

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIAS DA COMPUTAÇÃO**  
(Bacharelado)

**TUTORIAL PARA ORIENTAÇÃO E ESTUDOS SOBRE  
TÉCNICAS DE IMPLEMENTAÇÃO DE INTERFACES  
GRÁFICAS**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE  
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA  
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA  
COMPUTAÇÃO — BACHARELADO

**RENATO KERN GOMES**

BLUMENAU, DEZEMBRO/2002

2002/2-53

# TUTORIAL PARA ORIENTAÇÃO E ESTUDOS SOBRE TÉCNICAS DE IMPLEMENTAÇÃO DE INTERFACES GRÁFICAS

**RENATO KERN GOMES**

ESTE TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO  
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE  
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

**BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO**

---

Prof. Carlos Eduardo Negrão Bizzotto — Orientador na  
FURB

---

Prof. José Roque Voltolini da Silva — Coordenador do TCC

**BANCA EXAMINADORA**

---

Prof. Carlos Eduardo Negrão Bizzotto

---

Prof. Dalton Solando dos Reis

---

Prof. Marcel Hugo

# DEDICATÓRIA

Dedico este trabalho a toda minha família, entretanto, com um carinho especial à minha mãe, Anita Kern, sem a qual certamente não teria chegado a este ponto importante de minha vida.

Jamais esquecerei a sua bravura e esforço para permitir que conseguíssemos estudar em condições tão adversas as quais somente nós, seus filhos, podemos avaliar.

# AGRADECIMENTOS

Agradeço aos amigos que me apoiaram e deram suporte e dicas para a confecção do trabalho.

Excluído:

Especial agradecimento ao Prof. Dr. Carlos Eduardo Negrão Bizzoto, pela atenção e paciência na difícil tarefa de ensinar e orientar.

# SUMÁRIO

DEDICATÓRIA.....	III
AGRADECIMENTOS.....	IV
SUMÁRIO.....	V
LISTA DE FIGURAS.....	VII
RESUMO.....	IX
ABSTRACT.....	X
1 INTRODUÇÃO.....	1
1.1 OBJETIVOS DO TRABALHO.....	2
1.2 MOTIVAÇÃO E JUSTIFICATIVA.....	2
1.3 ESTRUTURA DO TRABALHO.....	3
2 FUNDAMENTOS DE INTERFACES GRÁFICAS.....	4
2.1 CONCEITOS.....	4
2.2 COMO PROJETAR UMA INTERFACE GUI.....	4
2.3 ELEMENTOS DE UMA INTERFACE GRÁFICA.....	8
2.3.1 WINDOWS (JANELAS).....	8
2.3.2 CAIXAS DE EDIÇÃO.....	9
2.3.3 DESENHOS X BITMAPS.....	9
2.3.4 MENUS.....	10
2.3.5 CAIXAS DE MENSAGEM E DE DIÁLOGOS.....	10
2.3.6 CONTROLES.....	11
2.3.7 BARRAS DE FERRAMENTAS.....	11
3 RECOMENDAÇÕES PARA O PROJETO DE INTERFACES GRÁFICAS.....	12
3.1 DISPOSIÇÃO DA INFORMAÇÃO.....	13

3.2 FORMA.....	15
3.3 CORES.....	17
3.4 TEXTO.....	20
3.5 SAÍDA DE INFORMAÇÕES.....	22
3.6 ILUSÃO DE ÓPTICA.....	24
4 METODOLOGIA OOHDM – OBJECT ORIENTED HYPERMEDIA DESIGN METHOD.....	26
4.1 PROJETO CONCEITUAL.....	27
4.2 PROJETO NAVEGACIONAL.....	28
4.3 PROJETO DA INTERFACE ABSTRATA.....	30
4.4 IMPLEMENTAÇÃO.....	33
5 ESPECIFICAÇÃO.....	34
6 IMPLEMENTAÇÃO.....	43
6.1 TÉCNICAS E FERRAMENTAS UTILIZADAS.....	43
6.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO.....	43
6.3 ANÁLISE E VALIDAÇÃO DO TUTORIAL.....	53
7 CONCLUSÕES.....	55
7.1 EXTENSÕES.....	56
REFERÊNCIAS BIBLIOGRÁFICAS.....	57

## LISTA DE FIGURAS

FIGURA 2.1 – ESQUEMA USUÁRIO-INTERFACE-APLICAÇÃO.....	04
FIGURA 2.2 – SÍMBOLO DA CRUZ VERMELHA – DIFERENÇA CULTURAL .....	05
FIGURA 2.3 – BOTÕES ANTIGOS DO WINDOWS .....	06
FIGURA 2.4 – BOTÕES ATUAIS DO WINDOWS .....	06
FIGURA 2.5 – UMA CAIXA DE DIÁLOGO INADEQUADA.....	07
FIGURA 2.6 – JANELA CONTROLES .....	11
FIGURA 2.7 – BARRA DE FERRAMENTAS DO MS-WINDOWS.....	11
FIGURA 3.1 – ESQUEMA DAS RECOMENDAÇÕES PARA O PROJETO DE INTERFA - CES GRÁFICAS .....	12
FIGURA 3.2 – PRINCÍPIOS DE ORDENAÇÃO DE LISTA .....	14
FIGURA 3.3 – CURSOR POSICIONADO NO 1º CAMPO NA ABERTURA DO PROGRA - MA .....	15
FIGURA 3.4 – ÍCONES DE ACORDO COM A FUNÇÃO .....	17
FIGURA 3.5 – TAMANHO E ESCALA DOS OBJETOS.....	17
FIGURA 3.6 – USO INADEQUADO DE COMBINAÇÃO DE CORES .....	19
FIGURA 3.7 – OPÇÕES DE COMANDO COM PALAVRA-CHAVE À ESQUERDA.....	21
FIGURA 3.8 – JANELA INFORMANDO AO USUÁRIO COMO RESOLVER O PROBLE - MA.....	23
FIGURA 3.9 – MOVIMENTO DE OBJETOS .....	23
FIGURA 3.10 – MESMO TAMANHO? .....	24
FIGURA 3.11 – DISTORÇÃO APARENTE .....	24
FIGURA 3.12 – AS RETAS HORIZONTAIS SÃO PARALELAS .....	24
FIGURA 3.13 – PONTOS PISCANTES .....	25
FIGURA 3.14 – QUADRADOS COM MESMO TOM DE CINZA.....	25
FIGURA 3.15 – O MESMO TOM DE CINZA .....	25
FIGURA 3.16 – SÓ DUAS CORES PRESENTES .....	25
FIGURA 4.1 – EXEMPLO DE PROJETO CONCEITUAL .....	28
FIGURA 4.2 – EXEMPLO DE PROJETO NAVEGACIONAL.....	30

Formatado

FIGURA 5.1 – PROJETO CONCEITUAL.....	35
FIGURA 5.2 – PROJETO NAVEGACIONAL .....	36
FIGURA 5.3 – DIAGRAMA HIERÁRQUICO FUNCIONAL 1.....	37
FIGURA 5.4 – DIAGRAMA HIERÁRQUICO FUNCIONAL 2.....	38
FIGURA 5.5 – FLUXOGRAMA PRINCIPAL .....	39
FIGURA 5.6 – FLUXOGRAMA DISPOSIÇÃO / FORMAS / CORES .....	40
FIGURA 5.7 – FLUXOGRAMA TEXTO / SAÍDAS / MENU .....	41
FIGURA 5.8 – PROJETO DA INTERFACE ABSTRATA .....	42
FIGURA 6.1 – TELA INICIAL .....	43
FIGURA 6.2 – DISPOSIÇÃO DA INFORMAÇÃO.....	44
FIGURA 6.3 – FORMA – AGRUPAMENTO .....	45
FIGURA 6.4 – ERRADO/CERTO.....	45
FIGURA 6.5 – ERRADO/CERTO – NORMAL .....	46
FIGURA 6.6 – ERRADO/CERTO – QUENTE.....	46
FIGURA 6.7 – TEXTO DESALINHADO .....	48
FIGURA 6.8 – ORIENTAÇÃO AO USUÁRIO.....	48
FIGURA 6.9 – LINKS .....	51
FIGURA 6.10 – EXERCÍCIO JOGO DOS 7 ERROS.....	52
FIGURA 6.11 – EXERCÍCIO ESCOLHA DE ÍCONES.....	52
FIGURA 6.12 – RESULTADOS DO TESTE DE VALIDAÇÃO .....	54

Formatado

Formatado



## **RESUMO**

Este Trabalho de Conclusão de Curso tem a finalidade de desenvolver um tutorial capaz de apresentar conceitos e informações para a implementação e avaliação de interfaces de programas de computador. Para tanto, utiliza recomendações ergonômicas que envolvem o uso de cores, formas, modos de disposição e saída da informação, além do uso correto de textos. O resultado final do trabalho é um software (tutorial) que permite a programadores e estudantes de Ciências da Computação a familiarização com as técnicas de desenvolvimento de interfaces gráficas, através de pesquisas no tutorial e também pela realização de exercícios.

## ABSTRACT

This course conclusion work has the finality to develop a tutorial able to show concepts and information for the implementation and valuation of computer programs interfaces. For that, utilizes ergonomic recommendations that involve the use of colors, forms, ways of information disposition and information output and also the correct use of texts. The final result of the work is a software (tutorial) that permits to the programmers and students of computing sciences the familiarization with the development graphical interfaces techniques, through researching at the tutorial and also for the exercises realization.

Excluído: ¶

# 1 INTRODUÇÃO

Nos primeiros computadores, onde o processamento era feito em lotes (*batch*), a interação com o usuário era muito pequena e, quando havia, muito difícil e pouco natural. Com a descentralização do processamento, a necessidade de interação aumentou consideravelmente, gerando dispositivos e métodos de interação mais eficazes.

