

UNIVERSIDADE FEDERAL DE PELOTAS
INSTITUTO DE FÍSICA E MATEMÁTICA
CURSO DE BACHARELADO EM INFORMÁTICA

**Sistema de Visualização de Dados
para o
Centro de Pesquisas Meteorológicas
da UFPel**

por

ISABEL CRISTINA SIQUEIRA DA SILVA

Dissertação do Projeto de Conclusão submetida à avaliação,
como requisito parcial para a obtenção do grau de
Bacharel em Informática – ênfase Sistemas de Computação.

Anal. Sist. Carlos Augusto Moreira dos Santos
Orientador

Prof. Maurício Nunes Porto, Espec.
Co-Orientador

Pelotas, dezembro de 2000.

UNIVERSIDADE FEDERAL DE PELOTAS

Reitora: Prof^a. Ingelore Scheunemann de Souza

Pró-Reitor de Graduação: João Nelci Brandalise

Diretor do Instituto de Física e Matemática: Prof. Amauri de Almeida Machado

Coordenador do Curso de Informática: Prof. Gil Carlos Medeiros

Bibliotecário-Chefe da Biblioteca Setorial das Ciências Exatas: Ubirajara Cruz

A procura é incessante, mas o encontro é casual.
Anônimo

Agradecimentos

Para a realização do presente trabalho contei com a ajuda de várias pessoas, as quais colaboraram nos mais variados aspectos. A atenção dispensada, o apoio, os ensinamentos e a amizade são alguns exemplos da dedicação e do companheirismo que recebi e que serviu como incentivo durante todo o período de desenvolvimento deste projeto.

Agradeço primeiramente, e como não poderia deixar de ser, ao meu orientador Carlos A. M. Santos, que, atenciosamente, dedicou muito do seu pouco tempo disponível ao desenvolvimento do meu trabalho, depositando em mim sua confiança e me despertando novas expectativas quanto à área da Computação Gráfica.

Outra pessoa a quem gostaria de agradecer é o meu co-orientador Maurício N. Porto, que, na reta final, deu significativa contribuição para a consolidação e adequação do projeto à sua proposta inicial.

Agradeço, também, ao meteorologista Júlio Marques pela sua especial atenção dedicada a este projeto, no que diz respeito às inúmeras dúvidas que lhe levei relativas à área da Meteorologia; e ao analista de sistemas Rogério Silva, que me auxiliou significativamente na aprendizagem do software GrADS.

Por fim, agradeço ao amigo Ricardo Galli, pela indicação do Carlos Santos; agradeço à professora Eliane Diniz e à Carla Freitas pela motivação inicial que me levou a dar continuidade na linha de pesquisa para a qual foi voltada este projeto.

A todos, meus mais sinceros agradecimentos!

Sumário

Lista de Figuras	7
Lista de Abreviaturas	8
Resumo	9
Introdução	10
1.1 Motivação do Trabalho	10
1.2 Funcionamento do CPMet/UFPeI	11
1.3 Objetivos	11
1.4 Organização do Texto	12
2 Considerações Sobre Visualização Científica	13
2.1 Conceito	13
2.2 Taxonomia da Visualização Científica	14
2.3 Visualização Científica na Meteorologia	15
3 Interfaces Homem-Máquina	16
3.1 Interfaces e seus Usuários	16
3.2 Desenvolvimento e Adequação	16
3.2.1 Aspectos Ergonômicos	17
3.2.2 Principais Dificuldades	17
3.3 Arquitetura Model-View-Controller (MVC)	17
3.3.1 Definição da Estrutura MVC	18
3.3.2 Independência entre Componentes	19
3.4 Ferramentas para o Desenvolvimento de Interfaces	19
3.5 Algumas Considerações sobre o X Window System	20
4 Fundamentos de um Sistema Meteorológico	21
4.1 Estrutura	21
4.2 Ferramentas para Visualização Meteorológica	22
4.2.1 Ferret	23
4.2.2 GrADS	25
4.2.3 MetAP	26
5 Sistema de Visualização de Dados Meteorológicos para o CPMet	29
5.1 Implementação	29
5.1.1 Ambiente de Trabalho: Hardware	29
5.1.2 Ambiente de Trabalho: Software	29
5.1.3 Armazenamento dos Dados	30
5.1.4 Sistema Voltado à Pesquisa	30
5.1.5 Variáveis Principais	30
5.2 Apresentação	31
5.2.1 Adequação do Sistema ao Conceito MVC	31
5.2.2 Importância do Design	33
5.2.3 Descrição das Janelas que Compõem a Interface	34

6	Conclusões	41
6.1	Resultados Obtidos	41
6.2	Principais Dificuldades Encontradas	42
6.3	Perspectivas para o Futuro	42
Anexos		43
	Anexo 1: Código Fonte Parcial do Sistema de Visualização	45
	Anexo 2: Conteúdo do Arquivo pgcfctl	48
Glossário		49
Referências Bibliográficas		50

Lista de Figuras

2.1	Taxonomia da Visualização	14
3.1	Arquitetura MVC	18
3.2	Arquitetura X Window System	20
4.1	Modelos de dados, baseado no esquema de Günter e Voisard	22
4.2	Janela principal da interface Motif para Ferret	24
4.3	Janela principal da interface WWW para Ferret	24
4.4	Tipos de dados no banco de dados meteorológico	27
4.5	Visão esquemática das janelas do MetAP	28
5.1	Trecho de código encarregado de mostrar a vista	32
5.2	Trecho de código responsável pela implementação das interface	33
5.3	Janela principal	34
5.4	Janela de seleção das variáveis	35
5.5	Janela de seleção do horário	36
5.6	Janela de seleção da pressão	36
5.7	Janela de composição do dia	37
5.8	Janela de composição do mês	37
5.9	Janela de composição do ano	38
5.10	Janela para a visualização das variáveis temperatura e vento	39
5.11	Janela para a visualização das variáveis com o botão “zoom” ativado	40

Lista de Abreviaturas

ANSI	American National Standards Institute
API	Application-Programs Interface
BUFR	Binary Universal Form for the Representation of meteorological data (OMM)
CG	Computação gráfica
CPTEC	Centro de Previsão do Tempo e Estudos Climáticos (INPE)
CPMET	Centro de Pesquisas Meteorológicas (UFPEL)
ECMWF	European Centre for Medium-Range Weather Forecasts
EMBRAPA	Empresa Brasileira de Pesquisas Agropecuárias
FTP	File Transfer Protocol
GOES	Geostationary Operational Environmental Satellites
GrADS	Grid Analysis and Display System
GRIB	Gridded Binary (OMM)
GTS	Global Telecommunications System (OMM)
LMS	Laboratório de Meteorologia Sinótica
MetAP	Meteorogister Arbeitsplatz
MVC	Model-View-Controller
INPE	Instituto Nacional de Pesquisas Espaciais
NCEP	National Center for Environmental Prediction (EUA)
NEONS	Naval Environmental Operational Nowcasting System (NRS)
NetCDF	Network Common Data Format
NOAA	National Ocean and Atmospheric Administration (EUA)
NRS	Naval Research Laboratories (Marinha dos EUA)
OMM	Organização Meteorológica Mundial (WMO)
PC	Personal Computer
PI	Processamento Digital de Imagens
RADAR	Radio Detection and Ranging
RENPAc	Rede Nacional de comunicação de Dados por Comutação de Pacotes
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Structured Query Language
SM	Sistema Meteorológico
TMAP	Thermal Modeling and Analysis Project
UFPEL	Universidade Federal de Pelotas
VISC	Visualização em Computação Científica (VC)
VC	Visualização Científica (VISC)
WMO	World Meteorological Organization (OMM)
WWW	World Wide Web

Resumo

Este trabalho apresenta um estudo das técnicas utilizadas na construção de aplicações de Visualização Científica, com ênfase à Meteorologia, discutindo, de maneira introdutória, algumas abordagens que serviram de base para a fundamentação da ferramenta proposta, em suas várias etapas de implementação.

Em um recente estudo que envolveu as atividades do CPMet, foi constatado que o sistema de visualização existente possuía sua utilidade limitada à atividade operacional (elaboração da previsão) e não servia como ferramenta de pesquisa, visto que não permitia que fossem realizadas consultas a dados referentes a períodos de tempo anteriores à data corrente.

Partindo desta observação, desenvolveu-se um sistema de visualização de dados meteorológicos que torna possível a realização de pesquisas meteorológicas a partir da escolha de uma data específica. As informações estão armazenadas em uma hierarquia de diretórios em disco, organizada em *dia*, *mês* e *ano*.

O projeto foi desenvolvido junto ao CPMet/UFPel (Centro de Pesquisas Meteorológicas da UFPel) e sua contribuição para o mesmo está em fornecer uma interface, propriamente desenvolvida, de controle e manipulação de dados climáticos e na síntese (ou simulação visual) destes. A partir desta interface, o meteorologista passa a ter a possibilidade de recuperar os dados já analisados e de realizar a conversão destes para a mesma forma gráfica utilizada na operação. Objetiva, com isso, servir de ferramenta para uso em pesquisas que virão a ser desenvolvidas no CPMet.

Capítulo 1

Introdução

1.1 Motivação do Trabalho

A Visualização Científica (VC) exerce um papel de fundamental importância para as mais diversas áreas cujos trabalhos de pesquisa são auxiliados por computador [3]. Sua principal meta é a obtenção do entendimento de um determinado problema a partir de sua representação gráfica.

Atualmente, a VC constitui uma área de pesquisa científica emergente, que envolve uma série de conceitos e propriedades igualmente importantes, como interfaces com o usuário, arquiteturas cliente-servidor, modelagem de dados, entre outros. Identificou-se, desta forma, a necessidade de ampliar os conhecimentos a respeito do tema, uma vez que a sua abordagem no Curso de Informática é insatisfatória, bem como sua discussão em trabalhos de conclusão realizados anteriormente.

Este trabalho destina um enfoque especial à área da Meteorologia, devido à existência na UFPel de um Centro de Pesquisas Meteorológicas que fornece previsão do tempo à comunidade, tanto em âmbito local quanto estadual.

Considerando a relevância que a previsão alcançou recentemente, muitas atividades econômicas dependem dela e, para o meteorologista, é necessário que existam ferramentas de trabalho flexíveis e adequadas para este propósito [17].

Em um estudo a respeito das atividades do CPMet, foi constatado que o sistema de visualização existente possuía sua utilidade limitada à atividade operacional (elaboração da previsão) e não servia como ferramenta de pesquisa, visto que não permitia que fossem realizadas consultas a dados referentes a períodos de tempo anteriores à data corrente. Devido a esta observação, a ferramenta, descrita neste trabalho, foi proposta e desenvolvida de modo a permitir que o meteorologista tenha a possibilidade de recuperar os dados já analisados e possa convertê-los para a mesma forma gráfica usada na operação.

Como o CPMet carece de um banco de dados contendo informações meteorológicas, a armazenagem dos arquivos de dados é realizada, ainda, organizando-os em uma hierarquia de diretórios em disco (a maioria no formato GRIB [4]). Futuramente, pretende-se a utilização de um banco de dados mais sofisticado, possivelmente baseado em um sistema de consultas via SQL, quesito levado em consideração durante o desenvolvimento da nova ferramenta.

1.2 Funcionamento do CPMet/UFPel

O CPMet (Centro de Pesquisas Meteorológicas) é uma unidade de ensino, pesquisa e extensão, subordinada à Faculdade de Meteorologia da UFPel. Entre as atividades realizadas pelo CPMet está a emissão diária de dois boletins de previsão do tempo. O primeiro é divulgado às 10h, com a previsão para o dia, e o segundo às 16h, com a previsão para o dia seguinte.

A equipe responsável pela previsão é constituída por seis meteorologistas, que se revezam por semana. Os boletins gerados são fornecidos para diversos meios de comunicação, como a televisão, jornais e rádios. Existe, ainda, a possibilidade do acesso aos boletins pela RENPAC (Rede Nacional de Comunicação de Dados por Comutação de Pacotes), através do sistema VisiMet. Este sistema, atualmente, dispõe de mais de 700 usuários cadastrados.

Na elaboração da previsão, os meteorologistas do CPMet utilizam, além de resultados de modelos matemáticos previamente preparados (global NCEP/NOAA; global e regional do CPTEC/INPE), imagens dos satélites GOES e TIROS, sondagens realizadas pelo radar meteorológico e os dados observados na Estação Agroclimatológica existente no campus da UFPel (mantida pelo convênio existente entre a Universidade e a EMBRAPA) [17].

A visualização dos resultados dos modelos fica a cargo do software GrADS (ver seção 4.2.2), com a utilização dos recursos oferecidos para tanto, através do conjunto de *scripts* escritos para este software e, para a visualização e tratamento das imagens de satélite, utiliza-se o software *ImageMagick*, além de um programa de conversão desenvolvido no próprio CPMet.

1.3 Objetivos

Este projeto objetivou o estudo e o desenvolvimento do domínio de técnicas relacionadas à Visualização Científica para que, desta forma, pudessem ser identificados os principais problemas existentes na visualização de grandes volumes de dados pertencentes à data corrente ou a períodos anteriores a esta.

Partindo deste estudo inicial, foi implementada uma aplicação destinada à utilização por parte dos meteorologistas do CPMet em seu trabalho diário e provendo-os de certas facilidades relacionadas à realização de pesquisas envolvendo previsões meteorológicas. Uma das premissas básicas, levadas em consideração no desenvolvimento desta ferramenta, foi que houvesse compatibilidade com o software existente no CPMet.

1.4 Organização do Texto

De modo a fornecer um satisfatório entendimento da abrangência do projeto em questão, bem como a obtenção dos objetivos expostos anteriormente, o texto foi dividido considerando as várias etapas que serviram de base para o desenvolvimento do sistema.

Além deste capítulo introdutório, o texto apresenta, no capítulo 2, alguns conceitos necessários a uma melhor compreensão da Visualização Científica e de sua importância na área da Computação Gráfica.

A seguir, no capítulo 3, faz-se referência à importância do estudo e desenvolvimento de interfaces homem-máquina e suas conseqüências relacionadas aos usuários, bem como as ferramentas usadas para tanto.

No capítulo 4, são apresentados os fundamentos envolvidos na estrutura de um sistema meteorológico, através de sua estratificação em funções específicas.

No capítulo 5, é apresentado o sistema de visualização proposto, com a descrição das suas fases de implementação e relacionando-o com os estudos apresentados anteriormente.

Por fim, o capítulo 6 traz as conclusões relativas ao desenvolvimento deste projeto, onde são apontados os resultados obtidos, as principais dificuldades encontradas e as perspectivas para o futuro.

Capítulo 2

Considerações Sobre Visualização Científica

2.1 Conceito

Com o avanço dos tempos, os mais variados campos da ciência estão voltados às novas descobertas. Dada a complexidade destas, seu estudo foi inviabilizado anteriormente pela falta de recursos computacionais apropriados que proporcionassem uma visão (simulação) do problema a ser estudado. Hoje, com o avanço da tecnologia computacional, já se obtém mais facilmente imagens e/ou simulações dos fenômenos estudados. O conceito chave que possibilitou tais avanços da ciência denomina-se *Visualização em Computação Científica (VISC)* ou, simplesmente, *Visualização Científica (VC)*.

