u
            ilamina = gdlam(1, i)
            Call procuraFaixa
            faixa(1, i) = ifaixa
        Next i
    End If

```

```

Picture2.Picture = Nothing

```

```

Picture2.ForeColor = vbBlack
Picture2.FillColor = vbWhite
Picture2.FillStyle = vbTransparent

```

```

Picture2.Circle (3000, 2650), 2440

```

```

For i = u To 1 Step -1

```

```

    qtfaixa = faixa(1, i) - faixa(1, i - 1)

```

```

        a = faixa(1, i)
        Rx = gr(i)

```

desenhaCirculo

If qtfaixa = 0 Then

a = faixa(1, i)

Rx = gr(i - 1)

desenhaCirculo

End If

If qtfaixa = 1 Then

dx = LInf(a)

a = a - 1

Rx = gr(i) - (gr(i) - gr(i - 1)) * (gdlam(1, i) - dx) / (gdlam(1, i) -
gdlam(1, i - 1))

desenhaCirculo

End If

If qtfaixa = 2 Then

For k = 1 To 2

dx = LInf(a)

a = a - 1

Rx = gr(i) - (gr(i) - gr(i - 1)) * (gdlam(1, i) - dx) / (gdlam(1, i) -
gdlam(1, i - 1))

desenhaCirculo

Next k

End If

If qtfaixa = 3 Then

For k = 1 To 3

dx = LInf(a)

a = a - 1

Rx = gr(i) - (gr(i) - gr(i - 1)) * (gdlam(1, i) - dx) / (gdlam(1, i) -
gdlam(1, i - 1))

desenhaCirculo

Next k

End If

If qtfaixa = 4 Then

For k = 1 To 4

dx = LInf(a)

a = a - 1

```

Rx = gr(i) - (gr(i) - gr(i - 1)) * (gdlam(1, i) - dx) / (gdlam(1, i) -
gdlam(1, i - 1))
  desenhaCirculo
  Next k

End If

If qtfaixa = -1 Then

  dx = LIsup(a)
  a = a + 1
  Rx = gr(i) - (gr(i) - gr(i - 1)) * (gdlam(1, i) - dx) / (gdlam(1, i) -
gdlam(1, i - 1))
  desenhaCirculo

End If

If qtfaixa = -2 Then

  For k = 1 To 2
    dx = LIsup(a)
    a = a + 1
    Rx = gr(i) - (gr(i) - gr(i - 1)) * (gdlam(1, i) - dx) / (gdlam(1, i) -
gdlam(1, i - 1))
    desenhaCirculo
  Next k

End If

If qtfaixa = -3 Then

  For k = 1 To 3
    dx = LIsup(a)
    a = a + 1
    Rx = gr(i) - (gr(i) - gr(i - 1)) * (gdlam(1, i) - dx) / (gdlam(1, i) -
gdlam(1, i - 1))
    desenhaCirculo
  Next k

End If

If qtfaixa = -4 Then

  For k = 1 To 4
    dx = LIsup(a)
    a = a + 1
    Rx = gr(i) - (gr(i) - gr(i - 1)) * (gdlam(1, i) - dx) / (gdlam(1, i) -
gdlam(1, i - 1))
    desenhaCirculo
  
```

Next k

End If

Rx = gr(i - 1)
desenhaCirculo

Next i

Picture2.ForeColor = &H80000003
Picture2.FillStyle = vbSolid
Picture2.FillColor = &H80000003

Rx = gr(1)
'area(a) = 3.14159 * Rx ^ 2
Picture2.Circle (3000, 2650), 2400 * Rx / gr(u)
'areaAcum(a) = areaAcum(a) - area(a)
Picture2.ForeColor = vbBlack
Picture2.FillStyle = vbTransparent

Picture2.ForeColor = vbBlack
Picture2.FillStyle = vbTransparent

Picture2.Line (800, 3750)-(5200, 1550)
Picture2.Line (1670, 4700)-(4330, 600)
Picture2.Line (800, 1550)-(5200, 3750)
Picture2.Line (1670, 600)-(4330, 4700)
Picture2.Line (550, 2650)-(5450, 2650)
Picture2.Line (3000, 200)-(3000, 5100)

Picture2.Font = "Tahoma"
Picture2.CurrentX = 5490
Picture2.CurrentY = 2550
Picture2.Print "90°"
Picture2.CurrentX = 200
Picture2.CurrentY = 2550
Picture2.Print "270°"

Picture2.CurrentX = 2900
Picture2.CurrentY = 20
Picture2.Print "0°N"
Picture2.CurrentX = 2900
Picture2.CurrentY = 5150
Picture2.Print "180°"

legenda

```

End Sub
Private Sub desenhaCirculo()

If a = 1 Then
    Picture2.ForeColor = &HC0C0FF
    Picture2.FillStyle = vbSolid
    Picture2.FillColor = &HC0C0FF

End If
If a = 2 Then
    Picture2.ForeColor = &HFF00FF
    Picture2.FillStyle = vbSolid
    Picture2.FillColor = &HFF00FF