Com a difusão da informática em praticamente todos os ramos do conhecimento humano, a forma de utilização do computador modificou-se profundamente. Dentre os aspectos mais visíveis desta modificação, destaca-se o componente de interação do usuário com o sistema, ou seja, a interface com o usuário.

A difusão do computador como ferramenta de trabalho e a técnica de orientação a objetos (OO), dentre outros avanços, proporcionaram o surgimento dos ambientes gráficos, base para a criação das interfaces gráficas com o usuário (*Graphical User Interface – GUI*). Outro fator que alavancou o desenvolvimento das interfaces GUI é a demanda por esse produto. As pessoas investem em hardware para suportar interfaces mais amigáveis (Minasi, 1994).

Atualmente, o *design* de interfaces é bastante estudado, mas é apenas uma das matérias de uma nova disciplina chamada Interação Homem/Computador (IHC – *Human Computer Interaction* em inglês). A IHC estuda os fenômenos envolvidos no *design*, implementação e avaliação de sistemas computacionais para uso humano. Para isso, é importante que o profissional dessa área, o *designer*, reúna conhecimentos em áreas distintas como psicologia, ciência cognitiva, sociologia, lingüística, *design*, ciência da computação, engenharia, ergonomia, realidade virtual, percepção visual, computação gráfica e semiótica (Baranauskas, 1998; Labiutil, 2000; Reis, 2000).

No curso de Bacharelado em Ciências da Computação já foram desenvolvidos alguns Trabalhos de Conclusão de Curso (TCC) nessa área de IHC, tais como o *Tutorial de Recomendações Ergonômicas Aplicadas a Implementação de Software*, apresentado por Camilo (1995) e ainda a proposta apresentada por Pasold (2000), na qual sugere-se a utilização de um gerador de *Home Pages* de apoio ao ensino, utilizando princípios de *design* de interfaces. Já em Nikel (1994), é proposta a criação de um formalismo para a implementação de interfaces gráficas, levando em consideração os modelos mentais das atividades dos usuários, a fim de facilitar a utilização dos programas de computador.

No presente trabalho, o objetivo é gerar um tutorial para auxiliar o processo de aprendizagem dos desenvolvedores de software na construção de interfaces mais “amigáveis” e de fácil compreensão, tomando como base as recomendações ergonômicas encontradas na literatura (Camilo (1995); Labiutil (2000); Nickel (1994)).

Para o desenvolvimento do tutorial proposto será utilizada a metodologia *Object Oriented Hypermedia Design Method* (OOHDM), apresentada em Auri (2002), Aragão (2002) e Schwabe (1996a). A linguagem de programação utilizada será o *Object Pascal*, da *Borland*, no ambiente *Delphi5*, com conceitos e técnicas absorvidas de Cantu (2000) e Cornell (1996).

## 1.1 OBJETIVOS DO TRABALHO

O objetivo principal deste trabalho é desenvolver um tutorial com recomendações sobre os conceitos de *design* de interface.

Os objetivos específicos são:

- a) permitir que o usuário avalie, através de exercícios fornecidos pelo tutorial, seus conhecimentos sobre o conteúdo apresentado;
- b) permitir a pesquisa das recomendações ergonômicas, indexados por área;
- c) permitir a navegação por telas que mostram a maneira correta e a errada de construção de interfaces, possibilitando a visualização das diferenças entre ambas;
- d) permitir a apresentação de exemplos práticos de cada recomendação ergonômica;
- e) validar a metodologia OOHDM no desenvolvimento de tutoriais.

## 1.2 MOTIVAÇÃO E JUSTIFICATIVA

A relevância do presente trabalho reside no fato do tema abordado ser, normalmente, pouco explorado pelos desenvolvedores, mas que influencia diretamente na utilização do aplicativo de *software* pelo usuário final. Adicionalmente, as recomendações ergonômicas são muito abstratas, dificultando sua compreensão por parte do desenvolvedor de software. Assim, a apresentação, no tutorial proposto, de exemplos práticos para cada recomendação, facilita a compreensão e aplicação.

Em termos de desenvolvimento do protótipo, a relevância está na validação da metodologia OOHDH (*Object Oriented Hypermedia Design Method* – Método Orientado a Objetos para Design Hipermédia), para o desenvolvimento de tutoriais multimídia.

### 1.3 ESTRUTURA DO TRABALHO

No capítulo 1 mostra-se a introdução, na qual é explorada a origem da preocupação dos estudiosos com relação ao desenvolvimento de interfaces gráficas. A seguir são apresentados os objetivos, a motivação e a justificativa do trabalho.

No capítulo 2 são abordados os conceitos relacionados a *design* de interfaces.

No capítulo 3 são apresentadas as recomendações para o correto desenvolvimento de “interfaces amigáveis”, relacionados a diversos aspectos como a disposição da informação, cores, formas, textos e saída de informação. No final, ainda são abordadas as diversas formas de ilusão de óptica e falsa percepção de cores pelo olho humano.

O capítulo 4 apresenta o método de especificação OOHDH (*Object Oriented Hypermedia Design Method*).

O capítulo 5 é destinado a especificação do tutorial propriamente dito.

No capítulo 6 são expostas as técnicas e ferramentas utilizadas e ainda a funcionalidade da aplicação, com apresentação das principais telas do tutorial.

O capítulo 7 é destinado à conclusão e sugestões para trabalhos futuros.

Excluído: ¶

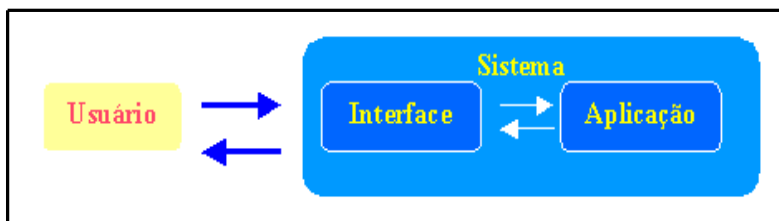
## 2 FUNDAMENTOS DE INTERFACES GRÁFICAS

### 2.1 CONCEITOS

Um bom conceito de *design* é encontrado em Reis (2000), segundo o qual “O *design* é uma atividade criativa que procura fazer um ambiente artificial harmônico que ofereça a mais completa satisfação das nossas necessidades físicas e espirituais”.

Interfaces são meios pelos quais entidades distintas podem interagir entre si sem, no entanto, estabelecerem contato direto, como um tradutor proporcionando interação entre duas pessoas que falam línguas diferentes. A interface intermedia o diálogo do usuário com a funcionalidade da aplicação. É um componente separado da lógica da aplicação, conforme mostrado na Figura 2.1.

**FIGURA 2.1 - ESQUEMA USUÁRIO-INTERFACE-APLICAÇÃO**



Pode-se considerar a interação homem/computador como um processo de comunicação entre dois sistemas cognitivos que fazem tratamento de informação simbólica. De um lado o homem, cuja estrutura cognitiva trata representações, portanto símbolos, da realidade. De outro o computador, visto como uma máquina simbólica que realiza tratamentos de sinais criados pelos programadores para produzir os sinais que os usuários interpretam e manipulam em suas interfaces (Labiutil 2000).

### 2.2 COMO PROJETAR UMA INTERFACE GUI

Segundo Minasi (1994), duas regras devem ser observadas:

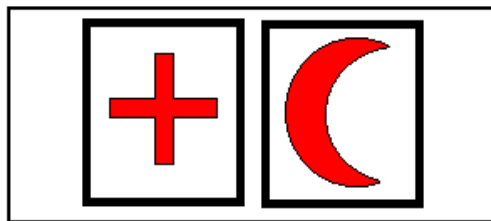
- uma interface GUI não deve ser “percebida” pelo usuário, isto é, quanto mais natural e fácil de interpretar, melhor será o desempenho funcional do software.

- todo bom *designer* sabe quando quebrar as regras!

Não existe, em termos absolutos, uma forma “certa” de se construir uma interface gráfica. Existem, porém, princípios orientadores que devem ser levados em consideração na hora da implementação. Deve-se lembrar que a prototipação também pode ser usada para adequar a interface aos usuários potenciais, principalmente no que diz respeito à subjetividade e características ambientais. Alguns princípios são brevemente descritos a seguir em ordem alfabética, sem, no entanto, atribuir-se maior ou menor importância a cada um:

- **Ajustes à diversidade** - Símbolos, cores e formas têm significados diferentes em culturas diferentes. Deve-se levar em conta também deficiências e diferenças de comportamento, percepção, habilidade e capacidade de memória dos usuários. De acordo com Camilo (1995), a eficácia do ícone como veículo de transmissão de mensagens (comunicação visual) está intimamente ligado ao contexto onde está representado. O significado de um ícone depende das associações que ele causa na mente do usuário. A mesma mensagem pode ter associações muito diferentes em culturas diferentes. Para a utilização de ícones, deve-se selecionar aqueles capazes de transmitir a idéia desejada da mesma forma para todo o público alvo. Um exemplo a ser observado é o da Organização Internacional Cruz Vermelha, mostrado na Figura 2.2. Nos países ocidentais, este símbolo está associado ao cristianismo. Já nos países árabes, de religião muçulmana, a associação correspondente à cruz vermelha chama-se Crescente Vermelho e é representada por uma lua crescente vermelha no lugar de uma cruz.

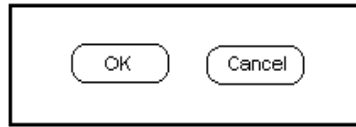
**FIGURA 2.2 - SÍMBOLO DA CRUZ VERMELHA INTERNACIONAL –  
DIFERENÇA CULTURAL.**



Fonte: Camilo (1995)

- **Atratividade** – A inserção de controles 3-D a partir do Windows 3.x tornou este ambiente mais atrativo (Figura 2.4), quando comparado às versões anteriores (Figura 2.3) (Minasi, 1994). Interfaces simples, porém, adequadas ao programa e ao público alvo também são mais atrativas.