A Visualização Científica emergiu como uma ferramenta, no campo de estudos computacionais, de fundamental importância para as mais diversas áreas da ciência como a química, a meteorologia, a medicina, entre outras.

Mas o que é realmente a VC? Vários especialistas em VC defendem o conceito que rege que “*o propósito da computação não são os números propriamente ditos, mas sim o insight que envolve a resolução de problemas, com o uso da tecnologia disponível*”. A presença de *insight* (em português, equivalente à *perspicácia*) caracteriza a facilidade e rapidez na compreensão da natureza das situações ou problemas complexos [1].

O conceito de VC envolve o entendimento de três outros conceitos, profundamente relacionados entre si, a saber: *Computação Gráfica (CG)*, *Processamento Digital de Imagens (PI)* e *Visão Computacional*. A primeira, trata da modelagem e construção de imagens; a segunda, da manipulação da imagem adquirida; e, a última, da análise dos diversos aspectos da imagem, sendo que todas elas fundamentam-se no caráter exploratório.

Além destes três campos apresentados, existem também outros não menos importantes: *projeto auxiliado por computador*, *processamento de sinais* e *estudo de interfaces de usuários* [3].

Com base nestas definições, a VC visa o entendimento de um determinado problema a partir da visão de uma representação gráfica. Como ferramenta, destina-se à visualização de problemas antes considerados imperceptíveis e, como tecnologia, oferece um melhoramento na relação homem-máquina através de interfaces.

2.2 Taxonomia da Visualização Científica

A visualização envolve o entendimento e a síntese de imagens, isto é, é uma ferramenta tanto para a interpretação de imagens em um computador quanto para sua geração (Fig. 2.1).

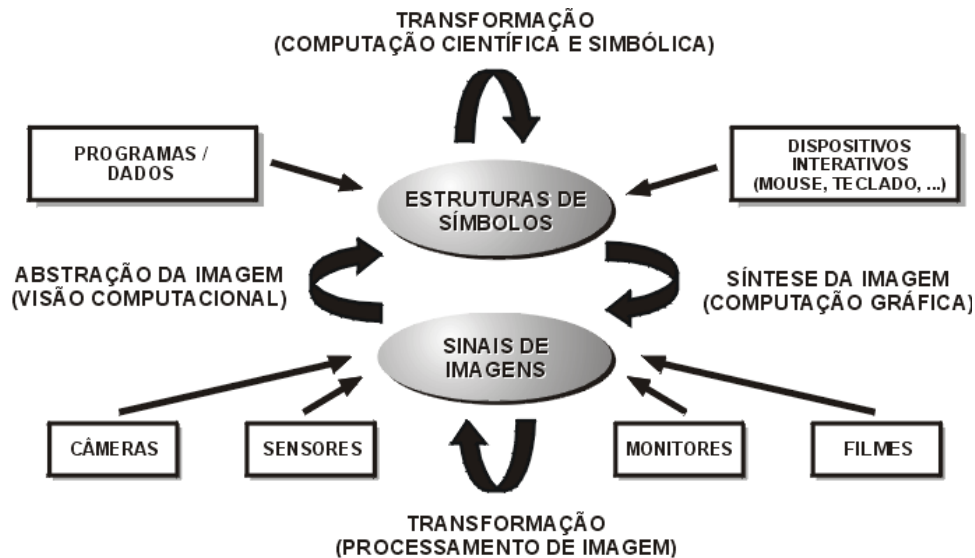


Fig. 2.1: Taxonomia da Visualização [3]

A VC molda geometricamente a informação que é representada internamente como estruturas de dados (estruturada), com base em símbolos, possibilitando aos pesquisadores a observação de suas explorações. Pode-se dizer que a visualização e as ferramentas de análise envolvem três passos principais: o escopo da visualização é especificado; o modelo geométrico é sintetizado no computador; e este modelo é renderizado, ou seja, simulado graficamente [3].

Como exemplo de renderização, pode-se mencionar uma linha de corrente, pertencente a um campo de fluxo, passando através de um ponto no espaço. A interface de usuário especifica quais os pontos que serão atravessados pela linha. Um algoritmo da linha de corrente faz um levantamento dos elementos da linha, acessando cada ponto do vetor pertencente ao longo desta. Ao final da operação, estes elementos são renderizados em um sistema gráfico 3D.

As técnicas de renderização (2D ou 3D), de sombreamento, de transparência e de perspectiva são aplicadas de acordo com o tipo de captura de imagem utilizada. Esta imagem também pode ser gerada a partir de um sistema de projeção, moldando-a de acordo com os interesses que envolvem a sua utilização.

Em Computação Gráfica, uma das principais dificuldades na implementação de aplicações visuais se dá na organização das operações necessárias para converter objetos definidos em um espaço 3D para um espaço 2D (tela do computador).

2.3 Visualização Científica em Meteorologia

Considerando que o hardware e o software evoluíram rapidamente junto com as modernas estações de trabalho, cresceu a necessidade do desenvolvimento de programas de VC voltados à área da Meteorologia, uma vez que os prognósticos meteorológicos apresentam grande importância na detecção e prevenção de eventos climáticos.

Para a investigação destes problemas, se fazem necessárias ferramentas de visualização destinadas à aquisição e simulação de fenômenos da atmosfera terrestre (temperatura, velocidade do vento, umidade relativa, etc.).

Na simulação das variáveis climáticas, existem algumas técnicas especialmente desenvolvidas para este propósito. Dentre estas, pode-se citar a *renderização de volume*, o *contorno texturizado de superfícies* e a *renderização de vetores de campo* [13].

Além das técnicas de visualização que compõem um software de sistema meteorológico, a interface de usuário constitui outro fator de extrema importância no contexto da VC aplicada a esta área. A interface deve apresentar eficiência, clareza e abrangência, oferecendo a possibilidade de controle e manipulação de parâmetros de visualização e, em determinados casos, suporte para desenvolvimento de macros e funções mais complexas.

Existem, atualmente, vários sistemas para visualização de dados meteorológicos que implementam os conceitos mencionados. À medida que avançam as pesquisas no meio, surgem novos softwares, os quais passam a oferecer uma nova gama de opções de trabalho, com inovações que contribuem gradativamente para otimizar o trabalho dos seus usuários.

Como exemplos destes softwares, pode-se citar: GrADS [5], MetAP [16], Ferret [8], Metview [11], Vis5d [9], entre outros. Os três primeiros são abordados mais detalhadamente no capítulo 4, onde é realizada uma análise sobre suas principais características e contribuições no campo das pesquisas meteorológicas.

Capítulo 3

Interfaces Homem-Máquina

3.1 Interfaces e seus Usuários

O computador passou a exercer, nos últimos tempos, um importante papel nas atividades diárias dos seus usuários, seja no campo profissional ou no lazer. As pessoas, aos poucos, foram se familiarizando com a nova tecnologia que passou a representar uma revolução na realização de tarefas antes consideradas repetitivas, complexas e/ou demoradas.

Porém, ainda que o computador proporcione inúmeras facilidades, há quem prefira o seu antigo sistema de trabalho, em que não existia entre o usuário e o seu objetivo um intermediário denominado *interface*, o qual, muitas vezes, impede a realização das mais simples rotinas de trabalho.

A interface representa o meio de comunicação entre usuário e máquina. É através dela que se dá a interação das pessoas com o computador e, para que os usuários possam controlar, usar e manipular aplicações de software, é de fundamental importância a qualidade da interface apresentada, como também a sua adequação à arquitetura de sistema na qual a mesma está inserida.

3.2 Desenvolvimento e Adequação

O principal problema na interação entre usuários e computador reside no fato de que vários aplicativos são criados de profissionais para profissionais, não oferecendo grandes facilidades de uso para os leigos da área. Isto constitui um ponto negativo em certos softwares que, para alcançar determinada difusão no mercado, são acompanhados por enormes manuais, carregados de detalhamentos, ou por manuais incompletos [14].

Ao mesmo tempo em que os usuários procuram uma ferramenta maleável, que supra suas principais necessidades, eles também esperam poder sentar em frente ao seu micro e partir para a solução dos seus problemas, com simplicidade e satisfação. Eles não estão interessados no processo de geração de gráficos, em aberturas de arquivos, nos *traps* emitidos pelos processos. Sua atenção está voltada para a sua área de interesse e o micro deverá se apresentar como um poderoso aliado no desenvolvimento de suas atividades, não como uma barreira.

Contudo, isto não implica no fato de que uma boa ferramenta não deva apresentar certa complexidade. À medida que a abrangência do sistema aumenta, se faz necessária a representação de muitos comandos. A questão chave aqui é a representação destes, que deverá apresentar certas familiaridades com a representação de outros comandos, em outras ferramentas similares já conhecidas, proporcionando uma identificação mais facilitada por parte dos usuários.

3.2.1 Aspectos Ergonômicos

A ergonomia constitui um fator de extrema importância a ser levado em consideração durante o processo de desenvolvimento de um software, podendo reduzir, consideravelmente, as dificuldades geralmente encontradas por parte dos usuários. Através da ergonomia, obtém-se uma maior eficiência na interação homem-máquina e, no contexto deste projeto, uma importante contribuição para possibilitar maior e mais facilitado desempenho das tarefas dos previsores.

Uma definição para ergonomia rege que *“ergonomia é o conjunto de conhecimentos científicos relativos ao homem e necessários à concepção de instrumentos, máquinas e dispositivos que possam ser utilizados com o máximo de conforto, segurança e eficiência”* [12].

A ergonomia tem contribuído substancialmente para o desenvolvimento de interfaces homem-máquina que apresentem maior usabilidade, pois a abordagem ergonômica busca privilegiar a lógica de utilização (aplicação do ponto de vista do usuário), que busca no software um conjunto coerente com a execução de sua tarefa, ao invés da lógica de funcionamento, verificada especialmente na estruturação dos sistemas e elaborada a partir da visão de projetistas de software.

3.2.2 Principais Dificuldades

Como a maioria dos softwares, não existem padrões ou regras gerais para o desenvolvimento e implementação de interfaces. Em decorrência disto, pode-se destacar certas dificuldades enfrentadas pelos projetistas ao desenvolver um determinado sistema [14]:

- *falta de conhecimento das atividades realizadas pelos usuários a que se destina, o que pode acarretar certas dificuldades enfrentadas pelos projetistas em relação ao objetivo para o qual foi proposto;*
- *ausência de clareza e simplicidade de operação, proporcionando, dessa forma, menor interatividade entre usuário e máquina;*
- *falta de objetividade das ferramentas, levando o usuário à incerteza e aumentando a complexidade do sistema;*
- *necessidade de adequação do sistema ao multiprocessamento e processamento em tempo real;*
- *falta de ambientes destinados ao desenvolvimento de interfaces;*
- *dificuldade no emprego da modularização no software de interface, ou seja, empregar uma clara e eficiente subdivisão da interface em várias janelas, cada uma encarregada da execução de uma tarefa dentro do contexto ao qual está inserida.*

3.3 Arquitetura Model-View-Controller (MVC)

A arquitetura MVC constitui uma técnica de projeto de software voltada a proporcionar uma maior adequação e portabilidade do escopo para o qual os modelos (instâncias) de objetos estão voltados, realizando, assim, um isolamento entre aspectos que referem-se ao domínio do problema (estrutura e atributos) e aspectos voltados ao domínio computacional (interação do sistema com o modelo de objeto).

A técnica *Model/View/Controller* – Modelo/Vista/Controlador – foi introduzida e popularizada através da linguagem de programação orientada a objetos Smalltalk, a partir da versão Smalltalk-80. Com base no paradigma MVC, esta versão teve um importante papel para a programação por implementar interfaces multijanelas interativas [2].

A MVC é voltada para a operação computacional que envolva múltiplas e sincronizadas representações do mesmo tipo de dados [19]. Ela permite o tratamento uniforme destes dados e implementa o conceito de divisão de tarefas entre controladores, modelos e vistas, possibilitando independência funcional entre eles.

3.3.1 Definição da Estrutura MVC

O processamento dos dados é realizado, coordenadamente, pelos três componentes da arquitetura, cada qual encarregado de uma tarefa específica.

O *modelo* é o objeto que representa e gerencia o comportamento dos dados da aplicação, provendo informações sobre estes (geralmente para a vista) e realizando mudanças (na maioria das vezes, requeridas pelo controlador). No entanto, o modelo não possui uma referência interna ao controlador ou à vista. No caso específico da linguagem Smalltalk, por exemplo, existe um *dicionário de identidade* encarregado de fazer a ligação entre o modelo e a vista, notificando este último das possíveis mudanças no primeiro.

A *vista* é a parte encarregada da disponibilização visual do dado referenciado pelo modelo. Ela produz a representação do objeto, possibilitando alterações nesta e disponibilizando-a para o usuário.

Por fim, o *controlador* é a entidade responsável pela interação do usuário com o modelo, através de uma interface gráfica ou de um subprocesso que roda em “background”, por exemplo. Ele interpreta a entrada de dados, através do teclado ou mouse, indicando ao modelo ou à vista as alterações necessárias.

O paradigma MVC permite que sejam implementados, simultaneamente, mais de uma vista e/ou controle para um mesmo modelo de dados, possibilidade esta existente em decorrência do isolamento dos aspectos relativos ao comportamento de cada componente. Pode-se considerar que a vista e o controlador “prestam serviços” ao modelo (Fig. 3.1).

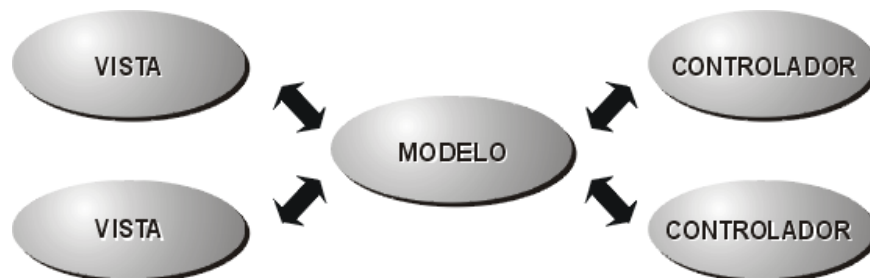


Fig. 3.1: Arquitetura MVC [19]

3.3.2 Independência entre Componentes

Em um projeto modular, cada parte do programa é desenvolvida separadamente, de acordo com a sua funcionalidade dentro do contexto no qual está inserida e, ao final, todas são reunidas em um único bloco, com a possibilidade de ser substituído o componente que não apresentar a adequação requerida, sem que esta alteração cause impacto aos demais componentes. O diferencial do MVC, neste sentido, está no isolamento de aspectos de um mesmo modelo de objetos e não simplesmente à condição da “decomposição funcional”.

Com a definição de uma clara separação entre os aspectos relativos ao escopo computacional e ao escopo do problema, possíveis modificações que ocorram em cada domínio poderão ser resolvidos separadamente, sem que isto afete o sistema como um todo [19] e interfira diretamente no modelo de objetos. Esta é a principal característica da flexibilidade do sistema, não o tornando inteiramente inativo na possibilidade de mudanças específicas em um ou outro componente, desde que sejam preservadas as interfaces entre eles.