End If
If a = 3 Then
    Picture2.ForeColor = &HFFFF00
    Picture2.FillStyle = vbSolid
    Picture2.FillColor = &HFFFF00

End If
If a = 4 Then
    Picture2.ForeColor = &H808080
    Picture2.FillStyle = vbSolid
    Picture2.FillColor = &H808080

End If
If a = 5 Then
    Picture2.ForeColor = &HFF0000
    Picture2.FillStyle = vbSolid
    Picture2.FillColor = &HFF0000

End If

Picture2.Circle (3000, 2650), 2400 * Rx / gr(u)

End Sub

Public Sub carregaDados()
Set BD = OpenDatabase(strAccessDb)
Set TBTabela = BD.OpenRecordset("Tabela", dbOpenTable)
Set TBDados = BD.OpenRecordset("Dados", dbOpenTable)

TBDados.MoveFirst
npernas = TBDados("nlinhas")

TBTabela.MoveFirst

```

```

For j = 1 To 500

If TBtabela.EOF = False Then
    distancia(j) = TBtabela("distancia")
    TBtabela.MoveNext
    nlinhas = j
End If
Next j


For i = 1 To npernas
    lamina = "lamina" & CStr(i)
    TBtabela.MoveFirst

    For u = 1 To nlinhas

        glamina(i, u) = TBtabela(lamina)

        TBtabela.MoveNext

    Next u
Next i


For i = 1 To npernas
    azimute = "azimute" & CStr(i)
    azimuteH(i) = TBDados(azimute)
    azimuteK(i) = TBDados(azimute)
Next i


TBDados.Close
TBtabela.Close
BD.Close
End Sub


Private Sub mnuDistribuicao_click()
geradoMapa = True
    FazMapa
    Picture2.Picture = Picture2.Image
    SavePicture Picture2.Picture, App.path & "\graf1D.bmp"
    Toolbar1.Buttons.Item("salvar").Enabled = True
    mnuSalvar.Enabled = True
    mnuSalvarComo.Enabled = True
    cmdEditaLegenda.Enabled = True

End Sub

```

- **FrmSalvarGrafico**

```

Private Sub cmdCancelar_Click()
frmPrincipal1.Enabled = True
frmPrincipal1.Show

Unload frmSalvaGrafico
End Sub

Private Sub cmdSalvar_Click()

Dim fso As New FileSystemObject

    arquivo = Dir1.path & "\" & txtArquivo.Text & ".bmp"
    temporarioDb = "c:\proPIVOT\grafico.bmp"

    If txtArquivo.Text = "" Then
        MsgBox ("Digite um nome para o arquivo")
    Else

        Call fso.CopyFile(temporarioDb, arquivo)

        Unload Me

    End If

End Sub

Private Sub Dir1_Change()
File1.path = Dir1.path

End Sub

Private Sub Drive1_Change()
Dir1.path = Drive1.Drive
End Sub

Private Sub File1_Click()
    arquivo = File1.Filename
    ' tira a extensão do arquivo
    arquivo = Left$(arquivo, InStr(arquivo, ".") - 1)
    txtArquivo.Text = arquivo

End Sub

Private Sub Form_Load()

```

```

Dir1.path = App.path
Drive1.Drive = Dir1.path

If arquivo <> "" Then
    arquivo = Left$(arquivo, InStr(arquivo, ".") - 1)
End If

    txtArquivo.Text = arquivo

If salvaGrafico = True Then
    File1.Pattern = "*.bmp"
End If

End Sub

Private Sub txtArquivo_Click()
txtArquivo.Text = ""

End Sub
'Private Function SalvaCopia()

'Dim fso As New FileSystemObject

'Call fso.CopyFile(strAccessDb, temporarioDb)

'End Function

Function SóNúmeros(Key As Integer) As String
'selecione os caracteres que desejar
Const Números$ = ",."
SóNúmeros = Key
If Key <> 8 Then 'conferindo se é backspace
    'vê se é um caracter permitido
    If InStr(Números$, Chr(Key)) <> 0 Then
        SóNúmeros = 0
    End If
End If
End Function

Private Sub txtArquivo_KeyPress(KeyAscii As Integer)

    KeyAscii = SóNúmeros(KeyAscii)

End Sub

Private Sub procuraArquivo()
Dim retval As String

retval = Dir(arquivo)

```

```

If retval <> "" Then
    'MsgBox "arquivo existe"
    Kill arquivo

```

```

End If

```

```

End Sub

```

- **FrmSalvar**

```

Private Sub cmdCancelar_Click()
    frmLerBD1.Enabled = True
    frmInserirDados.Enabled = True

```

```

Unload frmSalvar
End Sub

```

```

Private Sub cmdSalvar_Click()

```

```

    Dim fso As New FileSystemObject

```

```

    arquivo = Dir1.path & "\" & txtArquivo.Text & ".mdb"
    strAccessDb = arquivo

```

```

    ' verifica se o arquivo existe
    procuraArquivo

```

```

If txtArquivo.Text = "" Then
    MsgBox ("Digite um nome para o arquivo")
    Exit Sub
Else

```