**FIGURA 2.3 - BOTÕES ANTIGOS DO WINDOWS**



**FIGURA 2.4 - BOTÕES ATUAIS DO WINDOWS**

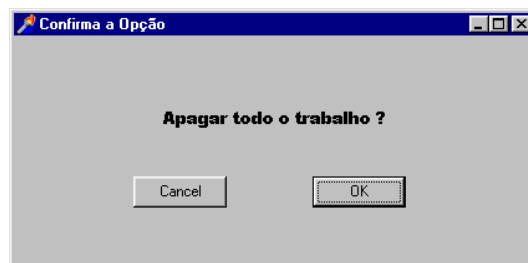


- **Ergonomia e Design** - Qualquer produto que seja projetado para uso do homem, deve ser concebido através de uma análise ergonômica, a qual considera os fatores humanos nas suas interações. Além disso, a participação do *design* gráfico proporciona uma adequação dos elementos gráficos de interação que compõem as telas do aplicativo (Reis, 2000).
- **Harmonia** - Como o próprio nome indica, a harmonia é o princípio relacionado com a busca do conforto para o usuário através de elementos como cor, forma, fonte, os quais trabalham em conjunto para conseguir este conforto. A harmonia é obtida, portanto, através da manipulação correta dos elementos com os princípios de *design*, de maneira uniforme e organizada, considerando os fundamentos da percepção humana (Reis, 2000).
- **Integração com outros aplicativos** – Suportar diversos formatos (rtf, bat, txt, doc) e permitir interação com o recurso de *Clipboard* (Ctrl C, Ctrl V), por exemplo, sempre são muito úteis e procurados (Minasi, 1994).



- **Legibilidade** – As telas ainda são mais difíceis de ler do que o papel, portanto os tipos e tamanhos de fontes, além de combinações de cores em primeiro e segundo planos, devem ser bem escolhidos. Testes em diferentes tipos de monitores (do monocromático ao *Display* de Cristal Líquido - LCD) também podem ser feitos a fim de se determinar as melhores combinações (Minasi, 1994).
- **Personalização** – Esta característica é inerente ao ser humano, portanto deve ser levada em conta. É recomendável que o usuário sempre tenha a possibilidade de personalizar a interface e o software como um todo às suas preferências.
- **Preditabilidade** – Prever o que usuário quer ou vai fazer também é outro aspecto importante de uma boa interface. Caixas de listas que lembram quais foram os últimos tópicos pesquisados, campos de entrada telescópicos (que filtram as opções conforme se vai digitando a palavra) e padronização na ordem de colocação de botões ajudam a antecipar ações do usuário. Uma ação errada deve ser prevista e devidamente tratada, a fim de fornecer uma opção de saída de emergência bem definida (Minasi, 1994). A caixa de diálogo da Figura 2.5, por exemplo, é pouco adequada por dois motivos:
  1. a ação padrão é o botão OK, o que representa a eliminação do trabalho em questão. É recomendado que quando envolver eliminação de arquivos ou partes de um trabalho, a opção padrão seja a “resposta negativa”.
  2. o botão OK está à direita, quando deveria estar à esquerda.

**FIGURA 2.5 - UMA CAIXA DE DIÁLOGO INADEQUADA**



Estes foram alguns princípios para construção de interfaces. Mais adiante, será dedicado um capítulo somente sobre recomendações para construção de interfaces gráficas.

## 2.3 ELEMENTOS DE UMA INTERFACE GRÁFICA

Neste tópico são introduzidos conceitos dos principais elementos que compõem a interface gráfica do *MS-Windows (Microsoft)*, pois esta é a mais difundida na realidade atual.

São abordados, nesta ordem, tópicos como as Janelas, que são o elemento central da interface gráfica *MS-Windows*, colocando-se os diversos tipos de janelas utilizadas. Após isso, é exposto o elemento Caixa de Edição, que é também bastante explorado nas interfaces gráficas. De Reis(2000), extraiu-se o exposto sobre as diferenças existentes entre os *Bitmaps* e Desenhos. Adicionalmente serão abordados os menus e as caixas de mensagens. Por fim, são exemplificados os Controles e as Barras de ferramentas, sendo colocados inclusive figuras para ilustrar os conceitos citados.

### 2.3.1 WINDOWS (JANELAS)

O elemento central da interface gráfica do *MS-Windows* é a própria janela, podendo ser dos seguintes tipos:

- **Janelas de aplicativo:** Estas compreendem a estrutura visual para dados e comandos de um aplicativo. Elas são o que se vê quando um aplicativo é aberto. A maioria das ações em um aplicativo ocorre em uma janela.
- **Janela de Interface de Múltiplos Documentos (MDI - *Multiple Document Interface*):** Este tipo de janela permite que o aplicativo manipule mais de um documento (ou uma cópia do mesmo documento) de cada vez. Apenas uma destas janelas está ativa de cada vez.
- **Janelas de ajuda:** Estas são na verdade aplicativos independentes, e podem ser movidos para fora da janela do aplicativo primário.
- **Caixas de diálogo:** Alguns comandos do aplicativo exigem mais informação do que uma simples tecla pode fornecer. Quando for este o caso, o acionamento de tal comando produz uma caixa de diálogo que pede ao usuário mais informações e contém os controles necessários para fornecê-las.

## 2.3.2 CAIXAS DE EDIÇÃO

A evolução das interfaces gráficas e a ampliação de ferramentas e bibliotecas disponíveis nos sistemas operacionais, alavancou a editoração eletrônica de textos com aparência tipográfica. Assim, editores de texto que apresentavam somente fontes do tipo *courier*, passaram a não ser mais encontrados e até mesmo aceitos pelos usuários. Estes preferem agora editores os quais ampliam as possibilidades de fontes como *arial*, *Times*, *Impact*, dentre outras.

As caixas de edição são janelas nas quais o usuário pode digitar pequenos textos dentro da interface do aplicativo. Da mesma forma que os editores de texto, estas caixas de edição evoluíram. Nos aplicativos em que elas estão presentes é possível, atualmente, executar diversas edições até então só realizadas nos softwares com funções gráficas específicas, o que permite um processamento de textos mais apurado.

## 2.3.3 DESENHOS X BITMAPS

As interfaces gráficas se baseiam na máxima de que uma imagem vale mais que mil palavras. Dentro do universo GUI, existem duas formas de se armazenar e construir imagens:

- a) como *bitmaps*, também conhecidos como *rasters*;
- b) como desenhos, também conhecidos como vetoriais, ou diagramas de objeto.

A diferença é simples, mas importante. Com um *bitmap*, uma imagem é armazenada como uma coleção de pontos (*Pixels* ou *Pels*), que podem ser apresentados na tela ou em uma impressora. Os desenhos são armazenados como uma série de objetos, instruções sobre como desenhar a imagem.

Tanto os arquivos de desenho como os *bitmaps* possuem seu lugar no mundo GUI. As desvantagens associadas aos *bitmaps* são o tamanho dos arquivos (geralmente grandes) e a dependência da resolução. Como vantagens, o *bitmap* pode ser manipulado com precisão e "Anti-Aliased" (um processo que suaviza a aparência de um *bitmap*).

Os desenhos, por outro lado, são fáceis de armazenar, independentes da resolução, e fáceis de revisar, tornando-os apropriados para situações que irão exigir revisões e impressões em tamanhos diferentes, como por exemplo, desenhos técnicos.

Os exemplos de formatos de arquivo *bitmap* no *Windows* são:

- TIFF (*Tagged Image File Format*): As imagens escaneadas são normalmente armazenadas no formato TIFF.
- PCX (*Paintbrush*) é um antigo formato *bitmap* em que muitos *bitmaps* são distribuídos.
- BMP é o formato *bitmap* do MS-Windows, adaptado do formato *bitmap* desenvolvido pela *Microsoft* para o OS/2.

Os formatos comuns de arquivos de desenho do MS-Windows incluem:

- WMF (*Windows Metafile*), adaptado de um formato de imagens desenvolvido pela *Microsoft* para o OS/2.
- DXF, um formato da *AutoDesk* normalmente utilizado para exportar arquivos Auto CAD.
- CGM (*Computer Graphics Metafile*), usado pelo *Freelance* e outros pacotes.
- EPS (*Encapsulated PostScript*), uma linguagem vetorial baseada na linguagem de controle das impressoras *PostScript*.

#### 2.3.4 MENUS

Os menus são uma parte essencial do que torna as GUI atrativas. Se todas as funções de um programa forem apresentadas em um menu, então existirá um incentivo para se olhar dentro de cada uma delas, para se saber exatamente o que cada uma faz.

Na maioria dos ambientes do tipo janela, normalmente se vê:

- Dois níveis padrão de menus;
- Menus em cascata (que se ramificam em outros itens de menu);
- Menus destacáveis (que podem ser movidos para outras localizações na tela).

#### 2.3.5 CAIXAS DE MENSAGEM E DE DIÁLOGOS

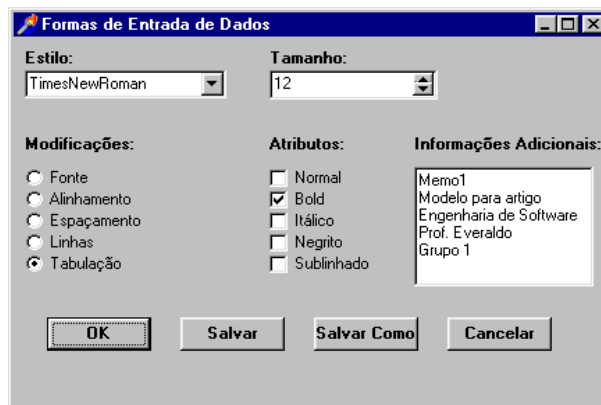
Os programas *MS-Windows* podem exibir outras pequenas janelas que pedem ou fornecem informações. Estas janelas são chamadas de caixas de diálogo. Uma caixa de diálogo que fornece apenas informações, oferecendo um botão OK (uma mensagem de erro, por exemplo) tem o nome específico caixa de mensagem. Elas são tão comuns que

normalmente são tratadas como algo separado das caixas de diálogo, mas na verdade, são uma forma especial de caixa de diálogo.