3.4 Ferramentas para o Desenvolvimento de Interfaces

Juntamente com o notável desenvolvimento da Computação Gráfica no meio tecnológico, cresce a necessidade da utilização de um padrão gráfico eficiente e portátil à diversas plataformas. As linguagens de programação atuais não oferecem suporte para o desenvolvimento de interfaces de usuários. Seus recursos destinados à implementação de primitivas gráficas, para entrada e saída de dados, não são suficientes para a obtenção de uma interface eficiente e amigável [14]. Uma ferramenta ideal para a construção de interfaces deve permitir que estas apresentem estabilidade e boa definição, além do encapsulamento, ou seja, a ocultação da informação.

Devido a este fato, um grande número de ferramentas destinadas à construção de interfaces de usuários têm sido desenvolvidas, embora não exista uma biblioteca (*toolkit*) padrão para tanto, razão pela qual muitas podem, ainda, não oferecer as mesmas facilidades de uso. Os *toolkits* existentes variam desde simples conjuntos de *widgets*, usados em uma única aplicação, até bibliotecas sofisticadas, carregadas de recursos.

Algumas destas ferramentas são voltadas, especificamente, para a implementação de interfaces de programação para o *X Window System* (descrito na seção 4.4), com base no padrão deste. Dentre estas, pode-se citar o *Motif*, da OSF (Open Software Foundation) – padrão nos sistemas Unix comerciais –, o *Athena*, pertencente ao MIT (Massachusetts Institute of Technology), e o *Open Look*, da Sun e AT&T [15].

Pode-se destacar, ainda, outros *toolkits* de grande difusão no contexto do desenvolvimento de interfaces, como o *Gtk* (biblioteca escrita em C e fortemente influenciada pelo *Motif*), da FSF, o *Qt* (em C++ e sob influências do Windows), da Troll Tech, e o *Tcl/Tk* (onde *Tcl* constitui uma linguagem de scripts e *Tk* o seu *toolkit* gráfico independente), a qual suporta diversas plataformas como X Window, Microsoft Windows e Macintosh.

A seguir, são apresentadas algumas considerações sobre o *X Toolkit*, uma biblioteca integrante do X Window System, voltada à criação de interfaces gráficas e que serve de ponto de partida para a implementação das ferramentas citadas anteriormente.

3.5 Algumas Considerações sobre o X Window System

O X Window System foi desenvolvido com o intuito de se tornar um padrão, uma especificação de software, para a construção de aplicações gráficas interativas. Ele surgiu como resultado de pesquisas voltadas à construção de aplicações multiprocessadas distribuídas [15].

Por possuir arquitetura do tipo cliente-servidor, e devido à independência entre aplicação e hardware gráfico, facilita a implementação de programas portáteis entre diferentes arquiteturas. Aliado a esta versatilidade de desenvolvimento, destaca-se o protocolo responsável por reger o X – o *protocolo X* – baseado em especificações formais, que garantem a estabilidade e a confiança do sistema, embora continue evoluindo por meio de extensões (Fig. 3.2).

O X não deve ser encarado como uma interface de usuário ou possuidor de uma; ele provê mecanismos para que diferentes tipo de interfaces possam ser desenvolvidas. O gerenciamento das interfaces é função dos clientes e de bibliotecas externas ao servidor X e às próprias bibliotecas do sistema.

No X, existem dois tipos de bibliotecas que simplificam o desenvolvimento de aplicações gráficas, oferecendo rotinas e tipos de dados para interação com o sistema de janelas. São elas: *Xlib* e *X Toolkit*.

A biblioteca Xlib provê funções gráficas primitivas que originam interfaces de baixo nível, enquanto que a biblioteca X Toolkit, construída “sobre” a Xlib, oferece funções que tratam dos elementos constituintes da interface – os *widgets* – obtendo, assim, os mecanismos necessários à construção de interfaces de alto nível.

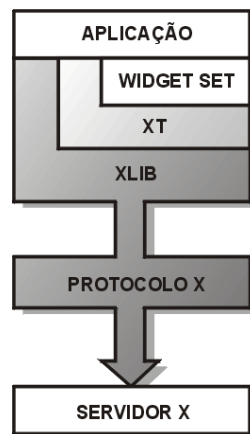


Fig. 3.2: Arquitetura X Window System (adaptado de Nye [15]).

A base do Toolkit é uma biblioteca implementada em C, conhecida por libXt. Sua principal aplicação se dá ao combiná-la com uma biblioteca de widgets pré-construídos – *widget set* – como menus, botões de comandos, barras de rolagem, que originam uma aplicação com “*look and feel*” consistente [15].

Com o uso de widgets, obtêm-se a dissociação entre o código da aplicação e o código da interface de usuário. O widget opera independente da aplicação, mas a aplicação pode alterar os atributos do widget através de invocações de funções do toolkit.

Capítulo 4

Fundamentos de um Sistema Meteorológico

4.1 Estrutura

Para que se possa entender um sistema meteorológico (SM) em sua totalidade, é necessário, primeiramente, o entendimento das diversas etapas que compõem o processo de elaboração da previsão do tempo, hierarquizadas de acordo com o tratamento necessário dispensado aos dados.

As atividades de um SM começam no *sistema de coleta de dados* (também chamado de *assimilação*), onde é realizada a aquisição das diversas condições climáticas, em intervalos regulares de tempo e em posições fixas pré-determinadas. Esta coleta é feita junto a postos de superfície e de altitude, rede de postos oceânicos fixos, navios e aviões, satélites, radares, foguetes e balões meteorológicos. As observações são reunidas pelos organismos nacionais e retransmitidas para centros mundiais. Os dados coletados são reunidos em “janelas” de seis horas, sendo que a hora central especificada é a hora sinótica, ou seja, a hora especificada, entre os centros de previsão, para a realização da previsão do tempo. A Organização Meteorológica Mundial (OMM) reúne os dados, coletados pelas instituições que elaboram as previsões do tempo a ela filiadas, e os redistribui através do *Global Telecommunications System* (GTS), uma rede de comunicações mantida pela própria OMM. Os formatos de dados padronizados pela OMM são o BUFR (para observações de estações meteorológicas) e o GRIB (para campos), ambos inadequados, sem conversão, para o processamento e visualização [17].

O segundo passo é o *pré-processamento*, ou seja, a recepção crítica[♦] dos dados provenientes das diversas fontes de coleta. Aqui é realizada aceitação ou rejeição dos dados, com a eliminação de leituras inválidas.

A seguir, os dados vindos do sistema de aquisição e os resultados das tarefas operacionais juntamente aos modelos matemáticos são armazenados em um banco de dados apropriado para tanto e controlado por um *sistema de gerenciamento de banco de dados* (SGBD). Os dados também podem ser organizados em uma hierarquia de diretórios, na ausência de um SGBD, como é feito no CPMet, por exemplo. Para a indexação dos dados armazenados são utilizados *metadados*, próprios para cada informação e que contêm informações sobre o formato e o conteúdo destes.

Um esquema proposto por Günter e Voisard [7] consiste em um esquema de uso de metadados no domínio da informação geográfica e ambiental. Eles propõem um modelo de dados de três vias (Fig. 4.1), onde os *objetos ambientais* constituem objetos do mundo real, os *objetos de dados ambientais* representam entidades abstratas, as quais podem ser manipuladas por computadores ou tomadores de decisão e têm seu formato e conteúdo especificados por um objeto metadado, os *metadados ambientais*.

[♦] Consistência, avaliação da qualidade dos dados.

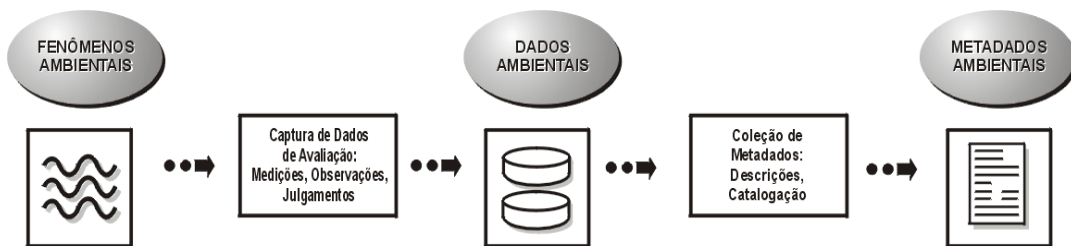


Fig. 4.1: Modelos de dados, baseado no esquema de Günter e Voisard

Com os dados devidamente armazenados, passa-se à fase de *análise*, etapa encarregada da preparação dos dados para que estes sirvam de entrada ao modelo, que pode exigir uma divisão temporal (e/ou espacial) e unidades de medição diferentes daquelas encontradas nos dados originais.

Após, faz-se a *modelagem* e a *simulação* dos fenômenos meteorológicos. São utilizados dois tipos principais de modelos: *globais*, que abrangem todo o planeta, e *regionais*, restritos a uma determinada área. O formato de saída mais utilizado é o sistema de 5 dimensões – latitude, longitude, altitude, tempo e campo. Muitos modelos assumem o formato GRIB, o qual permite o uso de grades e, por suportar um número considerável de sistemas de projeção (Polar Estereográfica, Gaussiana, etc.), os softwares de visualização necessitam, freqüentemente, da aplicação de conversões, a partir deste formato, para formatos que estejam aptos a manipular.

Por fim, é realizada a tarefa da *previsão do tempo*, através do uso de ferramentas apropriadas para a visualização dos dados e que permita aos previsores avaliar, com um bom grau de segurança, as condições do tempo. Pode-se somar a esta fase a monitoração das condições climáticas, com a emissão de alertas, e com operação em tempo real (ou próximo disto).

4.2 Ferramentas para Visualização Meteorológica

Um software de visualização e manipulação de dados meteorológicos deve apresentar uma série de características para ser considerado adequado e eficiente à atividade a que se propõe. Além de técnicas de visualização e de uma interface de usuário apropriada, existem outras características não menos importantes que, se acrescentadas ao software, elevam a sua eficiência, como: *agendamento de tarefas* (por parte do previsor), *operação em tempo real*, *suporte à programação*, com invocação de rotinas externas em outras linguagens, *operação em ambiente distribuído*, *suporte à comunicação com SGBD*, entre outras [17].

A seguir, são abordados três softwares para simulação gráfica e seus principais aspectos, como origem, contribuições, pontos positivos e negativos e interface de usuário.

4.2.1 Ferret

- **Descrição**

Ferret é uma ferramenta de visualização integrada à análise de dados climáticos e baseada em estações de trabalho. Foi desenvolvida no *Thermal Modeling and Analysis Project* (NOAA/TMAP) como auxílio a pesquisas de modelagem numérica por parte de físicos, que estudam as interações oceano-clima [8].

A principal contribuição desta ferramenta está na unificação do gerenciamento dos dados com a análise destes e a apresentação dos respectivos gráficos. Sua flexibilidade de trabalho, aliada à automação, constitui um eficiente sistema de auxílio no tratamento de grandes e complexos conjuntos de dados, podendo prover diferentes estilos de gráficos científicos, como contornos e diagrama de vetores.

Modelos de dados simples, como grades de variáveis multi-dimensionais, podem ser utilizadas para integrar gráficos científicos bi-dimensionais, análises baseadas em transformações simétricas-axiais (rotação simétrica em torno do próprio eixo), hierarquias criadas por usuário, na definição de variáveis, e arquivos auto-descritivos (arquivos que contém dados e metadados que os descrevem [7]). Esta integração é o fator principal para o elevado nível do sistema Ferret, um ambiente de visualização tanto para o uso de cientistas da área como também por especialistas da computação.

- **Interfaces**

Ferret pode ser apresentado em três tipos de interfaces diferentes, dependendo da natureza da operação a que se destina. São elas: *X-Window*, *World Wide Web* (interfaces gráficas) e a interface por linha de comando.

A interface X-Window (Fig. 4.2), implementada a partir do toolkit Motif (ver seção 3.4), caracteriza-se pelo uso de *widgets*, como botões, barras de rolagem e caixas de texto. Por apresentar estes componentes, é uma interface “amigável” e de simples operação. O usuário determina o modo de visualização da variável (latitude, longitude, tempo) e o tipo de saída desejada (linhas, contornos, etc.).

A interface World Wide Web (WWW) (Fig. 4.3) permite o acesso remoto de dados via Ferret por uma conexão Internet. Sua principal aplicação está na área naval, porém apresenta o problema de que dados oceanográficos são demasiadamente grandes para serem transferidos pela Internet, e, uma vez que é realizada a transmissão, ainda pode haver incompatibilidade entre arquivos de dados, por falta de uma padronização de formato.

Com a interface por linha de comando, em terminais ANSI, Ferret pode ser operado através de 20 comandos (PLOT, CONTOUR, LIST, ...) e uma variedade de subcomandos. A linguagem de comando do Ferret é também uma linguagem de *script*, onde grupos de comandos podem ser guardados em um arquivo (*script*) e executado com o comando GO, com a passagem ou não de argumentos.

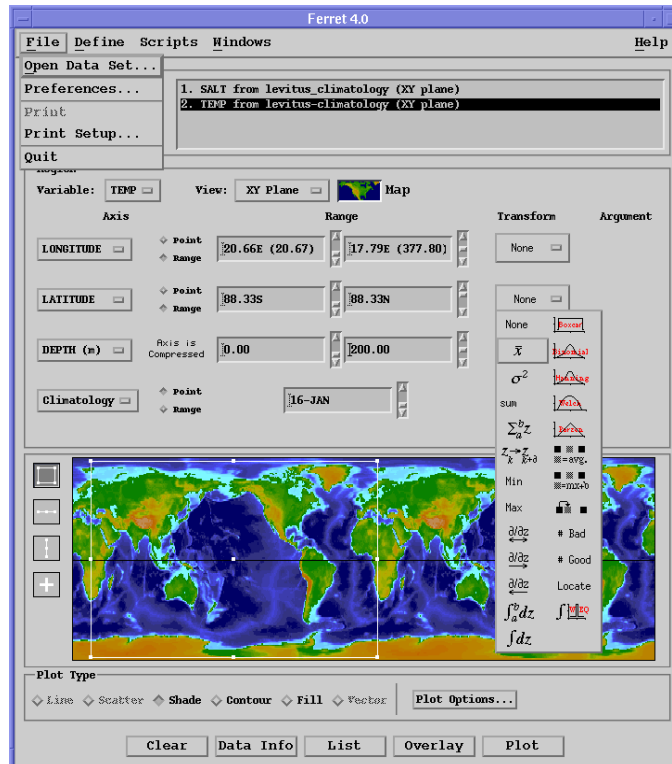


Fig. 4.2: Janela principal da interface Motif para Ferret [8]

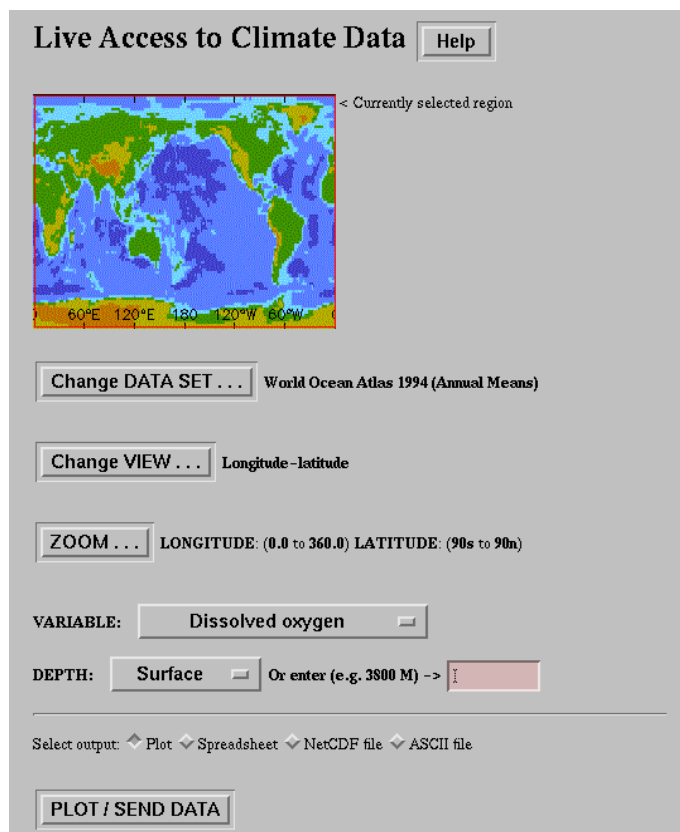


Fig. 4.3: Janela principal da interface WWW para Ferret [8]

4.2.2 GrADS

- **Descrição**

O GrADS (*Grid Analysis and Display System* – Sistema de Análise e Apresentação de Grades) foi implementado por Brian Doty e possui livre distribuição. Por apresentar grande portabilidade, pode ser utilizado nas principais estações de trabalho hoje conhecidas, o que justifica sua grande difusão junto ao meio [5].