```

    Call fso.CopyFile(temporarioDb, arquivo)

```

```

'retirando o caminho do arquivo
    i = Len(arquivo)
    Do While Mid$(arquivo, i, 1) <> "\"
        i = i - 1
    Loop

```

```

    arquivo = Right$(arquivo, Len(arquivo) - i)

```

```

    Unload frmSalvar

```

```

End If
If formulario = "inserir" Then
    frmInserirDados.Enabled = True

```

```

frmInserirDados.Show

Else
    frmLerBD1.Enabled = True
    frmLerBD1.Show
End If

End Sub

Private Sub Dir1_Change()
    File1.path = Dir1.path

End Sub

Private Sub Drive1_Change()
    Dir1.path = Drive1.Drive
End Sub

Private Sub File1_Click()
    arquivo = File1.Filename
    ' tira a extensão do arquivo
    arquivo = Left$(arquivo, InStr(arquivo, ".") - 1)
    txtArquivo.Text = arquivo

End Sub

Private Sub Form_Load()

    Dir1.path = App.path
    Drive1.Drive = Dir1.path

    If arquivo <> "" Then
        arquivo = Left$(arquivo, InStr(arquivo, ".") - 1)
    End If

    txtArquivo.Text = arquivo

    If salvaGrafico = True Then
        File1.Pattern = "*.bmp"
    End If

End Sub

Private Sub txtArquivo_Click()
    txtArquivo.Text = ""

End Sub

'Private Function SalvaCopia()

```

```

'Dim fso As New FileSystemObject

'Call fso.CopyFile(strAccessDb, temporarioDb)

'End Function

Function SóNúmeros(Key As Integer) As String
    'selecione os caracteres que desejar
    Const Números$ = ",."
    SóNúmeros = Key
    If Key <> 8 Then 'conferindo se é backspace
        'vê se é um caracter permitido
        If InStr(Números$, Chr(Key)) <> 0 Then
            SóNúmeros = 0
        End If
    End If
End Function

Private Sub txtArquivo_KeyPress(KeyAscii As Integer)

    KeyAscii = SóNúmeros(KeyAscii)

End Sub

Private Sub procuraArquivo()
Dim retval As String

retval = Dir(arquivo)
If retval <> "" Then
    'MsgBox "arquivo existe"
    Kill arquivo

End If

End Sub

```

- **FrmVisualizarRelatorio**

```

Option Explicit
Dim db As Database
Dim rs As Recordset
Dim area As Workspace
Dim pagina, numpagina As Integer
Dim objprint As Object
Dim ratio As Double
Dim controle As Single
Dim imprimir As Boolean
Dim i, k As Integer

```

```

Private Sub visualiza()
    Dim dRatio As Double
    Picture1.Picture = Nothing
    dRatio = ScalePicPreviewToPrinterInches(Picture1)
    PrintRoutine Picture1, dRatio
End Sub

```

- **ModDraw**

```

oldfont = SelectObject(Canvas.hdc, newfont)

' Display the text.
' Note that the X and Y are in the scale units of the canvas,
' so if we are on a form, by default, it will be twips.
' No calculation to Pixels is necessary!

' The pivot point of the box is the top-left corner. You may
' need to do some extra calculations to move the pivot point.
'(x and y would have to be different for each call.)
Canvas.CurrentX = X
Canvas.CurrentY = Y
Canvas.Print txt

' Restore the original font.
newfont = SelectObject(Canvas.hdc, oldfont)

' Free font resources (important!)
DeleteObject newfont
End Sub

```

- **Module1**

```

Function FindFile$(srcPath$, Filename$)
    Dim DirReturn As String, i As Integer
    Dim subdirs$
    Dim filefound$
    Dim path$

    If Right$(srcPath, 1) <> "\" Then srcPath = srcPath & "\"
    srcPath = UCase$(srcPath)
    Filename$ = UCase$(Filename$)

    ' Agora procura pelo arquivo
    DirReturn = Dir(srcPath & Filename, 0)
    If DirReturn$ <> "" Then
        Kill srcPath$ & Filename$
        Exit Function
    End If

End Function

```

- **modWinHelp**

```
Public Const HELP_CONTEXTNOFOCUS = &H108&
Public Const HELP_POPUPID = &H104&
Public Const HELP_FINDER = &HB&
Public Const HELP_FORCEFILE = &H9&
Public Const HELP_HELPONHELP = &H4
Public Const HELP_INDEX = &H3
Public Const HELP_KEY = &H101
Public Const HELP_MULTIKEY = &H201&
Public Const HELP_PARTIALKEY = &H105&
Public Const HELP_CLOSEWINDOW = &H107&
Public Const HELP_QUIT = &H2
Public Const HELP_SETCONTENTS = &H5&
Public Const HELP_SETINDEX = &H5
Public Const HELP_SETWINPOS = &H203&
Public Const HELP_SETPOPUP_POS = &HD&
Public Const HELP_TCARD = &H8000&
Public Const HELP_TCARD_DATA = &H10&
Public Const HELP_TCARD_OTHER_CALLER = &H11&
Public Const HELP_WM_HELP = &HC&

Public Const HELPMMSGSTRING = "commdlg_help"
```