### 2.3.6 CONTROLES

O conteúdo das caixas de diálogo é, normalmente, formado por itens gráficos que fornecem alternativas de entrada. As entradas simples são executadas apenas com botões, mas existem diversos outros itens de opção de entrada, chamados de controles, como mostrados na Figura 2.6.

FIGURA 2.6 - JANELA CONTROLES



### 2.3.7 BARRAS DE FERRAMENTAS

Outra característica pós-versão 3.1 que se vê freqüentemente nos programas *MS-Windows*, são as barras de ferramentas, uma tira de ícones que ativa itens de menu. Elas se tornaram indispensáveis, mas nem sempre são bem-feitas. Um exemplo de Barra de ferramentas bem preparada se vê na figura 2.7, que mostra a barra de ferramentas do *MS-Windows Explorer*. Neste aplicativo, ao passar o cursor por sobre o ícone, é aberto um texto abaixo do mesmo mostrando qual a sua finalidade ou aplicação.

FIGURA 2.7 - BARRA DE FERRAMENTAS DO MS-WINDOWS

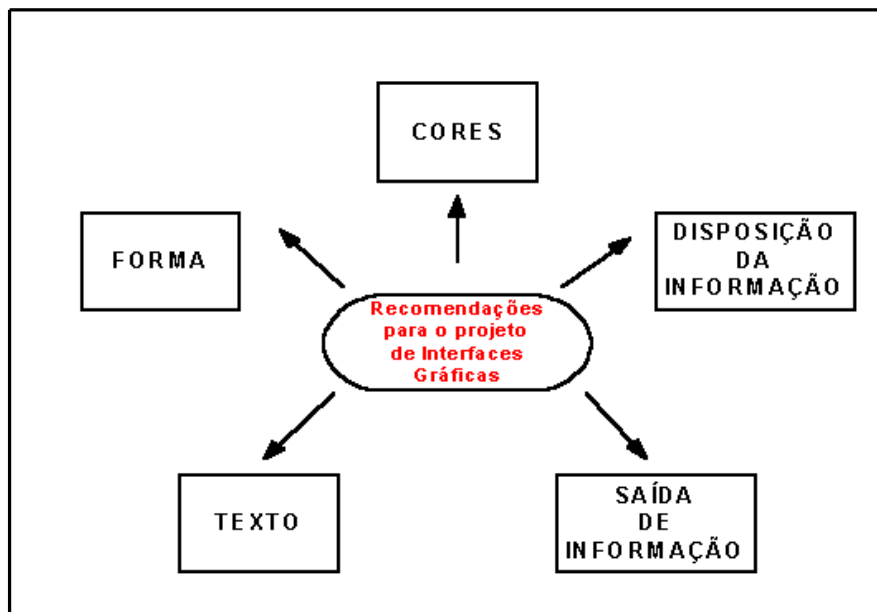


### 3 RECOMENDAÇÕES PARA O PROJETO DE INTERFACES GRÁFICAS

Neste capítulo, serão citadas diversas recomendações para tornar a interface gráfica mais agradável e o programa mais produtivo, tomando-se como base os trabalhos de Reis (2000), Labiutil (2000) e Baranauskas (1998).

Para facilitar a compreensão, foram dispostos em tópicos separados, indexados por tipo, recomendações com relação à disposição da informação, forma, uso de cores, texto e saída de informações, conforme ilustrado na Figura 3.1.

**FIGURA 3.1 - ESQUEMA DAS RECOMENDAÇÕES PARA O PROJETO DE INTERFACES GRÁFICAS**



Em algumas das recomendações são apresentados exemplos práticos do aspecto em questão. A decisão de se colocar ou não um exemplo baseou-se na complexidade da recomendação. Finalizando o capítulo, é abordado o fator "Ilusão de Óptica", mostrando a influência deste item na concepção de interfaces gráficas.

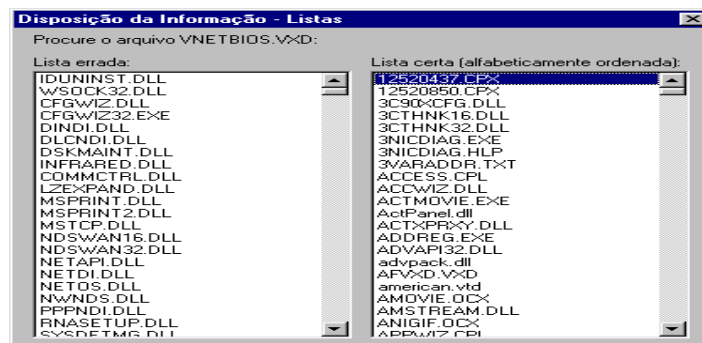
### 3.1 DISPOSIÇÃO DA INFORMAÇÃO

A disposição da informação em forma não-linear, a utilização de interfaces padronizadas, a possibilidade de seleção de informações de interesse e conexões múltiplas a outras informações, não é característica exclusiva dos *softwares* aplicativos. Os jornais, revistas e outros meios de comunicação já trabalham isto através das manchetes, dos textos de chamada da primeira página, da numeração das páginas, da divisão dos cadernos por assunto específico, dentre outras características. Esses mecanismos permitem uma atenção “flutuante”, ou de interesse potencial pela informação. Ao deparar com a primeira página de um jornal, o leitor "passa os olhos" pelas manchetes, seleciona o assunto e através de mecanismos padronizados de busca, "acessa" a informação de seu interesse, sem necessariamente ter de examinar todo o conteúdo do jornal (Mendes, 2001).

Da mesma forma, os aplicativos de *software* devem considerar a disposição da informação, adotando os cuidados a seguir:

- a) Evitar que o usuário tenha que realizar conversões, traduções ou interpretações. Isto irá trazer desconforto na utilização, pois, nesse caso, o usuário é obrigado a buscar uma informação que deveria estar disponível.
- b) Permitir personalização dos modos de apresentação da informação, de acordo com as características dos usuários. Para usuários novatos, por exemplo, utilizar uma interface com condução por exemplos; já para os experientes, permitir a utilização de macro funções. Um exemplo deste tipo de disposição é a proposta pela Microsoft em seus programas de instalação de aplicativos, que permite instalação avançada e típica.
- c) Adotar princípios lógicos para ordenação de listas conforme mostrado na Figura 3.2. A ordenação alfabética somente deve ser utilizada quando não for possível a adoção de outro princípio lógico, mais coerente com o tipo de dados buscados.

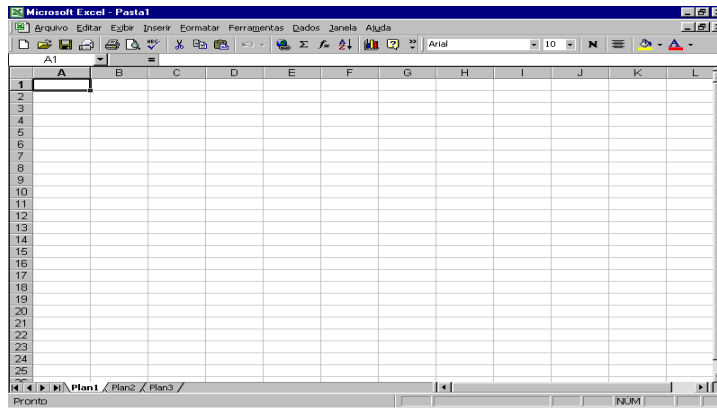
**FIGURA 3.2 - PRINCÍPIOS DE ORDENAÇÃO DE LISTAS**



- d) Fornecer documentação em tela sobre o que, onde e como fazer para prosseguir a transação. Não se pode deixar o usuário com dúvida sobre como realizar uma dada atividade. Ele sempre deve saber onde está e como fará para prosseguir na interação.
- e) Preencher no máximo 25% da tela com elementos gráficos e textuais. Se a tela está muito cheia fica bastante difícil encontrar o que se procura.
- f) Minimizar o número de divisões principais da tela. Agrupar as informações em blocos lógicos e graficamente hierarquizados facilita a ação e a memorização. De acordo Labiutil (2000), nossa memória de curto termo (memória intermediária, onde tratamos as informações recebidas para memorização ou descarte) tem normalmente a capacidade limitada ao número de sete (mais ou menos dois) itens.
- g) Identificar a tela com um título breve e explicativo. Uma hierarquia de telas com seus respectivos títulos traz uma melhor definição dos mesmos, contribuindo para homogeneidade e clareza.
- h) Usar o menor número de colunas possíveis na composição das telas e janelas. Conforme Labiutil (2000), o usuário suporta com tranqüilidade até quatro colunas com doze linhas cada, para as telas apresentadas.
- i) Posicionar o cursor no primeiro campo de entrada de dados, conforme mostrado na Figura 3.3. O usuário não deve ir com o cursor até a posição onde deverá digitar alguma entrada de informação, evitando a perda de tempo.



**FIGURA 3.3 - CURSOR POSICIONADO NO 1º CAMPO NA ABERTURA DO PROGRAMA.**



- j) Apresentar na tela somente a informação necessária para executar a tarefa. As informações desnecessárias, além de aumentar a demora na escolha de uma opção, preenchem a memória com informações irrelevantes. Além disso, o usuário não deve precisar buscar informações em outra tela para realizar uma tarefa.
- k) Organizar as informações por nível de importância. Deve-se disponibilizar nas telas as informações realmente relevantes naquele momento, deixando as demais menos visíveis.