Na área de pesquisas meteorológicas, constitui uma importante ferramenta que integra operações de acesso, manipulação e visualização de dados ambientais, além da grande interatividade oferecida.

O GrADS manipula modelos de dados em quatro dimensões distintas: latitude, longitude, nível e tempo. Suporta dois tipos básicos de dados: dados reticulados (em grade) e dados de estações (pontuais). Quanto à representação de dados, existem duas opções para a gravação: ou eles estão gravados no formato próprio do GrADS (binário) ou estão compactados em formato GRIB [4] ou NetCDF [20] e acessados através de um arquivo descritor.

Os dados e os metadados (ou informações sobre os dados) são guardados em dois arquivos separados. O arquivo de metadados (extensão *ctl*) – também chamado de arquivo descritor ou de controle – contém uma descrição completa dos dados, além do nome do arquivo que os contém. Este último, é um arquivo puramente de dados (binário, GRIB ou NetCDF), sem nenhum identificador de espaço ou de tempo. Sem o auxílio dos arquivos descritores, o GrADS teria que percorrer sequencialmente todo o GRIB cada vez que o abrisse, inviabilizando o trabalho de análise em razão do período de tempo dispendido [18].

Uma das vantagens ao comparar o GrADS a outros softwares de visualização é que, enquanto estes necessitam do uso de conversões para que possam manipular arquivos com o formato GRIB, o GrADS possui um suporte direto a dados em GRIB, dispensando o uso de conversores.

O arquivo descritor, além da especificação de um conjunto binário de dados, é constituído por uma identificação dos dados, um valor que indica a ausência ou invalidez destes, as variáveis utilizadas e a definição do domínio dentro do ambiente dimensional: *xdef* (longitude), *ydef* (latitude), *zdef* (nível) e *tdef* (tempo). As altitudes são sempre dadas em níveis de pressão atmosférica e o tempo em variações de minutos, horas, dias, meses ou anos, considerando uma determinada data inicial.

A saída gráfica pode ser definida entre várias técnicas, como linhas, barras e gráficos (contorno, sombreamento, linhas de corrente, etc.) e sua gravação é permitida nos formatos GIF, JPEG, PostScript, BMP, entre outros.

Uma das principais vantagens, na implementação em GrADS, está na possibilidade do usuário invocar a execução de programas externos. Porém, como não é oferecida uma API, não é possível incorporar funções externas (escritas em outras linguagens de programação) ao seu próprio *script*, em forma de bibliotecas.

• Interface

Como o GrADS não possui uma linguagem visual, toda operação, em um modo interativo, deve ser feita por linha de comando. É apresentado pelo programa um sinal de espera de comandos – ga-> – e uma janela gráfica, onde serão expostas as sínteses de imagens em resposta aos comandos digitados. Esta janela é tratada como se fosse uma folha de papel, podendo ser exibida no modo paisagem – 11 (horizontal) x 8.5 (vertical) polegadas –, normalmente o mais utilizado; ou retrato – 8.5 (horizontal) x 11 (vertical) polegadas –, no qual este projeto foi implementado.

Contudo, a partir da versão 1.7 do GrADS, foi incorporado ao *script* da linguagem um suporte primitivo para *widgets*, desenvolvido por Arlindo da Silva [18], baseado em X Toolkit e *widgets* Athena (ver seção 3.5). Dessa forma, podem ser implementados *widgets* como caixas de diálogos, botões e menus, além da exibição de vários tipos de gráficos, plotagem de dados e gravação de resultados.

A seleção do subconjunto do domínio de dados é realizada através do comando “set” (atribui valores aos parâmetros *lat*, *lon*, *lev* e *time* – *x*, *y*, *z* e *t*). Também com esta seleção é determinado o tipo de exibição que será oferecido: dependendo do número de dimensões variáveis, uma plotagem respectiva será realizada. Neste trabalho, adotou-se a variação de duas dimensões, o que gera o desenho de uma fatia 2D do conjunto de dados.

Antes da exibição de um gráfico, como já mencionado, é necessário selecionar o tipo de plotagem e alguns parâmetros visuais. O tipo de gráfico é obtido a partir da variação das dimensões que compõem a linguagem: quando uma única dimensão varia, um gráfico de linha é obtido; se ocorre a variação de duas dimensões, uma plotagem em contorno é gerada. Alguns tipos possíveis de gráficos são: plotagem de contorno sombreado, flechas de vetor de vento, linhas de corrente, gráficos em barras ou em linhas, aletas de vento, etc.

O GrADS oferece, ainda, a possibilidade de uma sequência de animação quando tem-se a variação de três (ao menos) dimensões. A animação é obtida através de qualquer uma das três dimensões variantes. Pelo padrão, a dimensão de animação é o tempo.

Para as saídas gráficas de vetores, linhas de correntes e aletas, as rotinas de exibição necessitam de duas grades, onde a primeira é tratada como a componente U, e a segunda, como a componente V. Ainda para vetores e correntes pode ser especificada uma terceira grade, utilizada para a geração de cores para estas componentes.

4.2.3 MetAP

• Descrição

O MetAP (meteorologist Arbeitsplatz – *Swiss Meteorological Workstation Project*) foi desenvolvido no Instituto Meteorológico da Suíça, visando a substituição da “pesada” arquitetura do mainframe pela moderna arquitetura cliente-servidor [16]. Seu principal diferencial, quando comparado a outros softwares de visualização climática, está na integração que apresenta ao banco de dados NEONS (Naval Environmental Operational Nowcasting System), o qual é voltado ao gerenciamento de grandes volumes de dados meteorológicos.

Esta integração foi a base para a implementação modular do MetAP, que mapeia os tipos de dados genéricos do NEONS, como pontos, imagens, grades e volume, em objetos gráficos. Para cada um destes, podem ser desenvolvidos módulos para recuperação, transformação, visualização e edição, independentes da interface de usuário. A visualização dos dados pode ser realizada de cinco maneiras diferentes: gráficos 3D (voxels) e 2D (vetores), linhas, símbolos e texto (Fig. 4.4).

A ergonomia constitui outro aspecto importante no contexto do MetAP, uma vez que esta serve de parâmetro para aliar o software ao local de trabalho, proporcionando uma interação homem-máquina mais eficiente e contribuindo, dessa forma, para um melhor desempenho das tarefas dos previsores.

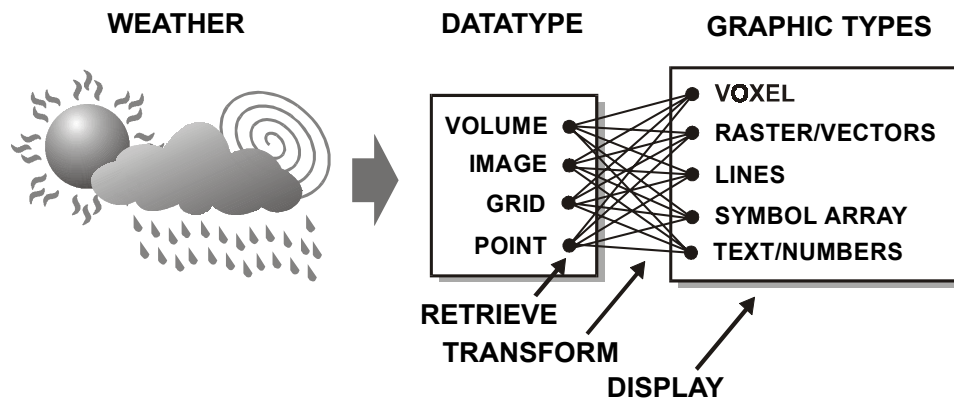


Fig. 4.4: Tipos de dados no banco de dados meteorológico e sua transformação em tipos gráficos no software MetAP, segundo Pauli et al. [16]

• Interface

O MetAP apresenta uma interface própria, desenvolvida especialmente para fins de visualização e produção de previsões. Esta é constituída por três janelas principais, cada uma desempenhando uma função específica e obedecendo a uma hierarquia de trabalho, de acordo com as etapas de previsão do tempo. São elas: *janela de seleção principal*, *janela de seleção e apresentação de parâmetros* e *janela de visualização e edição* (Fig. 4.5).

É importante salientar que o MetAP implementa um sistema de preparação da previsão baseado na *abordagem funil* [6], ou seja, as suas janelas conduzem a um único objetivo que é a previsão do tempo, ao passo que, em um sistema voltado para a pesquisa, tem-se uma gama de opções para a exploração dos dados.

A primeira janela apresentada ao usuário é a de seleção principal, responsável pela escolha da área de interesse temático. Esta está subdividida em três partes principais: *funções*, onde aparecem as opções como “Fim”, “Mudança de Usuário”, “Informação de Estado”, “Alarmes” e “Ajuda”; *Aplicações*, que ativa módulos de tratamento de dados (modelos numéricos, imagens de satélites, radar, etc.), através da passagem de parâmetros; e, por último, a parte de *Comandos*, que, pela linha de comando, possibilita a ativação de macros, as quais contêm o pré-cálculo das visualizações, via módulo de agendamento, e a integração à interface com o usuário.

Já a janela de seleção de parâmetros para a visualização está subdividida em quatro seções principais, que determinam o ambiente dimensional da visualização: *o quê*, para seleccionar o parâmetro físico que se quer observar (temperatura, pressão, etc.); *onde*, selecciona a área de onde virão os dados ou de onde estes serão vistos (global, continente, etc.); *quando*, define o período (data/hora) em que os dados serão analisados; *como*, responsável pela seleção do formato de exibição dos dados (valores, símbolos, linhas de contorno, etc.).

Por fim, na janela de visualização e edição, são apresentados e manipulados o conjunto de dados ou produto seleccionado, através de comandos direccionados a tais tarefas.

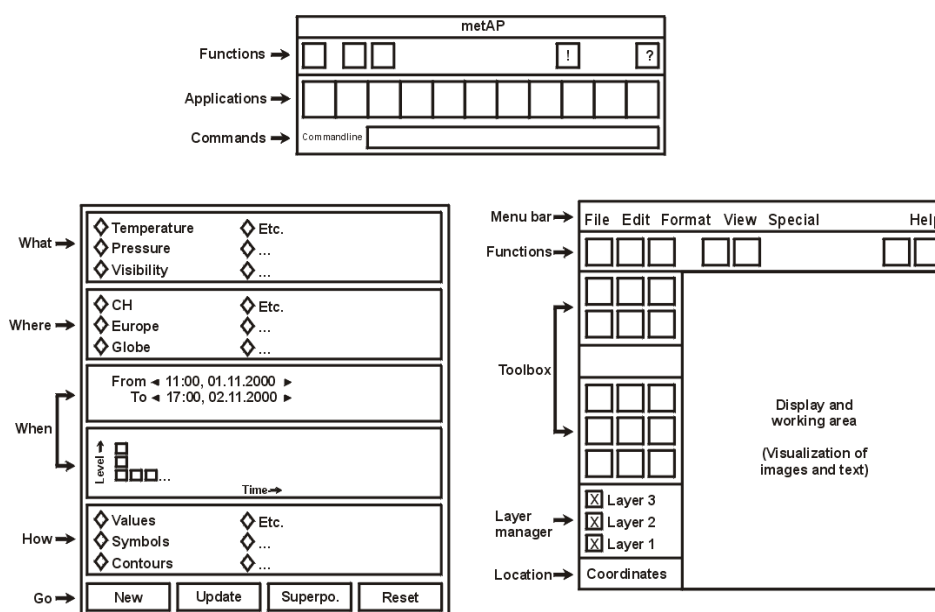


Fig. 4.5: Visão esquemática das janelas do MetAP: na parte superior, a janela principal, à esquerda, a janela de parâmetros, e, à direita, a janela de visualização [18].

Capítulo 5

Sistema de Visualização de Dados para o CPMet

5.1 Implementação

Além dos conceitos estudados e apresentados nos primeiros capítulos do presente trabalho, para que a construção deste sistema pudesse ter sido realizada com êxito e indo ao encontro das expectativas sobre ele depositadas, foi necessário um levantamento das atividades diárias realizadas pelos previsores e de suas principais expectativas quanto ao novo sistema.

Foram levantados dados a respeito da proveniência dos dados, das máquinas disponíveis para a implantação do software e para o armazenamento dos dados meteorológicos e do sistema operacional disponível. De posse destas informações, a abrangência relativa ao software em questão pôde ser definida e, a partir de então, sua implementação realizada.

A utilidade do sistema de visualização desenvolvido está em fornecer uma interface de controle e manipulação de dados climáticos e na síntese (ou simulação visual) destes, realizada a partir da data corrente ou de uma determinada data anterior a esta. Para tanto, o usuário determina os parâmetros que compõem a data (dia, mês, ano), um horário específico e um certo nível de pressão atmosférica. Objetiva, com isso, servir de ferramenta para uso, por parte dos meteorologistas, em pesquisas que virão a ser desenvolvidas no CPMet.

Nesta seção, faz-se uma descrição mais aprofundada do ambiente de trabalho em termos de software e hardware, além da apresentação de uma analogia entre o estudo realizado e a implementação do sistema de visualização proposto. É discutida a sua adequação ao escopo das rotinas de trabalhado CPMet/UFPel.