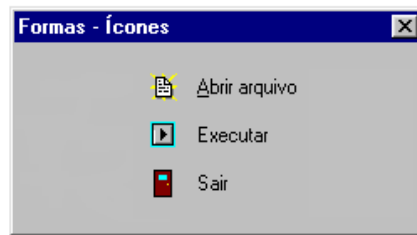
## 3.2 FORMA

Os conceitos com relação à forma estão ligados aos tipos de figuras ou ícones que são escolhidos para representar funções ou atividades do *software*. O uso de escala, “piscagem”, agrupamento de componentes e contraste são considerados quando se trata a forma da interface do aplicativo de software que se pretende criar. Nesse sentido, pode-se fazer as seguintes recomendações:

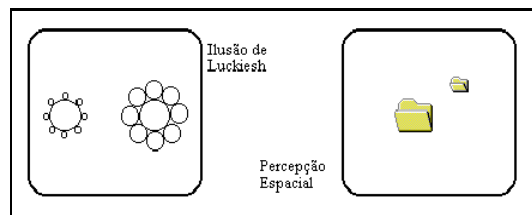
- a) Criar identidade visual para o software. O uso coerente das formas e cores entre os elementos e entre as telas personaliza o software e facilita a navegação. É importante que o software tenha características próprias com relação às cores e o formato gráfico dos objetos, garantindo uma uniformidade e harmonia na interface. Quebras de

padrão devem ser utilizadas apenas com o objetivo de chamar a atenção num determinado ponto da interface.

- b) Cuidados com agrupamento de componentes. Deve-se agrupar apenas itens correlatos, tornando a tela mais limpa. Elementos que estejam muito próximos entre si perdem sua identidade e são percebidos como se o grupo fosse um só elemento.
- c) Usar contraste para criar distinção e prevenir ambigüidades. O contraste é utilizado para evidenciar um determinado objeto na tela. As interfaces que não fazem uso do contraste correm o risco de deixarem imperceptíveis os componentes de importância na tela. Quando um ícone não possui contorno ele pode ser delimitado através do contraste do seu interior com o seu redor.
- d) Usar direção para distinguir e para indicar movimento. Um exemplo de distinção é a utilização de setas iguais em direções diferentes para denotar caminhos ou atividades diferentes. Já um símbolo que indica direcionamento repetido várias vezes, transmite ao usuário a sensação de movimento.
- e) Usar cor para distinção rápida e/ou agrupamento entre itens. As cores são rapidamente percebidas pelos usuários. A diferenciação da cor entre os elementos e seu entorno deve ter contraste suficiente para que não ocorram ambigüidades e problemas de visibilidade. A cor também pode ser utilizada para formar agrupamento de itens.
- f) Usar piscagem para codificar com alto grau de distinção. A piscagem tem o poder de chamar a atenção do usuário devido à indicação de uma advertência ou anomalia no sistema. No entanto, ela não deve ser utilizada em excesso, porque irrita o usuário e causa perda de impacto, ou seja, a piscagem passa a não chamar mais sua atenção e deixa de indicar um comportamento anômalo.
- g) Usar ícones e símbolos que realmente indiquem a ação. A descrição textual se faz necessária quando o ícone ou elemento gráfico não é claro o suficiente. A descrição deverá vir logo abaixo ou ao lado do ícone que é identificado, conforme mostrado na Figura 3.4

**FIGURA 3.4 - ÍCONES DE ACORDO COM A FUNÇÃO.**

- h) Evitar a possibilidade de ocorrerem ambigüidades na leitura do ícone ou símbolo. Como já citado, os símbolos podem assumir diferentes significados em ambientes diferentes. É necessária familiarização com o ambiente para evitar o problema. Além disso, o símbolo deve ter sempre o mesmo significado durante a interação.
- i) Usar tamanho e escala para sugerir tridimensionalidade nos ícones e símbolos. Este tipo de recomendação é baseada no princípio de tamanho e escala (Figura 3.5), onde o efeito visual pode provocar a sensação de tridimensionalidade nos ícones, permitindo a elaboração de desenhos com maior perfeição e mais próximo à realidade.

**FIGURA 3.5 - TAMANHO E ESCALA DOS OBJETOS**

- j) Usar gradação para denotar movimento ou transição na construção de símbolos e ícones. Um exemplo de gradação é a transformação das formas do maior para o menor, sugerindo a presença de movimento ao símbolo que forma o ícone.

### 3.3 CORES

O emprego das cores na concepção de interfaces homem-computador tem sido alvo de numerosas recomendações ergonômicas. Elas aconselham o uso de cores para transmitir informações, chamar a atenção, contrastar e associar objetos de interação. O uso puramente

decorativo é desaconselhado. Para que sua utilização seja eficaz, deve-se tomar cuidado com três aspectos:

- i) a legibilidade final da informação;
- ii) com os efeitos das cores sobre a performance cognitiva do usuário;
- iii) com as possibilidades dos dispositivos físicos.

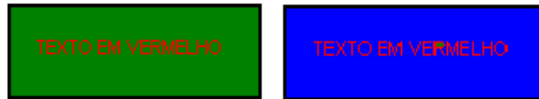
Estas precauções visam conter a confusão visual resultante do emprego arbitrário e exagerado de cores não pertinentes (Labiutil 2000).

Em termos de recomendações específicas sobre o uso de cores, pode-se citar as seguintes:

- a) Criar e aplicar cores em etapas separadas. Criar as telas cuidando da codificação da informação pelo seu conteúdo e forma, e só depois adicionar cuidadosamente cores onde elas possam ser úteis ao usuário.
- b) Usar preferencialmente informações em preto ou em cores, sobre o fundo claro. A profundidade de foco da visão do usuário é maior, o cansaço visual é menor, e a leitura das telas para preenchimento de campos é mais fácil.
- c) Usar no máximo sete e no mínimo três cores de maneira planejada para cada tela. A cor tem o poder de agrupar por distinção, havendo muitas cores se perceberá vários grupos distintos e dificultará na memorização.
- d) Relacionar a atribuição da cor à importância da informação. Usuários intuitivamente percebem o vermelho como plano frontal, o verde como plano intermediário e o azul como fundo de tela. Cores brilhantes são adequadas para sinais de perigo, chamar a atenção para um ponto, facilitando a memorização de elementos. Deve existir uma hierarquia de cores, com áreas brilhantes, neutras e de baixa iluminação, coerentes com a importância das informações.
- e) Usar combinações legíveis para textos ou figuras e fundos. As combinações: preto/branco, amarelo/preto, branco/vermelho, preto/amarelo, ciano/magenta, amarelo/marrom, ciano/marrom, branco/preto, verde/preto, verde/vermelho, preto/verde, branco/marrom, verde/marrom, magenta/marrom, dispostas, respectivamente, texto e fundo, são legíveis. Nestas combinações cromáticas estão presentes considerações de design gráfico.

- f) O uso de combinações de certas cores pode criar pós-imagens (fantasmas) no contorno da figura. Um exemplo disso é mostrado na figura 3.6, onde são combinadas as cores verde e vermelho e ainda azul e vermelho.

**FIGURA 3.6 - USO INADEQUADO DE COMBINAÇÃO DE CORES.**



- g) Usar cores similares para fundos de áreas que tenham relações fortes entre si. Esta combinação cromática contribui para que o usuário estabeleça relação conceitual entre as áreas, buscando harmonia e agrupamento.
- h) Considerar as características físicas do sistema visual do usuário na atribuição das cores. Uma das características é que a área central do campo visual é mais sensível ao verde e ao vermelho, e a periferia do campo é mais sensível ao azul, preto, branco e amarelo. O daltonismo pode ser auxiliado utilizando-se fundo verde e letras vermelhas sobre ele, permitindo a percepção de texto e fundo. Idosos e usuários que operam terminais por longo período percebem melhor cores saturadas, ou seja, com bastante pigmentação.
- i) Considerar os níveis culturais na codificação cromática. O significado das cores também pode variar culturalmente: em países asiáticos o branco denota luto; as caixas de correio na Inglaterra são vermelhas e na Grécia, amarelas. É importante considerar a cultura do usuário evitando más interpretações.
- j) Considerar o contexto profissional das cores. As conotações das cores podem ser convencionadas. Por exemplo, vermelho: pare, perigo, quente, fogo; amarelo: cuidado, vagaroso, teste; azul: frio, água, tranquilidade, gelo; cinza, branco e azul: discrição.
- k) Permitir personalização quanto às cores. Não se pode agradar todos os usuários quanto ao esquema cromático. Assim, é importante que o usuário possa atribuir cores aos elementos da interface mediante suas necessidades e gosto.

- l) Considerar restrições do ambiente e dos equipamentos na codificação cromática. É importante saber que a iluminação do ambiente pode interferir na percepção das cores. Os monitores de vídeo perdem sua capacidade de coloração com o uso através do tempo.

### 3.4 TEXTO

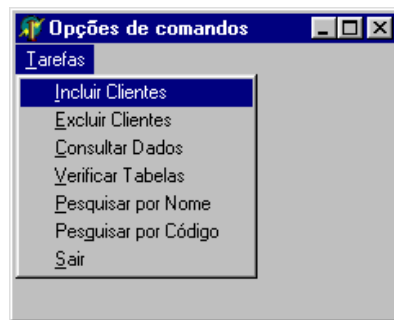
O texto que é colocado nas janelas e caixas de mensagens do *software* deve obedecer alguns critérios com relação a amenidades sociais, legibilidade, espaçamento e consistência de termos utilizados. Além disso, a simplicidade dos títulos e a consistência de comandos e frases devem ser observadas. Assim, são expostas abaixo algumas recomendações para que estas premissas sejam cumpridas:

- a) Eliminar amenidades sociais nos comandos e menus. O uso de expressões como "por favor", "bom dia", "para incluir pressionar o botão inclusão", etc., podem ser perfeitamente eliminados.
- b) Imprimir legibilidade ao texto. As maiúsculas devem ser utilizadas em palavras que se deseja dar ênfase, e não no texto inteiro. Para garantir compatibilidade com os formulários de dados, deve-se colocar letras escuras sobre fundo claro. Usar no máximo quatro a três tamanhos de letras e três fontes distintas nos textos a serem apresentados.
- c) Padronizar espaçamento no texto para otimizar legibilidade. Os espaçamentos entre as palavras devem ser menores que o espaçamento entre linhas, evitando a formação de manchas brancas ou vazios que interferem na leitura. Deve-se utilizar linhas brancas entre parágrafos para melhorar a clareza e legibilidade.
- d) Usar piscagem nos textos para mostrar erros ou iluminar pontos. A piscagem não deve ser usada com exagero, pois seu uso freqüente reduz a característica de impacto que possui. Além disso, em demasia, a piscagem pode irritar o usuário.
- e) Com relação aos títulos, simplificá-los ao máximo. Deslocar a palavra chave do título para o início do mesmo. Deve-se formar um título o mais simples possível.
- f) Imprimir consistência entre título e lista de opções. Como já colocado anteriormente, é interessante fazer uma hierarquia de chamada de telas sendo que, deve prevalecer a

consistência no título das telas com relação a sua opção de chamada na tela anterior. O título deve ser inequívoco em relação ao conteúdo da tela que encabeça.

- g) Imprimir identidade e legibilidade ao título. O título deve ser, preferencialmente, centralizado ou alinhado à esquerda. Deve-se colocar uma linha em branco entre o título e o resto da tela. O tamanho e tipo de fonte devem ser iguais para os títulos de mesma hierarquia. De maneira geral, deve-se dar harmonia entre tela e títulos, bem como entre os títulos de toda interface.
- h) Usar termos consistentes com relação aos comandos requeridos na interface. Os termos utilizados para ativar opções devem ser bem diferentes entre si para facilitar na memorização e na escolha da opção certa. O uso de termos familiares é importante, pois faz com que o usuário se adapte melhor à interface.
- i) Usar frases consistentes e concisas para as opções que ativam as ações. O usuário se sente mais confortável com opções do tipo “Animal”, “Vegetal” e “Mineral” do que com “Informações sobre Animais”, “Opções disponíveis de Vegetais” ou “Visualizar categorias Minerais”.
- j) Nas opções de comandos, trazer a “palavra-chave” para a esquerda da frase. A palavra-chave de uma opção colocada à esquerda auxilia na velocidade de identificação da mesma, conforme mostrado na Figura 3.7. Entende-se por palavra-chave, neste caso, aquela palavra que melhor identifica a ação a ser executada, considerando o contexto da interface e, ainda, o idioma utilizado no menu.

**FIGURA 3.7 -OPÇÕES DE COMANDO COM PALAVRA-CHAVE À ESQUERDA**



- k) Imprimir coesão aos grupos de textos e enfatizar a separação entre eles. É importante que os grupos de textos na tela sejam bem divididos, por exemplo, por linhas em

branco ou contornos. Recomenda-se utilizar em cada grupo o equivalente a quinze caracteres de comprimento e sete caracteres de altura. Este tamanho está de acordo com a área coberta pelo foco visual na leitura de telas. Deve-se, ainda, usar de seis a quinze grupos por tela.

- l) Com relação à abreviatura, adotar estratégia clara para abreviar, e mantê-la em toda a interface. Truncagem simples: telef (telefone) e teleg (telégrafo). Corte de vogais com truncagem: prgs. (programas). Primeira e última letras: ST (sort). Abreviações de outros contextos: BAK (*backup*), PC (*personal computer*). Através do som produzido pela pronúncia da palavra: CTRL (*control*).
- m) Usar o mínimo de palavras por linha nas tabelas. Estudos indicam que, se for usado um limite de três a quatro palavras por linha, o processo de releitura fica facilitado.

### 3.5 SAÍDA DE INFORMAÇÕES

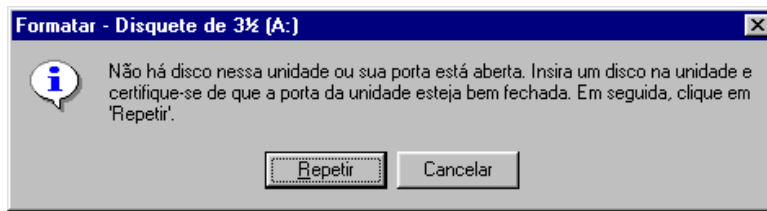
A saída de informação talvez seja o item mais relevante no que diz respeito à confecção de interfaces gráficas com o usuário, haja vista a intensa dependência que esta cria com relação às respostas fornecidas pelo computador às suas interações. Neste caso, o tempo de resposta deve ser informado instantaneamente para não causar uma espera por parte do usuário.

Outros fatores como o uso de linguagem positiva e quantidade de resposta adequada, são abordados neste tópico:

- a) Usar termos precisos e linguagem positiva e construtiva centrada no usuário para as mensagens. É mais conveniente informar ao usuário como resolver um determinado problema. Mensagens como “Entrada Inválida” ou “Operação Inadequada” são insuficientes, pois o usuário fica irritado sabendo que existe um problema e não consegue resolvê-lo. Na Figura 3.8, extraída do *MS-Windows 95*, é mostrada uma janela informativa de que existe um erro com o disquete ou com a unidade impedindo a formatação, entretanto, são colocadas algumas maneiras de se resolver o problema.

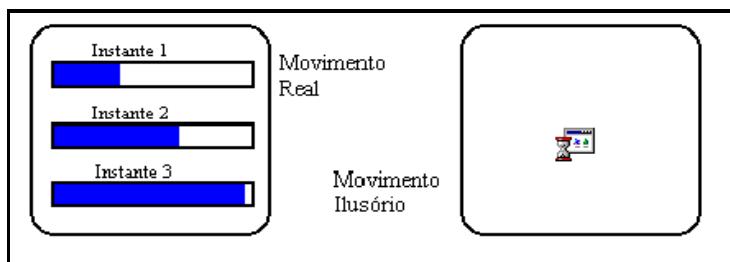


**FIGURA 3.8 - JANELA INFORMANDO AO USUÁRIO COMO RESOLVER O PROBLEMA.**



- b) Fornecer uma quantidade de respostas compatíveis com a capacidade do usuário. De acordo com Labitutil (2000), a capacidade de absorção do usuário está limitada a grupos de cinco a nove informações. Não adianta dar quinze respostas simultâneas na tela para o usuário se o mesmo só irá absorver nove delas, num dado momento.
- c) Para tempos longos de resposta, orientar o usuário com informações na tela. Elementos gráficos com movimento real ou movimento ilusório podem servir para informar o que o sistema está fazendo, num processamento demorado (Figura 3.9).

**FIGURA 3.9 - MOVIMENTO DE OBJETOS**

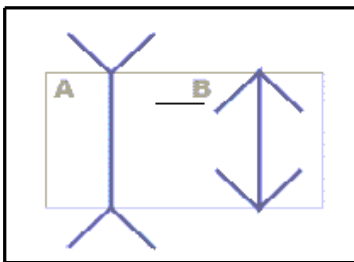


- d) Em se tratando de tempo de resposta ao usuário, respeitar as características do mesmo neste tempo. O tempo de resposta depende de características pessoais, nível de experiência, tipo de atividade a ser realizada, etc. Deve-se considerar esses fatores na determinação de tempos de resposta adequada para quem usa o software.
- e) Atribuir estabilidade aos tempos de resposta. Se o tempo de resposta no sistema for muito instável a *performance* e satisfação do usuário serão prejudicadas. É ruim não poder se acostumar com os tempos de resposta das ações efetuadas no software, frustrando as expectativas de espera pelo processamento.

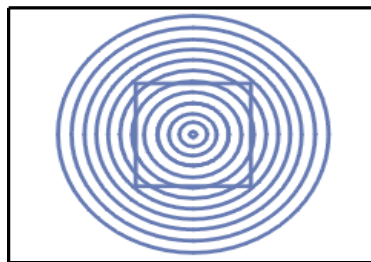
### 3.6 ILUSÃO DE ÓPTICA

Neste item são abordados os diversos tipos de ilusão de óptica que devem ser observados para evitar a implementação de interfaces geometricamente corretas, mas, que causem dificuldades de interpretação por parte do usuário. Dentre os casos de ilusão de ótica, os “problemas” geométricos são os mais freqüentes. Conforme mostrado na Figura 3.10, onde a semi-reta A aparenta ser maior que a B, quando, na verdade, são do mesmo tamanho, e também na Figura 3.11, onde o quadrado central é perfeito, não apresentando a distorção aparente e na Figura 3.12, onde as retas horizontais são paralelas mas aparentam uma distorção que não existe na realidade.