5.1.1 Ambiente de Trabalho: Hardware

O sistema foi desenvolvido com o uso de um PC K6-II 450, em ambiente Linux.

Ele deverá ser utilizado em máquinas, similares àquela em que foi desenvolvido em termos de hardware, e que estão ligadas em rede, compartilhando os dados meteorológicos.

Estas máquinas localizam-se no *Laboratório de Meteorologia Sinótica (LMS)*, situado no CPMet, construído especialmente para fins de pesquisa por parte dos meteorologistas e ainda em estado de aprimoramento.

5.1.2 Ambiente de Trabalho: Software

A implementação do sistema descrito neste trabalho foi realizada em ambiente Linux Slackware versão 7.0, mas o mesmo será utilizado em máquinas que rodam o sistema operacional FreeBSD versão 4.2. Em relação ao software de desenvolvimento, como já mencionado anteriormente, foi utilizado o software de visualização GrADS versão 1.7 beta9, o qual foi apresentado, em termos da importância que possui, na área da Visualização Científica voltada para a Meteorologia, na seção 4.2.2. No CPMet já

existe uma versão do GrADS mais atualizada rodando nas máquinas do LMS, a versão 1.7 beta 14.

5.1.3 Armazenamento dos Dados

A aquisição do arquivo que será utilizado para a visualização, no formato GRIB, contendo as variáveis e seus atributos, é feita acessando os dados do NCEP/NOAA, via Internet, por meio de processos agendados que fazem um pré-processamento desses dados, deixando-os prontos para a visualização através da geração de arquivos descritores que mapeiam os dados do arquivo binário.

Com os dados já devidamente compactados, estes são armazenados em disco, em uma hierarquia de diretórios organizada por ano, mês e dia, de modo a facilitar o posterior acesso para fins de pesquisas, às quais se destina este projeto. O armazenamento anterior a este sistema foi possível devido a um procedimento rotineiro de *backup* em fita magnética, o qual possibilitou a preservação de informações a partir do ano de 1994.

O novo sistema apresenta portabilidade quanto à origem das informações. Considerando-se que, futuramente, os dados possam vir a ser migrados para um sistema de consultas que utilize SQL, é viável a hipótese de adaptação do sistema de visualização.

5.1.4 Sistema Voltado à Pesquisa

Durante o acesso aos dados, por parte do meteorologista, será oferecida a possibilidade do mesmo indicar uma data arbitrária, na qual deseja efetivar a consulta. A data limite para acesso está definida de acordo com a capacidade do disco das máquinas responsáveis pelo armazenamento dos dados.

Com a possibilidade da recuperação de informações anteriores à data corrente, a ferramenta deixa de assumir um caráter eminentemente operacional e passa a permitir sua utilização voltada à pesquisa. Este constitui seu principal diferencial quando comparada as demais ferramentas de visualização implantadas atualmente no CPMet.

Uma das principais vantagens obtidas ao permitir a realização de pesquisas que envolvam dados anteriormente analisados, em um dado período de tempo estipulado pelo próprio operador, está na possibilidade da comparação da estimativa já efetuada, a partir destes dados, com as informações realmente ocorridas.

Através do cruzamento das informações estimadas com as ocorridas, passa a existir a possibilidade de traçar um perfil de acerto dos prognósticos meteorológicos. A partir do levantamento dos acertos na previsão, os técnicos envolvidos poderão reavaliar seus métodos de trabalho, localizando os principais problemas existentes.

5.1.5 Variáveis Principais

Os dados armazenados são compostos por 15 variáveis e seus principais atributos necessários à síntese da imagem. Estas variáveis são descritas a seguir:

- 1) *precipitação acumulada*, que diz respeito à estimativa da quantidade de chuva medida (em mm) em um determinado período de tempo – no caso do CPMet, de 12 em 12 horas;

- 2) *pressão reduzida*, relativa ao “nivelamento” da pressão atmosférica (em uma determinada localidade) ao nível do mar, de modo a alcançar a uniformização destes níveis;
- 3) *altura geopotencial*, que representa a medida da altura (em m) entre dois níveis de pressão atmosférica;
- 4) *temperatura*, ou seja, a temperatura do ar em um determinado nível de pressão atmosférica pré-estabelecido;
- 5) *umidade relativa*, que representa o percentual de partículas de água que uma determinada massa de ar pode conter, em um determinado nível de pressão;
- 6) *vento*, que corresponde ao deslocamento do ar, das áreas mais frias em direção às mais quentes.
- 7) *divergência*, relativa à quantidade de umidade (por unidade de área) que uma determinada massa de ar está “ganhando” ou “perdendo”;
- 8) *linhas de corrente*, que refere-se à direção do fluxo (canalização) de deslocamento de determinada massa de ar;
- 9) *advecção de vorticidade*, ou seja, a quantidade de vorticidade do vento em uma unidade de tempo específica;
- 10) *água precipitável*, que é a medida (em mm) da quantidade de água, no estado líquido, que uma coluna atmosférica contém sobre determinada camada de ar;
- 11) *advecção de temperatura*, relativa à quantidade de ganho de temperatura em uma unidade de tempo específica;
- 12) *espessura e pressão*, medição de uma determinada espessura – que é a altura (em m) de uma determinada camada de ar entre duas alturas geopotenciais – a uma pressão atmosférica específica.
- 13) *tendência 12h pressão*, diz respeito à estimativa da quantidade de variação da pressão por unidade de tempo (neste caso, de 12 em 12 horas);
- 14) *depressão de temperatura*, corresponde à medida de quanto determinado nível de pressão terá em relação à oscilação de temperatura;
- 15) *vorticidade relativa*, relacionada à quantidade de vorticidade do vento por unidade de área (com relação à “vizinhança”).

5.2 Apresentação

A importância da adequação, objetividade e funcionalidade na construção da interface com o usuário já foi discutida no capítulo 3. Será traçado, nesta seção, um paralelo entre a aplicação destes conceitos já explanados e o desenvolvimento da interface do novo sistema de visualização proposto.

5.2.1 Adequação do Sistema ao Conceito MVC

De acordo com as especificações e restrições presentes na definição do sistema proposto, decidiu-se, por conveniência, tomar como base a arquitetura MVC (ver seção 3.3), o que constituiu o ponto de partida para a implementação do projeto em questão.

Considerando que o conceito MVC adota a idéia da estratificação do sistema em três componentes principais – o modelo, a vista e o controlador –, e partindo de uma avaliação do escopo no qual o projeto está inserido, decidiu-se pela não adoção do paradigma MVC em sua plenitude, mas sim incorporar a este projeto o mesmo princípio de separação dos aspectos relativos a um modelo de objetos – aspectos relativos ao

domínio do problema e aspectos relativos ao domínio computacional –, visando o aumento de flexibilidade e reusabilidade do modelo de objetos.

Pela flexibilidade que possui a linguagem GrADS, foi possível implementar os componentes deste sistema em um único bloco, definindo-se a abrangência relativa a cada um, de acordo com os aspectos apresentados. Estes componentes são descritos a seguir.

• O papel de cada componente

Desempenhando a função do *modelo*, estão os dados provenientes do NCEP e compactados no formato GRIB – o *modelo pgcf*, que têm seu armazenamento organizado em diretórios em disco, hierarquizados em *dia*, *mês* e *ano*. Estes dados são acessados a partir do arquivo descritor de dados *pgcf.ctl* (ver anexo 2), que contém uma descrição completa dos dados, além do nome do arquivo que os contém, como já foi dito anteriormente.

A parte da vista é facilitada pelo uso do software de visualização GrADS (ver seção 4.2.2), propriamente desenvolvido para a síntese gráfica de dados meteorológicos. A plotagem dos dados é definida de acordo com a variação dos valores das dimensões. Neste trabalho, adotou-se a variação de duas dimensões, o que gera o desenho de uma fatia 2D do conjunto de dados (Fig. 5.10). A geração da imagem gráfica é possibilitada “instanciando-se” o arquivo *pgcf.ctl* ao código fonte do programa pelo comando `open pgcf.ctl`. O trecho do código mencionado corresponde à função apresentada na fig. 5.1 (descrita com maiores detalhes no anexo 1):

```
* FUNCAO P/ MOSTRAR A "VISTA" *****
function mostra()
flag7 = 1; zoom = 0; lat = '-80 10'; lon = '-120 0'
aaaa = _ano; mm = _mes; dd = _dia
while(flag7)
  b7.opt = 0; 'reinit'
  'open pgcf.ctl'; 'set t '_hora; 'q time'
  'set grid off'; 'set mpdset mres'; 'set map 1'
  'set lat 'lat; 'set lon 'lon; 'set lev '_pr
  (...)
  if(_vnt = 1)
    min = 20; int = 5; lv = '20 25 30 35 40 45 50 55 60'
    'd skip(u,2); skip(v,2); t';
    'set strsiz 0.15'
    'draw string VENTO - 'lev' hPa (m/s)'
  endif
  (...)
  'set mpdset brmap'; 'draw map'
  'draw xlab '_leg
  (...)
endwhile
return
* FIM FUNCAO P/ MOSTRAR A "VISTA" *****
```

Fig. 5.1: Trecho de código encarregado de mostrar a vista

Como *controlador*, tem-se a interface com o usuário. Esta interface é responsável pela execução das tarefas requeridas pelo usuário, através da manipulação dos dados, representados pelo modelo, por parte do controlador. Vale ressaltar que os *widgets* do GrADS não seguem o mesmo princípio de independência entre interface e aplicação, como no Xt, por exemplo (ver seção 3.5). A fig. 5.2 apresenta, parcialmente, o trecho do código fonte do programa responsável pela implementação da interface.


```

* INTERFACE INICIAL *****
(...)
while(flag)
  b.opt = 0; 'reinit'
  'set rgb 90 100 100 100'; 'set rgb 91 50 50 50'; 'set rgb 92 200 200 200'
  'set line 92'; 'set string 0 c 7'
  'set strsiz 0.40'; 'draw recf 0 0 8.5 11'
  'draw string 4.25 9.5 CPMET: Modelo PGCF'
  'set strsiz 0.20'
  'draw string 4.25 8.0 SELECAO DE ATRIBUTOS PARA VISUALIZACAO'
  'set strsiz 0.20'
  if(var = 1)
    botao(1,2.75,6.0,2.5,0.8,VARIAVEIS,1,15,11)
  endif
  if(var = 0)
    botao(1,2.75,6.0,2.5,0.8,VARIAVEIS,1,15,14)
  endif
  (...)
  botao(7,3.0,1.5,2.0,0.8,VISUALIZAR,1,15,0)
  botao(8,5.5,1.5,2.0,0.8,SAIR,1,15,0)
  'q pos'
  b.opt = subwrd(result,7)
  if(b.opt < 7)
    if(b.opt = 1)
      if(var = 0)
        ret = vars()
        if(_flagv = 0); var = 0; else; var = 1; endif
      else
        if(var = 1 & _flagv = 1)
          ret = vars()
          if(_flagv = 0); var = 0; else; var = 1; endif
        endif
      endif
    endif
    (...)
    if(b.opt = 7)
* Atualiza a vista
      ret = mostra()
    endif
  endif
  if(b.opt = 8); flag = 0; endif
  'clear'
endwhile
'clear'; 'quit'
* FIM INTERFACE INICIAL *****
(...)
* FUNCAO ESCOLHA DE VARIAVEIS *****
(...)
* FUNCAO ESCOLHA DO DIA *****
(...)
* FUNCAO ESCOLHA DO HORARIO *****
(...)
* FUNCAO ESCOLHA DO MES *****
(...)
* FUNCAO ESCOLHA DA PRESSAO *****
(...)
* FUNCAO ESCOLHA DO ANO *****
(...)
* FUNCAO P/ MOSTRAR A "VISTA" *****
(...)
* FUNCAO P/ DESENHAR BOTAO WIDGET *****

```

Fig. 5.2: Trecho de código responsável pela implementação das interface

5.2.2 Importância do *Design*

Partindo-se da idéia de Hobart [10], a respeito da adequação de interfaces, na qual este defende que “*uma aplicação deve refletir as perspectivas e o comportamentos dos seus usuários*”, diversos aspectos relativos às principais expectativas depositadas pelos meteorologistas no contato e na manipulação de uma interface foram considerados.

Adotando como ponto principal a importância do “humano” nas interfaces, o processo de design se voltou para a preocupação com o entendimento do usuário de um sistema de visualização de dados meteorológicos, bem como suas atitudes, suas tarefas e sua percepção entre outras características.

Para tanto, foi necessário o conhecimento do usuário final (o meteorologista) e suas principais necessidades inerentes a um sistema deste porte. A identificação dos requisitos necessários ao sistema, a observação das tarefas e a sequência de ações executadas no CPMet contribuíram para que a adequação do sistema em questão pudesse ser alcançada e atendesse às especificações e objetivos anteriormente especificados.

5.2.3 Descrição das Janelas que Compõem a Interface

Considerando que o principal objetivo deste sistema de visualização se concentra em possibilitar ao meteorologista a realização de pesquisas relativas à previsão do tempo, a interface do software foi implementada seguindo aspectos ergonômicos, de modo a prover autonomia aos seus usuários, permitindo que tenham liberdade ao indicar a ou as variáveis a serem pesquisadas e seus principais atributos, num dado período de tempo. Para tanto, preferiu-se a não adoção da *abordagem funil* (ver seção 4.2.3), que seria usada em um sistema específico para elaboração de previsões.

A interface do projeto (implementada a partir das primitivas gráficas do GrADS, ver seção 4.2.2) é constituída por uma janela principal (Fig. 5.3), inicialmente apresentada ao usuário quando este invoca o sistema. Esta janela está subdividida em seis partes fundamentais à síntese da imagem e representadas, cada qual, por seu respectivo botão, responsável pela apresentação e seleção dos parâmetros da visualização.

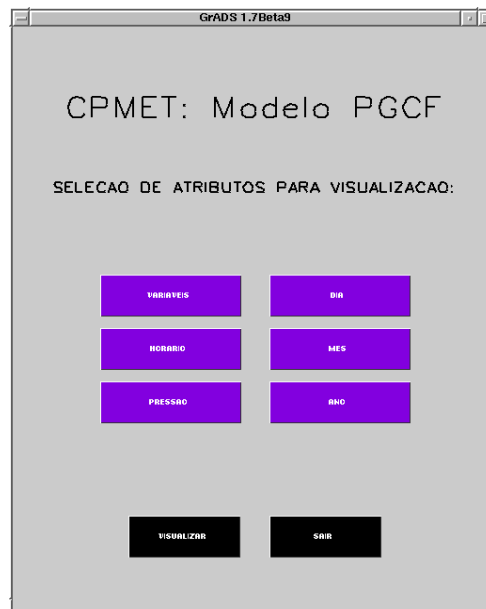


Fig 5.3: Janela principal.

Para facilitar a interatividade entre usuário e sistema, ao pressionar-se com o *mouse* em um dado botão este troca de cor, indicando que a área de interesse representada pelo mesmo já teve seus parâmetros selecionados. A concepção da mudança do aspecto inicial do botão visa um maior controle do software por parte do

operador, provendo-o do conhecimento da condição corrente relativa à pesquisa que realiza. Quando o meteorologista desejar alterar alguma seleção já realizada de parâmetros, antes do processo de visualização, ou mesmo o cancelamento de um certo comando, bastará pressionar novamente sobre o botão já selecionado.