**FIGURA 3.10 - MESMO TAMANHO?**



**FIGURA 3.11 - DISTORÇÃO APARENTE.**

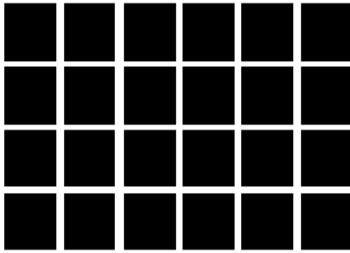
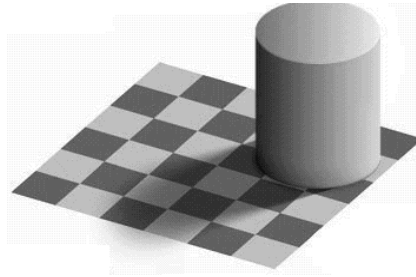


**FIGURA 3.12 - AS RETAS HORIZONTAIS SÃO PARALELAS.**

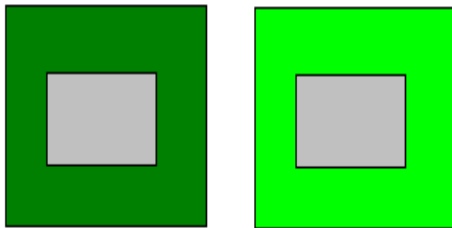
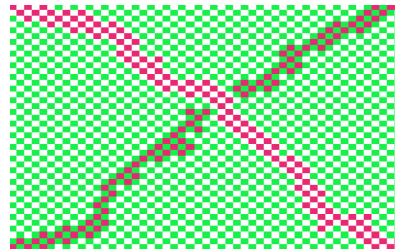


Na Figura 3.13 e Figura 3.14, são mostrados exemplos de ilusão de óptica causada por contrastes de preto, branco e tons de cinza. Na primeira, percebe-se pontos cinza piscando entre as junções dos quadrados pretos. Na segunda, o mais interessante: Os quadrados em tom mais claro da área sombreada têm o mesmo tom de cinza dos quadrados mais escuros da parte de fora da sombra.

Formatado

**FIGURA 3.13 - PONTOS PISCANTES.****FIGURA 3.14 - QUADRADOS COM MESMO TOM DE CINZA**

A seguir, são apresentados os exemplos considerando o contraste com cores. Na Figura 3.15, os quadrados internos possuem o mesmo tom de cinza. Entretanto, a primeira vista, imagina-se que o da esquerda possui um tom mais claro, devido ao tom de verde mais escuro que o envolve. Já na Figura 3.16, existem só duas cores presentes (com só um tom cada uma). É o mesmo caso explicado acima. A colocação das células em posições estratégicas na figura causa a impressão de que há, no mínimo, mais um tom de rosa na figura.

**FIGURA 3.15 - O MESMO TOM DE CINZA****FIGURA 3.16 - SÓ DUAS CORES PRESENTES**

De acordo com o que foi exposto neste capítulo, comprova-se a necessidade da verificação destes itens na construção das interfaces gráficas, pois, caso contrário, o usuário final pode ter problemas para identificar determinadas funções do aplicativo por não entender o seu significado ou ainda por estar sendo “traído” pelos próprios olhos, que, como se percebeu, pode ocorrer facilmente.

Excluído: ¶

## 4 METODOLOGIA OOHDM – OBJECT ORIENTED HYPERMEDIA DESIGN METHOD.

Aplicações hipermídia constituem importantes ferramentas para tratar informações não estruturadas através de uma interface amigável, e que têm sido longamente utilizadas em diversas áreas de aplicação, como, por exemplo, na documentação de sistemas, em sistemas de educação infantil, de treinamento e ensino. Porém, nestas aplicações percebe-se deficiências no que diz respeito ao controle, manutenção e reutilização de um grande volume de informações, já que suas modelagens não seguem os conceitos do paradigma de objetos (Auri, 2002).

Assim, seguindo uma abordagem orientada a objetos em sua modelagem, uma aplicação hipermídia procura fornecer um modelo de estruturação mais natural para as informações, através de classes, que definem a estrutura, e de suas instâncias (nós) que efetivamente mantêm a informação. Desta forma, é possível associar maior semântica às unidades do hiperdocumento, e, por consequência, dar suporte a mecanismos mais sofisticados de consulta e organização dos dados, além de facilitar a reutilização e manutenção.

No desenvolvimento de aplicações hipermídia, há a necessidade de se utilizar um modelo de base, que possa nortear as etapas de construção do projeto e o uso da tecnologia relacionada. Schwabe(1996a) propõe o uso do método OOHDM (*Object-Oriented Hypermedia Design Method*), que é composto por quatro atividades diferentes:

- Projeto conceitual (modela a semântica do domínio da aplicação);
- Projeto navegacional (leva em consideração o perfil do usuário e a tarefa a ser executada, dando ênfase aos aspectos cognitivos);
- Projeto da interface abstrata (modela objetos perceptíveis, implementa as metáforas escolhidas e descreve a interface para os objetos navegacionais);
- Implementação.

Excluído:

O OOHDM utiliza mecanismos de abstração e composição numa arquitetura orientada a objetos, que permite uma descrição concisa das informações complexas e uma especificação de padrões de navegação e interface (Schwabe, 1996a).

## 4.1 PROJETO CONCEITUAL

Em OOHDM o esquema conceitual é constituído sobre classes, relações e subsistemas. As classes são descritas como de costume em modelos orientados a objetos, mas seus atributos podem ser multi-tipados.

O esquema consiste de um conjunto de objetos e classes unidos entre si por relacionamentos; como os objetos são instâncias de classes, um relacionamento entre as classes faz com que o relacionamento entre os objetos seja abstraído. As classes podem relacionar-se com subsistemas. Um subsistema pode ser autocontido, possuindo um único ponto de entrada e saída. Neste caso, age como um "servidor de informações".

Contudo, do ponto de vista da hipermídia, é importante afirmar que os relacionamentos não devem ser escondidos nos atributos de classes. Isto significa que no caso de um atributo representar uma entidade conceitual complexa a ser explorada na hipermídia final, deve-se especificar um relacionamento. Os relacionamentos, assim como as classes, podem conter atributos (Aragão, 2002).

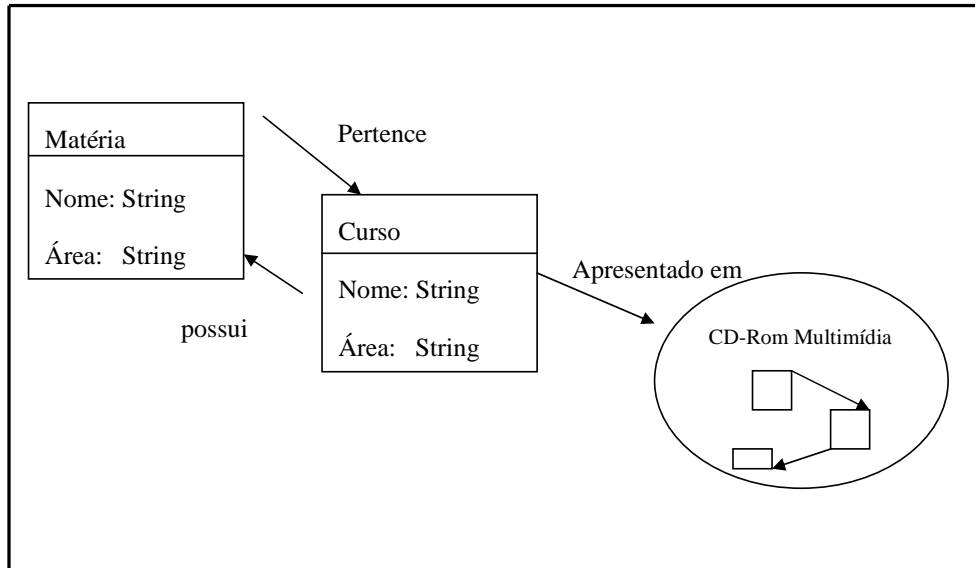
A modelagem oferece três mecanismos de abstração para lidar com a complexidade: Agregação, Generalização/Especialização e um conceito de empacotamento: os subsistemas. O primeiro é útil na descrição de classes complexas como agregadas de classes mais simples. Já o segundo é utilizado na construção de hierarquias de classes e no uso de herança como um mecanismo de compartilhamento. Os subsistemas são um mecanismo de empacotamento para a abstração de modelos de domínio complexos.

As classes herdam de suas superclasses atributos, estrutura de partes, relacionamentos e comportamento. Pode-se resumir nos seguintes itens os produtos da atividade de modelagem:

- Um esquema conceitual (um conjunto de esquemas conceituais, sendo um para cada subsistema).
- Um conjunto de definições de subsistemas, classes, objetos e relacionamentos (utilizando Cartões de Subsistemas, Classes, Objetos e Relacionamentos).

Adaptado de Pompermaier (2002), tem-se o exemplo mostrado na Figura 4.1. Trata-se do projeto conceitual de um software Guia de profissões, na forma de curso que possui várias matérias e é apresentado em um *CD-ROM*. Os quadrados no círculo representam as telas navegáveis no programa.

**FIGURA 4.1 – EXEMPLO DE PROJETO CONCEITUAL**



## 4.2 PROJETO NAVEGACIONAL

A maior contribuição da OOHDM está, justamente, na modelagem navegacional. Na OOHDM, uma aplicação é vista como uma visão navegacional do projeto conceitual. Esta visão é construída durante a modelagem navegacional levando-se em conta os tipos de usuários aos quais a aplicação se destina e o conjunto de tarefas que deverão desempenhar ao utilizá-lo.

De acordo com Schwabe (1996a), ao projetar a estrutura de navegação de um aplicativo de hipermídia, vários aspectos serão considerados, tais como:

- o Quais objetos serão navegados e que atributos possuem? Quais os relacionamentos entre estes objetos e aqueles definidos no esquema

conceitual? Isto será feito pela definição de nós e elos como visões orientadas a objetos das classes e relacionamentos conceituais.

- o Qual é a estrutura subjacente de navegação? Em que contextos o usuário irá navegar?
- o É preciso decidir se os objetos navegados poderão ter aparências diferentes de acordo com o contexto em que são visitados e especificar tais diferenças claramente.
- o Quais as estruturas de elo existentes entre os objetos que serão navegados (elos, caminhos, índices, roteiros guiados, etc.)?
- o Como procede a navegação quando o usuário “pula” de um assunto a outro, isto é, qual o efeito da navegação nos objetos de origem e alvo e em outros objetos relacionados?