Por exemplo, se já foi confirmada a seleção de parâmetros relativos às variáveis que comporão a visualização e deseja-se efetivar o cancelamento da escolha, o usuário deverá efetuar um clique no botão *variáveis* para que a janela de escolha destas seja novamente aberta, ainda com a presente seleção de parâmetros. Após ser realizado o cancelamento de determinado botão este voltará a sua apresentação inicial e anulará a escolha prévia dos parâmetros.

• Botão “VARIÁVEIS”

Este primeiro botão refere-se à escolha de uma ou mais variáveis que comporão o gráfico final (Fig. 5.4). Ao pressionar neste botão, uma nova janela, contendo quinze botões, representando cada um uma variável, é apresentada. O meteorologista realiza a seleção e confirma sua escolha com o botão “CONFIRMAR”. As variáveis que fazem parte desta janela são as já mencionadas na seção 5.1.5.

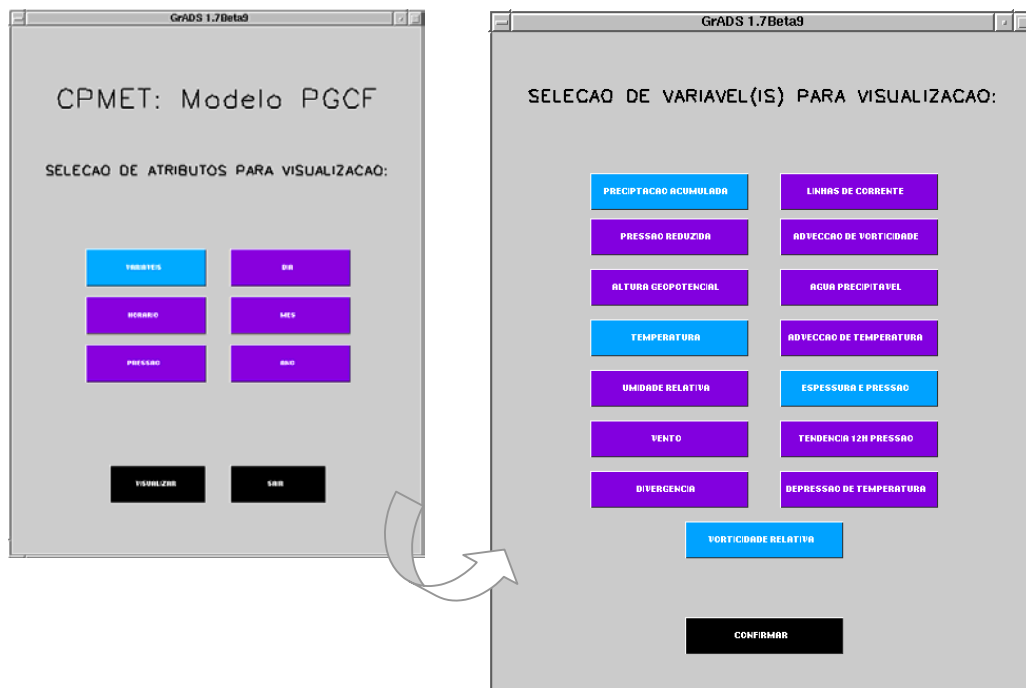


Fig. 5.4 : Janela de seleção das variáveis.

• Botão “HORÁRIO”

Para a visualização de uma determinada variável ser possível é indispensável a seleção do horário em que a mesma terá sua imagem sintetizada. Este horário representa o tempo, a partir da data e hora inicial, para o qual é calculado o prognóstico. O usuário terá que optar entre onze possibilidades de horários, com intervalo de 12 horas entre eles. Os horários disponíveis são: 00h, 12h, 24h, 36h, 48h, 60h, 72h, 84h, 96h, 108h e 120h. A figura a seguir (Fig. 5.5) faz uma demonstração desta escolha.

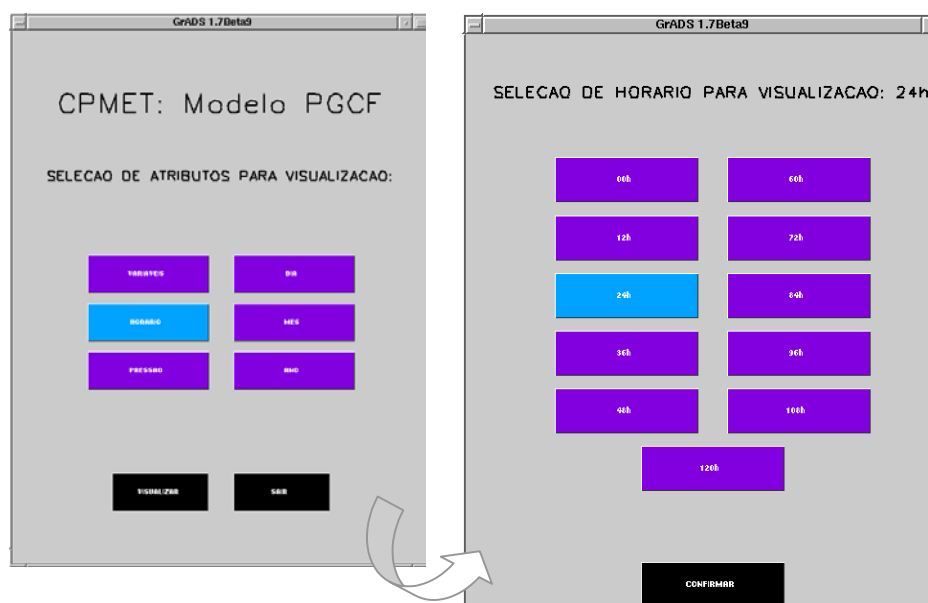


Fig. 5.5: Janela de seleção do horário.

- **Botão “PRESSÃO”**

O botão “PRESSÃO”, como o próprio rótulo já indica, permite a seleção do nível de pressão no qual as variáveis serão visualizadas (Fig. 5.6). São apresentadas onze possibilidades de escolha da pressão: *superfície*, 150 hPa, 200 hPa, 250 hPa, 300 hPa, 400 hPa, 500 hPa, 700 hPa, 850 hPa, 925 hPa e 1000 hPa.

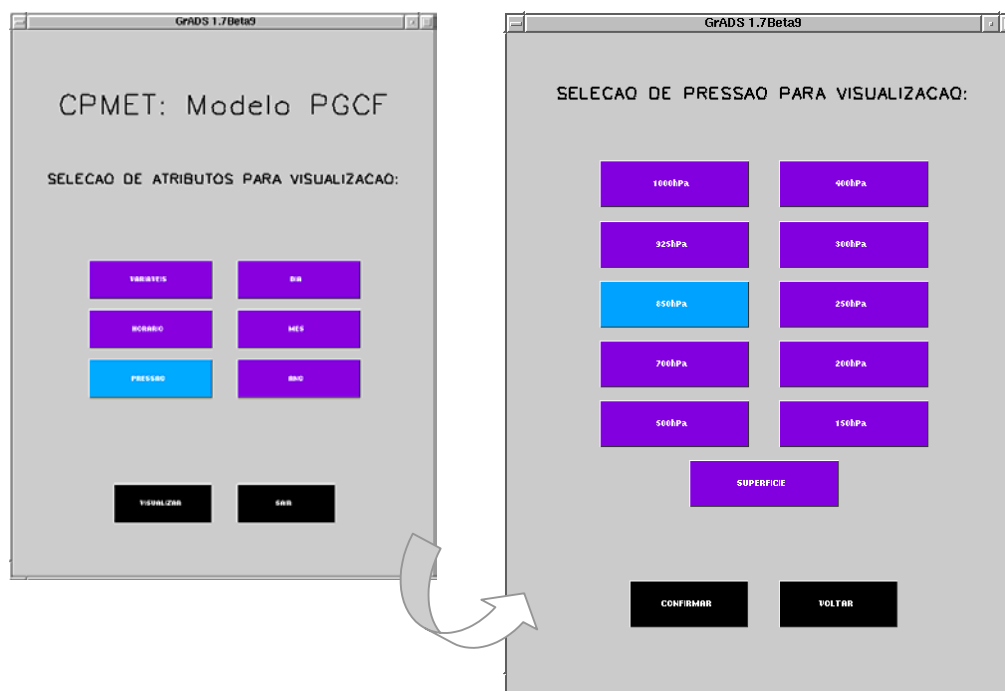


Fig. 5.6: Janela de seleção da pressão.

- **Botão "DIA"**

Na janela ativada por este botão, são apresentados outros dez botões, cada qual representando um dígito de "0" a "9" (Fig. 5.7). É permitido ao usuário "compor" o dia em será efetuada a visualização das variáveis.

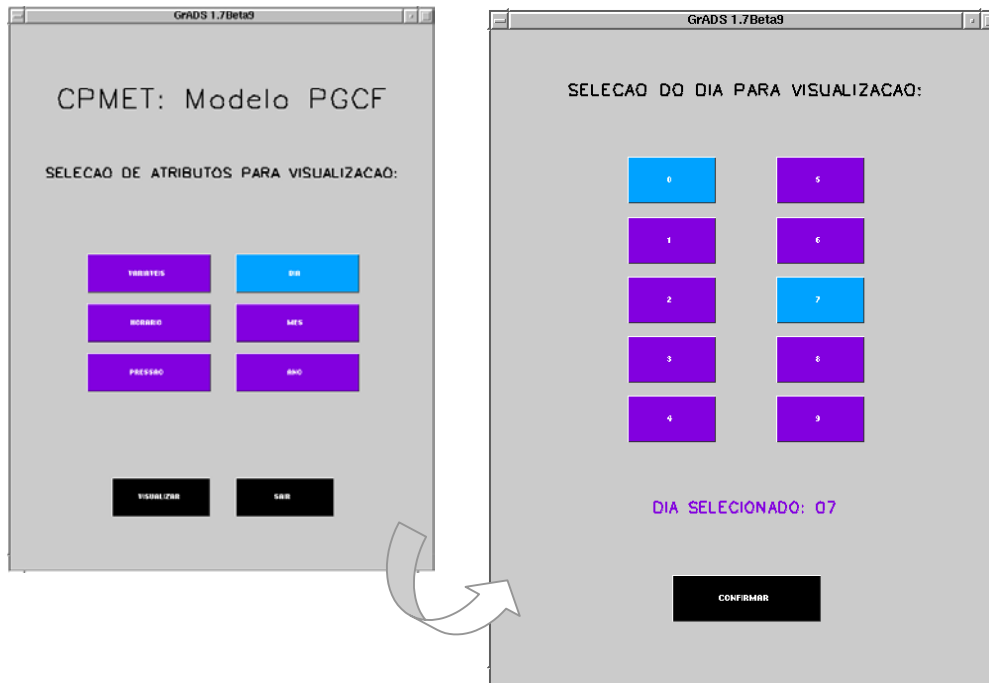


Fig. 5.7: Janela de composição do dia.

- **Botão "MÊS"**

Este botão ativa a janela encarregada de apresentar as opções dos meses do ano, escolhidos através do clique no respectivo botão (Fig. 5.8).



Fig. 5.8: Janela de composição do mês do ano.

- **Botão “ANO”**

Através do clique no botão “ANO”, uma nova janela é aberta, similar à janela da seleção do mês, a qual contém 8 botões – 1994, 1995, 1996, 1997, 1998, 1999, 2000 e 2001 –, responsáveis pela seleção do ano (Fig. 5.9). A data de 1994, data limite para a recuperação dos dados, foi estipulada a partir dos *backups* existentes no CPMet.

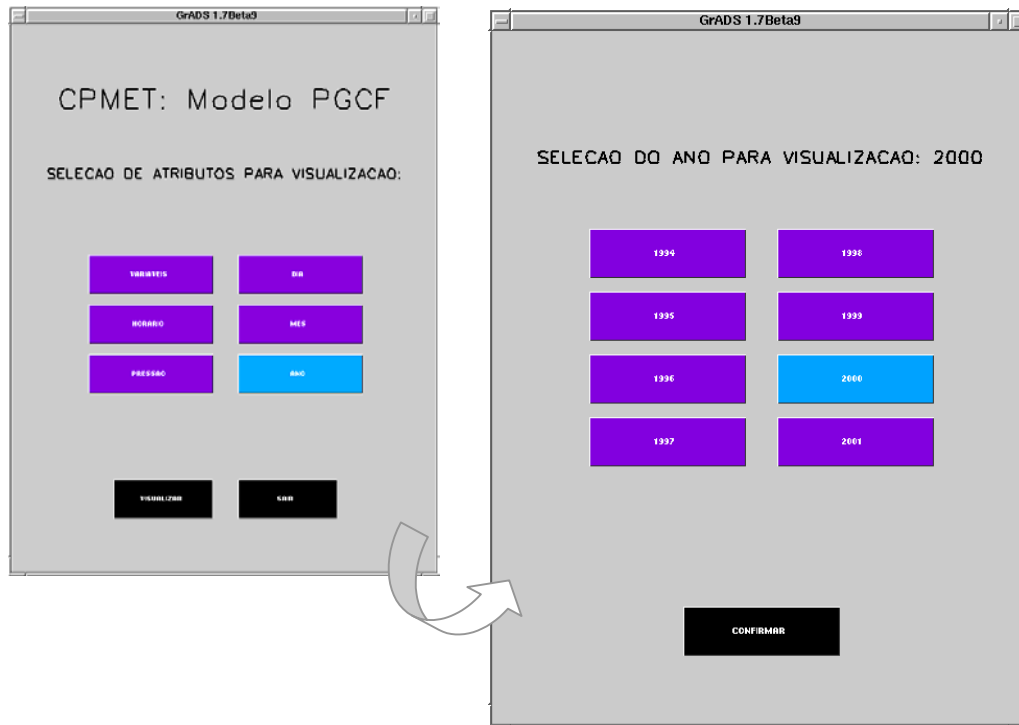


Fig. 5.9: Janela de composição do ano.

Esta forma de seleção de parâmetros apresentadas pelos botões *dia*, *mês* e *ano* foi adotada porque o GrADS não possui um *widget* para a entrada de texto pelo usuário.

- **Botão “VISUALIZAR”**

Assim que os botões relativos às variáveis, bem como os atributos necessários à visualização destas, estiverem selecionados, o usuário deverá pressionar no botão “VISUALIZAR”, encarregado da realização do tratamento dos parâmetros e conversão destes a restrições que definirão o escopo da sintetização da imagem. A partir de então, uma nova janela destinada à apresentação do gráfico gerado será aberta e, no caso da visualização de mais de uma variável, as imagens serão sobrepostas de modo a prover o usuário da possibilidade do cruzamento de informações das mesmas. (Fig. 5.10). Nesta janela, ainda existem dois botões: o “zoom”, que num primeiro passo apresenta a visualização do continente Sul-Americano como um todo e, quando ativado (Fig. 5.11), restringe a visualização somente à área relativa à região Sul do Brasil (Paraná, Santa Catarina e Rio Grande do Sul); e o “retorna”, responsável pelo encerramento da seção de visualização e retorno para a janela principal.

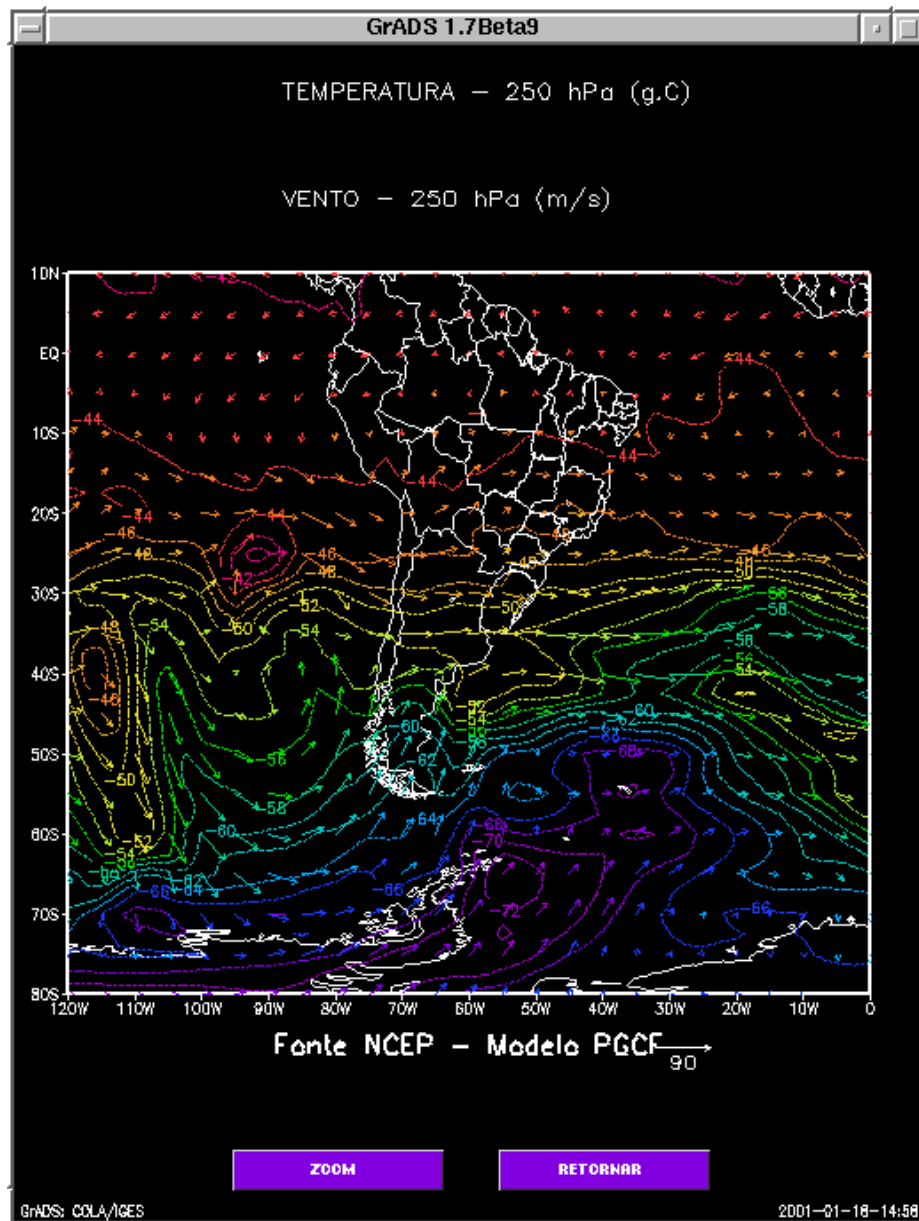


Fig. 5.10: Janela para a visualização das variáveis *temperatura* e *vento*.

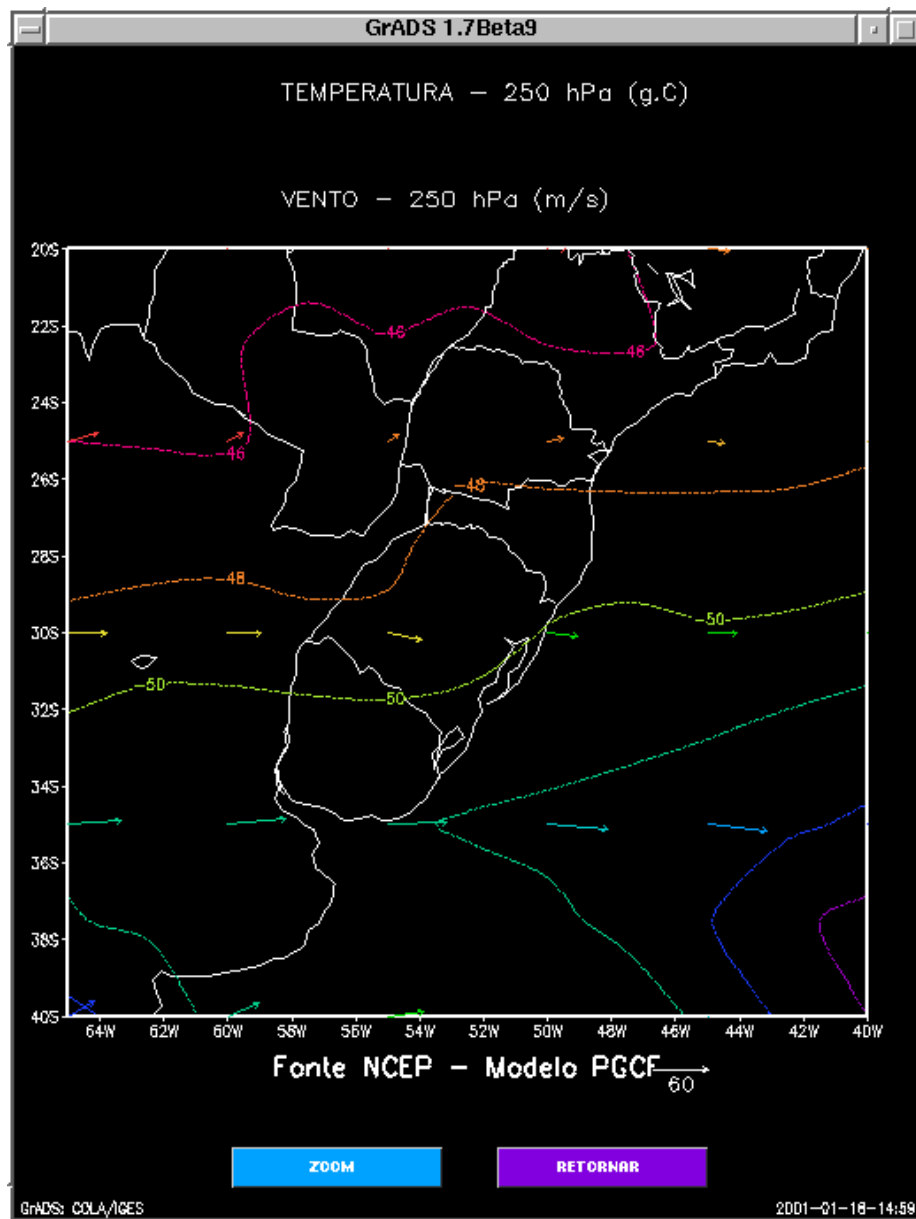


Fig. 5.11: Janela para a visualização das variáveis *temperatura* e *vento* com o botão *zoom* ativado.

- **Botão "SAIR"**

Este botão, quando ativado, encerra a execução do programa.

Capítulo 6

Conclusões

6.1 Resultados Obtidos

O presente trabalho permitiu, sob um ponto de vista de formação acadêmica, a assimilação de inúmeros conceitos não abordados durante o decorrer do curso de Informática.

A respeito do desenvolvimento e da adequação de interfaces com o usuário, estudou-se vários temas: as expectativas dos usuários relativas a uma nova interface, o porquê das principais dificuldades enfrentadas pelos desenvolvedores em adequá-la à necessidade do usuário, as principais ferramentas utilizadas na implementação de uma interface; e, a partir deste levantamento, se pôde definir uma interface que viesse ao encontro das necessidades dos meteorologistas e de modo a facilitar o trabalho de pesquisa destes.

Com relação à VC, o estudo feito permitiu traçar um perfil das suas principais aplicações em diversas áreas científicas e sua importância ao possibilitar a visualização de dados resultantes da computação científica. No caso específico deste trabalho, a visualização dos dados meteorológicos para fins de pesquisa visa possibilitar ao meteorologista do CPMet a realização de pesquisas de forma mais abrangente, através da indicação de uma data específica para a realização da sua análise e possibilitando a este a visualização de um grande volume de dados.

Já em relação ao paradigma MVC, um novo conceito envolvendo de projeto de software pôde ser compreendido. Com o entendimento dos domínios de cada componente do sistema, uma maior flexibilidade no uso do modelo de objetos foi possibilitada. A maneira como é definida a divisão de tarefas entre três componentes principais (model-view-controller) foi adaptada à realidade do sistema implementado. E, pelo fato do sistema ter sido implementado em GrADS, software já em uso no CPMet, a familiaridade deste com os meteorologistas foi aproveitada.

Uma vez que o GrADS é uma linguagem de *scripts*, a grande vantagem na sua utilização está na maleabilidade de implementação que esta proporciona, permitindo ao desenvolvedor definir suas funções de maneira mais adequada e com maior versatilidade.

Já em relação ao ambiente de operação, um estudo mais detalhado sobre os fundamentos de um sistema meteorológico foi desenvolvido, o que possibilitou o entendimento das diversas atividades diárias realizadas pelo meteorologista ao longo da composição dos seus prognósticos do tempo.

Tendo em vista estes conceitos expostos anteriormente, a ferramenta para visualização pôde ser desenvolvida de modo a reuni-los e adaptá-los ao objetivo para o qual ela foi proposta.

6.2 Principais Dificuldades Encontradas

Um dos principais fatores que dificultou a realização deste projeto foi a carência, apresentada no curso de Informática, de disciplinas voltadas à área da Computação Gráfica. Por esta razão, encontrou-se grande dificuldade no levantamento de material didático, como livros científicos, artigos de discussão, ferramentas voltadas à VC, etc.

Além disto, o tempo disponível entre a definição do projeto, aprovação e implementação constituiu outro fator negativo por ser relativamente insuficiente, o que restringiu a abrangência do sistema.

6.3 Perspectivas para o Futuro

Pretende-se, futuramente, dar continuidade ao trabalho com vista ao seu aprimoramento e extensão, através da incorporação de novas funcionalidades, como, por exemplo, a inserção de um botão responsável pelo arquivamento da imagem gráfica gerada pelo software, de modo que esta possa ser utilizada em um outro momento para fins de animação e pesquisa.

Além deste, a complementação do conhecimento já adquirido, aliada à incorporação do estudo envolvendo novas linhas de pesquisa, constitui um incentivo para o desenvolvimento de novos trabalhos que envolvam a síntese de imagens.

Anexos

Este anexo contém, primeiramente, o código fonte do programa em questão apresentado de forma parcial, uma vez que as funções que o compõem partem do mesmo princípio de implementação e, por isso, julga-se conveniente a apresentação de apenas algumas funções principais.

Em seguida, o próximo anexo está constituído pelo arquivo de metadados *pgcf.ctl*, o qual contém uma descrição completa dos dados que serão simulados visualmente pelo GrADS, no presente sistema.

Anexo 1: Código Fonte Parcial do Sistema de Visualização

```
*****
* Sistema de Visualização para o CPMet/UFPel
* Desenvolvimento: Isabel C. Siqueira da Silva
* Orientação: Carlos A. M. Santos
* Trabalho de Conclusão do Curso de Informática/UFPel
* Período de Desenvolvimento: de 07 - 11/2000
*****

* Definicao de variaveis globais que não sofrem alteracao de valores
* durante a execucao do programa
_hs = '00 12 24 36 48 60 72 84 96 108 120'
_pr = '1000 925 850 700 500 400 300 250 200 150 SUPERFICIE'
_dig = '0 1 2 3 4 5 6 7 8 9'
_m = 'JAN FEV MAR ABR MAI JUN JUL AGO SET OUT NOV DEZ'
_a = '1994 1995 1996 1997 1998 1999 2000 2001'
_leg = 'Fonte NCEP - Modelo PGCF'

* INTERFACE INICIAL *****
flag = 1; var = 0; hs = 0; pr = 0; _flagv = 0; _flagd = 0; _flagh = 0
dia = 0; mes = 0; ano = 0; _flagm = 0; _flagp = 0; _flaga = 0
_preac = 0; _pred = 0; _hgeo = 0; _temp = 0; _umre = 0
_vnt = 0; _div = 0; _vore = 0; _licor = 0; _advor = 0
_agpre = 0; _advt = 0; _espr = 0; _tend = 0; _depr = 0
_hora = 0; _pressao = 0; _dia = 'x'; _mes = 0; _ano = 0
while(flag)
    b.opt = 0; 'reinit'

* Definicao de cores e dimensoes apresentadas pela interface
'set rgb 90 100 100 100'; 'set rgb 91 50 50 50'; 'set rgb 92 200 200 200'
'set line 92'; 'set string 0 c 7'
'set strsiz 0.40'; 'draw recf 0 0 8.5 11'
'draw string 4.25 9.5 CPMET: Modelo PGCF'
'set strsiz 0.20'
'draw string 4.25 8.0 SELECAO DE ATRIBUTOS PARA VISUALIZACAO'
'set strsiz 0.20'

* Definicao dos botoes encarregados de invocar as funcoes responsaveis
* pela selecao de parametros necessarios a visualizacao
    if(var = 1)
        botao(1,2.75,6.0,2.5,0.8,VARIAVEIS,1,15,11)
    endif
    if(var = 0)
        botao(1,2.75,6.0,2.5,0.8,VARIAVEIS,1,15,14)
    endif

    (...)

* Botao encarregado de implementar a simulacao grafica dos dados
* (somente acionado apos a selecao de todos os parametros)
    botao(7,3.0,1.5,2.0,0.8,VISUALIZAR,1,15,0)

* Botao responsavel pelo termino da execucao do programa
    botao(8,5.5,1.5,2.0,0.8,SAIR,1,15,0)
    'q pos'
    b.opt = subwrd(result,7)
    if(b.opt < 7)
        if(b.opt = 1)
            if(var = 0)
                ret = vars()
                if(_flagv = 0); var = 0; else; var = 1; endif
            else
                if(var = 1 & _flagv = 1)
                    ret = vars()
                    if(_flagv = 0); var = 0; else; var = 1; endif
                endif
            endif
        endif
    endif

    (...)
```

```

        if(b.opt = 7)

* Atualiza a vista
        ret = mostra()
        endif
    endif
    if(b.opt = 8); flag = 0; endif
    'clear'
endwhile
'clear'; 'quit'
* FIM INTERFACE INICIAL *****

* FUNCAO ESCOLHA DE VARIAVEIS *****
function vars()
aux1 = 0; aux2 = 0; aux3 = 0; aux4 = 0; aux5 = 0
aux6 = 0; aux7 = 0; aux8 = 0; aux9 = 0; aux10 = 0
aux11 = 0; aux12 = 0; aux13 = 0; aux14 = 0; aux15 = 0
flag1 = 1
while(flag1)
    bl.opt = 0; 'reinit'
    'set rgb 90 100 100 100'; 'set rgb 91 50 50 50'; 'set rgb 92 200 200 200'
    'set line 92'; 'set string 0 c 7'
    'set strsiz 0.20'; 'draw recf 0 0 8.5 11'
    'draw string 4.25 10.0 SELECAO DE VARIAVEL(IS) PARA VISUALIZACAO:'

    (...)

    if(aux1 = 1 | (_flagv = 1 & _temp = 1))
        botao(1,2.75,6.2,2.5,0.6,TEMPERATURA,1,15,11)
    endif
    if(aux1 = 0 & _temp = 0)
        botao(1,2.75,6.2,2.5,0.6,TEMPERATURA,1,15,14)
    endif
    if(aux2 = 1 | (_flagv = 1 & _vnt = 1))
        botao(2,2.75,4.6,2.5,0.6,VENTO,1,15,11)
    endif
    if(aux2 = 0 & _vnt = 0)
        botao(2,2.75,4.6,2.5,0.6,VENTO,1,15,14)
    endif

    (...)

    botao(16,4.25,1.5,2.5,0.6,CONFIRMAR,1,15,0)
    'q pos'
    bl.opt = subwrd(result,7)

* Trecho encarregado da escolha das variaveis que irao compor a imagem
* grafica gerada
    if(bl.opt < 16)
        if(bl.opt = 1)
            if(aux1 = 0)
                aux1 = 1; _temp = 1; _flagv = 1
            else
                if(aux1 = 1 | (_flagv = 1 & _temp = 1))
                    aux1 = 0; _temp = 0
                endif
            endif
        endif
        if(bl.opt = 2)
            if(aux2 = 0)
                aux2 = 1; _vnt = 1; _flagv = 1
            else
                if(aux2 = 1 | (_flagv = 1 & _vnt = 1))
                    aux2 = 0; _vnt = 0
                endif
            endif
        endif
    endif

    (...)

endif
if(_preac = 0 & _pred = 0 & _hgeo = 0 & _temp = 0 & _umre = 0)
    if(_vnt = 0 & _div = 0 & _vore = 0 & _licor = 0 & _advor = 0)
        if(_agpre = 0 & _advr = 0 & _espr = 0 & _tend = 0 & _depr = 0)
            _flagv = 0
        endif
    endif
endif

```

```

        endif
        if(b1.opt = 16); flag1 = 0; endif
    endwhile
    return
* FIM FUNCAO ESCOLHA DE VARIABEIS *****

(...)

* FUNCAO ESCOLHA DO HORARIO *****
function horario()
flag2 = 1; aux = 0; ativo = 0; hs = 0
while(flag2)
    b2.opt = 0; 'reinit'
    'set rgb 90 100 100 100'; 'set rgb 91 50 50 50'; 'set rgb 92 200 200 200'
    'set line 92'; 'set string 0 c 7'
    'set strsiz 0.20'; 'draw recf 0 0 8.5 11'
    'draw string 4.25 10.0 SELECAO DE HORARIO PARA VISUALIZACAO: ' ft'h'
    if(_hora = 00 & _flagh = 1); cor1 = 11; else; cor1 = 14; endif
    if(_hora = 12); cor2 = 11; else; cor2 = 14; endif
    if(_hora = 24); cor3 = 11; else; cor3 = 14; endif

    (...)

    botao(1,2.75,8.5,2.5,0.8,00h,1,15,cor1)
    botao(2,2.75,7.5,2.5,0.8,12h,1,15,cor2)
    botao(3,2.75,6.5,2.5,0.8,24h,1,15,cor3)

    (...)

    botao(12,4.25,1.5,2.5,0.8,CONFIRMAR,1,15,0)
    'q pos'
    b2.opt = subwrd(result,7); hs = b2.opt
    if(b2.opt < 12)
        if(aux = 1)
            if(ativo = 0)
                if(b2.opt = 1)
                    _hora = subwrd(_hs,hs); cor1 = 11; ativo = 1
                    _flagh = 1; ft = _hora
                endif
                (...)

            endif
            aux = 0
        else
            if(aux = 0)
                if(b2.opt = 1)
                    if(cor1 = 11); ativo = 0; _flagh = 0; _hora = 0; ft = 0; endif
                    cor1 = 14
                endif
                (...)

                aux = 1
            endif
        endif
    endif
    if(b2.opt = 12); flag2 = 0; endif
endwhile
return
* FIM FUNCAO ESCOLHA DO HORARIO *****

(...)

* FUNCAO P/ MOSTRAR A "VISTA" *****
function mostra()
flag7 = 1; zoom = 0; lat = '-80 10'; lon = '-120 0'
aaaa = _ano; mm = _mes; dd = _dia
while(flag7)
    b7.opt = 0; 'reinit'

    * Instanciamento do arquivo descritor do arquivo GRIB de dados
    'open pgcf.ctl'

    * Selecao dos atributos da imagem grafica gerada
    'set t _hora; 'q time'
    'set grid off'; 'set mpdset mres'; 'set map 1'

```

```

'set lat 'lat; 'set lon 'lon; 'set lev '_pr
* Verificacao das variaveis selecionadas
if(_temp = 1)
    if(_pr = SUPERFICIE)
        'set gxout contour'
        'set cint 2'
        'd ts-273'
        'set strsiz 0.15'
        'draw string TEMPERATURA DE SUPERFICIE (g.C)'
    else
        'set gxout contour'
        'set cint 2'
        'd t-273'
        'set strsiz 0.15'
        'draw string TEMPERATURA - 'lev' hPa (g.C)'
    endif
endif
if(_vnt = 1)
    min = 20; int = 5; lv = '20 25 30 35 40 45 50 55 60'
    'd skip(u,2); skip(v,2); t';
    'set strsiz 0.15'
    'draw string VENTO - 'lev' hPa (m/s)'
endif
(...)

'set mpdset brmap'; 'draw map'
'draw xlab '_leg

* Implementacao do botao "zoom"
if(zoom = 0)
    botao(1,3.0,0.5,2.0,0.4,ZOOM,1,15,14)
else
    if(zoom = 1)
        botao(1,3.0,0.5,2.0,0.4,ZOOM,1,15,11)
    endif
endif
botao(2,5.5,0.5,2.0,0.4,CANCELAR,1,15,14)
'q pos'; b7.opt = subwrđ(result,7)
if(b7.opt = 2); flag7 = 0; endif
if(b7.opt = 1)
    if(zoom = 1)
        lat = '-80 10'; lon = '-120 0'; zoom = 0
    else
        lat = '-40 -20'; lon = '-65 -40'; zoom = 1
    endif
endif
endif

* Este botao finaliza a visualizacao graficas dos dados e volta a tela
* inicial
botao(2,5.5,0.5,2.0,0.4,CANCELAR,1,15,14)
'q pos'; b7.opt = subwrđ(result,7)
endwhile
return
* FIM FUNCAO P/ MOSTRAR A "VISTA" *****

FUNCAO P/ DESENHAR BOTAO WIDGET *****
function botao(bnum,xc,yc,dx,dy,texto,oncor,offcor,facecor)
'set button 'oncor' 'facecor' 91 1 'offcor' 91 1 6'
'draw button 'bnum' 'xc' 'yc' 'dx' 'dy' 'texto'
return
* FIM FUNCAO BOTAO *****

```

Anexo 2: Conteúdo do Arquivo pgcf.ctl

```
* Localizacao dos arquivos de dados na hierarquia de diretorios
* (aaaa = ano; mm = mês; dd = dia)
dset /home/Banco/aaaa/mm/dd/pgcf%f2.aaaamdd
dtype grib
index pgcf.map
options template
title PROGNOSTICOS
undef -9.99E+33
xdef 144 linear 0.000 2.500
ydef 44 linear -90.000 2.500
zdef 11 levels 1000 925 850 700 500 400 300 250 200 150 100
tdef 11 linear ddmmaaaa 12hr
vars 13
gh 10 7,100,0 Geopotential height (gpm)
u 10 33,100,0 u wind (m/s)
v 10 34,100,0 v wind (m/s)
t 10 11,100,0 Temperature (K)
vpv 10 39,100,0 Pressure vertical velocity (Pa/s)
rh 11 52,100,0 Relative humidity (percent)
ap 0 54,200,0 Precipitable water (kg/m**2)
prm 0 2,102,0 Pressure reduced to MSL (Pa)
ts 0 11,1,0 Temperature (K)
fr 0 212,1,0 Upward long wave radiation flux (W/m**2)
prt 0 61,1,0 Total precipitation (kg/m**2)
sh 0 51,105,2 Specific humidity (kg/kg)
tcc 0 71,200,0 Total cloud cover (percent)
endvars
```


Glossário

Backup: Cópia de segurança.

Design: Termo estrangeiro relativo a projeto, na língua portuguesa.

Insight: Não há uma tradução direta para a língua portuguesa. Caracteriza a facilidade e rapidez na compreensão da natureza das situações ou problemas complexos (equivalente à *perspicácia*).

Look and feel: Termo utilizado na área de desenvolvimento de interfaces que significa “aparência e sensação”.

Mouse: Dispositivo de interação entre usuário e aplicação.

Renderizar: Ato de gerar imagens no computador.

Script: Programa submetido a um interpretador.

Toolkits: Bibliotecas para a construção de interfaces.

Traps: Interrupções de processos, geradas pelo sistema operacional.

Widgets: Elementos ativos de uma interface.

Referências Bibliográficas

- [1] Bryson, S.; Kenwright, D.; Cox, M.; Ellsworth, D.; Haimes, R. **Visually Exploring Gigabyte Data Sets in Real Time**. Communications of the ACM, v. 42, n. 8, p. 83-90, Aug. 1999.
- [2] Burbeck, Steve. **Applications Programming in Smalltalk-80™: How to use Model-View-Controller (MVC)**. Disponível via WWW em <http://www.cs.uiuc.edu/users/smarch/st-docs/mvc.html> (1992)
- [3] DeFanti, Thomas A.; Brown, M. D.; McCormick, B. H. **Visualization in Scientific Computing**. IEEE Computer Graphics & Applications, Los Alamitos: IEEE Computer Society Press, p. 61-70, Jul. 1987.
- [4] Dey, Clifford H. **GRIB (Edition 1)**. NOAA/NCEP, 1998. Disponível via ftp em: <ftp://ftp.ncep.noaa.gov/pub/ncmc/docs/gribed1/> (11/jan/1999)
- [5] Doty, Brian. **GrADS Home Page**. Disponível via WWW em <http://grads.iges.org/grads/> (15/jul/1998)
- [6] Grote, U. Herb; Bullock, Carl S. **User Interface Design of The WFO-Advanced Workstation**. In: International Conference on Interactive and Processing Systems for Meteorology, Oceanography and Hidrology, 13., 1997, Long Beach. **Proceedings**. Long Beach: American Meteorology Society, 1996. p. 190-193.
- [7] Günter, O. & Voisard, Agnès. *Metadata in geographical and Environmental Data Management*. In: Klas, W. & Sheth (eds.). **Managing Multimedia Data: Using Metadata to Integrate and Apply Digital Data**. McGraw-Hill. 1997.
- [8] Hankin, S.; Harrison, D.E.; Osborne, J.; Davison, J.; O'Brien, K. **A Strategy and a Tool, Ferret, for Closely Integrated Visualization and Analysis**. Disponível via ftp em: <http://ferret.wrc.noaa.gov/Ferret/>.
- [9] Hibbard, W. **Interactive Visualization and Steering for Weather Computations**. In: Visualization, 1990. San Francisco. **Proceedings**. Los Alamitos: IEEE Computer Society Press, 1990. p. 28-35.
- [10] Hobart, J. **Principles of Good GUI Design**. San Francisco: Unix Review, vol. 13, núm. 10. Sep, 1995. p. 37-46. San Francisco: Unix Review. Disponível via WWW em http://axp16.iie.org.mx/Monitor/v01n03/ar_ihc2.htm
- [11] Karhila, V. E. **Metview in a member state: a dream or a reality?**. In: Workshop on Meteorological Operational Systems, 5., 1995, Reading. **Proceedings**. Reading: ECMWF, 1995. p. 321. p. 172-174.

- [12] Matias, M.; Heemann, V; Santos, N. **Aspectos Cognitivos da Iteração Humano-Computador Multimídia.** In: Workshop sobre Fatores Humanos em Sistemas Computacionais, 3., 2000, Gramado. Anais. Porto Alegre: Instituto de Informática da UFRGS, 2000. p. 22-32.
- [13] Max, N.; Crawfis, R.; Williams, D. **Visualization for Climate Modeling.** IEEE Computer Graphics & Applications, Los Alamitos: IEEE Computer Society Press, p. 34-40, Jul. 1993.
- [14] Myers, Brad A. **Why are Human-Computer Interfaces Difficult to Design and Implement?**
- [15] Nye, Adrian; O'Reilly, Tim. **X Toolkit Intrinsics Programming Manual – Volume Four.** 2ª ed. Local? O'Reilly & Associates, Inc. Sep, 1990. p. 3-125.
- [16] Pauli, C.; Matter, D.; Ambrosetti, P. **Swiss Meteorological Worstation Project.** In: Workshop on Meteorological Operational Systems, 5., Reading. **Proccedings.** Reading: ECMWF, 1995. 321 p. p. 273-282.
- [17] Santos, Carlos A. M. **Recursos de Visualização Científica Necessários para um Sistema Meteorológico Operacional.** Porto Alegre, RS: Universidade Federal do Rio Grande do Sul, Curso de Pós-Graduação em Ciência da Computação, 1999.
- [18] Silva, Arlindo da. **Win32 GrADS Distribution at DAO.** Disponível via WWW em [`http://dao.gsfc.nasa.gov/software/grads/win32/lates/`](http://dao.gsfc.nasa.gov/software/grads/win32/lates/) (05/Mai/1998)
- [19] Sundsted, Todd. **Observer and Observable.** JavaWorld. Disponível via WWW em: [`http://www.javaworld.com/javaworld/jw-10-1996/jw-10-howto.html`](http://www.javaworld.com/javaworld/jw-10-1996/jw-10-howto.html) p(Oct/1996)
- [20] University Consortium for atmosferic research. **Unidata NetCDF.** Disponível via WWW em: [`http://www.unidata.ucar.edu/backages/netcdf.`](http://www.unidata.ucar.edu/backages/netcdf)

**Sistema de Visualização de Dados Meteorológicos
para o Centro de Pesquisas Meteorológicas da UFPel**

por

Isabel Cristina Siqueira da Silva

Monografia apresentada aos Senhores:

Analista de Sistemas, Carlos Augusto Moreira dos Santos
Orientador.

Prof. Maurício Nunes Porto, Espec.
Co-Orientador.

Meteorologista, Julio Renato Marques, MSc.

Vista e permitida a impressão.
Pelotas, ____/____/____.