Os contextos navegacionais são um conjunto de nós, elos e outros contextos navegacionais (aninhados) que auxiliam na organização dos objetos navegacionais, fornecendo espaços de navegação consistentes e, deste modo, diminuindo as chances de o usuário “perder-se no hiperespaço”.

Os nós podem ser atômicos ou compostos e são descritos através de atributos e âncoras para elos no caso de serem atômicos, e como um conjunto de nós componentes no caso de serem compostos. Os contextos navegacionais definem um conjunto de classes “decoradoras” que adaptam cada nó no contexto para as estruturas âncoras-atributos necessárias ao contexto.

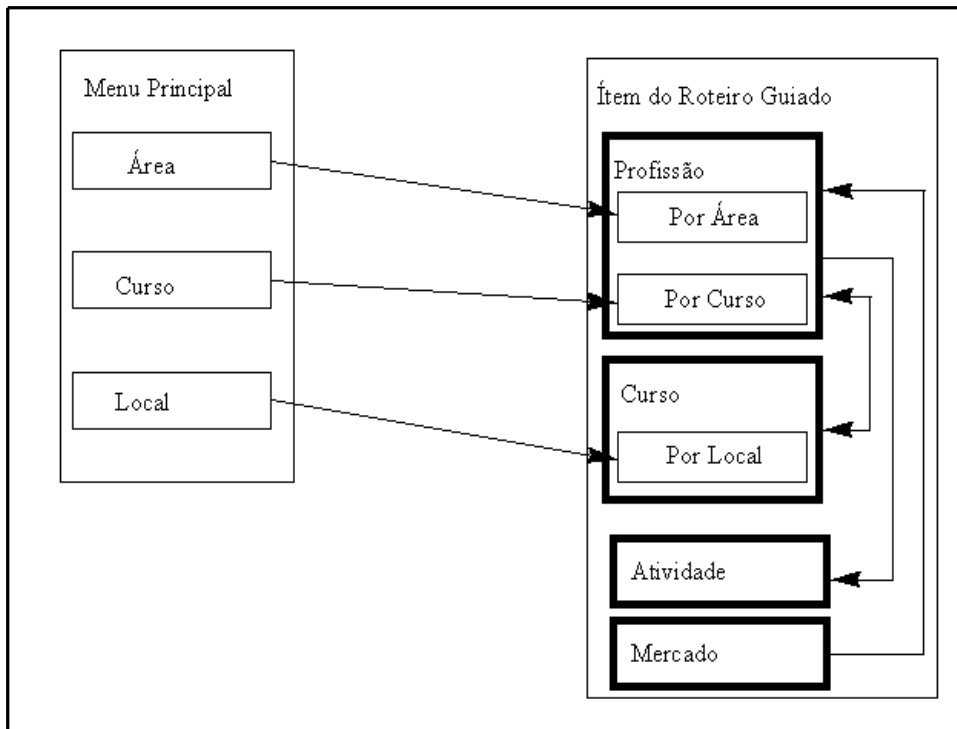
Os elos fazem a ligação entre os objetos navegacionais e podem ser de um-a-um ou um-a-muitos. O resultado da travessia de um elo é expresso pela definição das semânticas navegacionais, como resultado do comportamento do elo. As estruturas de acesso (tais como os índices ou roteiros guiados) são também definidas como classes e apresentam modos alternativos para a navegação no aplicativo hipermídia.

As classes e contextos navegacionais definem a estrutura estática da aplicação hipermídia. Contudo, é necessário também, especificar os aspectos dinâmicos da navegação.

Em OOADM, isto é feito pelo uso de Diagramas de Navegação, um modelo baseado em máquinas de estado no qual são mostradas transformações no estado do espaço navegacional.

Um exemplo de projeto navegacional é mostrado na figura 4.2, adaptado do guia de profissões de Pompermaier (2002).

**FIGURA 4.2 – EXEMPLO DE PROJETO NAVEGACIONAL**



### 4.3 PROJETO DA INTERFACE ABSTRATA

Para especificar o projeto da interface abstrata, é necessário definir metáforas de interface e descrever suas propriedades estáticas e dinâmicas, assim como seus relacionamentos com o projeto navegacional de uma forma independente de implementação. Ainda de acordo com Schwabe (1996a), é preciso especificar:

- a aparência de cada objeto navegacional que será percebido pelo usuário, isto é, a representação de seus atributos (incluindo as âncoras);



- o outros objetos de interface para oferecer as diversas funções da aplicação, como barras de menus, botões de controle e menus;
- o os relacionamentos entre os objetos de interface e navegacionais, tais como o modo com que um evento externo afetará a navegação;
- o as transformações de interface ocorridas pelo efeito da navegação ou de eventos externos no comportamento de diferentes objetos de interface;
- o sincronização de alguns objetos de interface, especialmente quando há meios dinâmicos, como áudio e vídeo, envolvidos.

Em OOHDm usa-se *Abstract Data Views* (ADV) para especificar o projeto da interface abstrata. O modelo de projeto ADV foi criado originalmente para especificar clara e formalmente a separação entre a interface do usuário e os componentes de um sistema de software, e para oferecer um método de projeto independente de implementação, gerando graus mais altos de reuso de componentes de projeto e de interface. Os ADV são geralmente usados para representar interfaces entre duas mídias diferentes. Um ADV, quando usado em um projeto de aplicação hipermídia, pode ser visto como um objeto de interface contendo um conjunto de atributos que definem suas propriedades de percepção e o conjunto de eventos com os quais pode lidar, como os eventos gerados pelo usuário. Valores de atributos podem ser definidos como constantes, determinando, então, um estilo específico de aparência como posição, cor ou som.

Os **Diagramas de Configuração** são úteis na expressão de padrões de comunicação entre objetos, em termos de serviços oferecidos e requeridos. No contexto de OOHDm, concentra-se o interesse no modo como o usuário irá interagir com a aplicação hipermídia e, especialmente, em quais objetos de interface causarão a navegação.

Os **Diagramas ADV** suportam o aninhamento de estados e ADV, permitindo a expressão da associação entre eventos externos e ADV. Os diagramas ADV são a expressão interfacial dos diagramas de navegação, isto é, expressam transformações no nível da interface de usuário e seu impacto nos objetos navegacionais (Schwabe, 1996a).

Um diagrama ADV é composto por ADV, estados, atributos e transições. A aninhagem de ADV permite mostrar a estrutura de agregação dos objetos de interface.

Conforme Schwabe (1996a), a definição do modelo de interface de uma aplicação hipermídia com OOHDH implica em:

- definir a estrutura geral da interface da aplicação;
- definir ADV para os nós, índices, etc.;
- definir, em cada nó, objetos de interface apropriados para atributos, âncoras, etc.
- especificar o diagrama de configuração, mostrando relacionamentos estáticos entre ADV e ADO (sendo estes objetos navegacionais);
- mostrar relacionamentos estáticos entre os componentes de um ADV. Deve ser possível expressar todas as restrições estruturais existentes entre os objetos de interface.;
- especificar, para cada ADV significativo, o diagrama ADV, mostrando a dinâmica da aplicação hipermídia.

Deve estar claro que, apesar de ser independente de implementação, a especificação da interface abstrata deve considerar certos aspectos do ambiente de implementação alvo, para que a especificação seja realista.

Para registrar o produto do Projeto de Interface Abstrata há cartões que registram o resultado da atividade de projeto de interface.

Cartões de ADV Estáticos, representando os relacionamentos estáticos entre ADV e objetos navegacionais e os Cartões de ADV Dinâmicos, mostram os efeitos da interação entre usuário e mídia na interface e seu impacto na navegação.

## 4.4 IMPLEMENTAÇÃO

A implementação de uma aplicação hipermídia de modo que esta resulte usável não é tarefa simples. Muitas questões técnicas e não-técnicas devem ser resolvidas. Uma vez que o ambiente de implementação tenha sido escolhido, o projeto deve ser mapeado para artefatos de implementação e todos os componentes hipermídia têm de ser instanciados.

Basicamente, é necessário definir os objetos de interface de acordo com a especificação da interface abstrata, implementar transformações da forma como foram definidas nos diagramas ADV e fornecer suporte para a navegação através da rede hipermídia.

A informação fornecida pela especificação baseada em ADV e, em especial, a estrutura aninhada dos ADV, oferece uma indicação sobre quais os objetos de interface que se precisa definir.

De acordo com Schwabe (1996b), os aspectos dinâmicos da interface lidam tanto com transformações de interface dentro de um ADV (por exemplo, mostrando/ocultando algum atributo), como com transformações de interface envolvendo navegação. Em ambos os casos, a natureza orientada a eventos dos diagramas ADV e o poder expressivo de seu modelo de transição de estados permite a implementação destas transformações em um estilo simples.

Em aplicações hipermídia, a navegação ocorre porque o usuário seleciona uma âncora em um nó. É possível implementar comportamento navegacional mais sofisticado, como, por exemplo, navegação baseada em tempo, ou baseada em mídia (quando um evento determinado ocorre na mesma mídia, pode-se proceder de modo a seguir um elo). Ambos os tipos de comportamento navegacional são especificados através de diagramas ADV, já que as transições de estado e o evento gerador de cada mudança estão documentados na especificação da transição.

Para manter a informação de rastreabilidade acumulada durante o processo do projeto, deve-se documentar, também, as decisões de implementação. Apesar de esta documentação poder ser facilmente automatizada e introduzida no ambiente de implementação, pode-se gerar cartões simples ao implementar a aplicação.

## 5 ESPECIFICAÇÃO

A especificação do protótipo foi feita de acordo com a metodologia OOADM. Assim, para ilustrar o projeto conceitual, é mostrado o diagrama de classes na Figura 5.1.

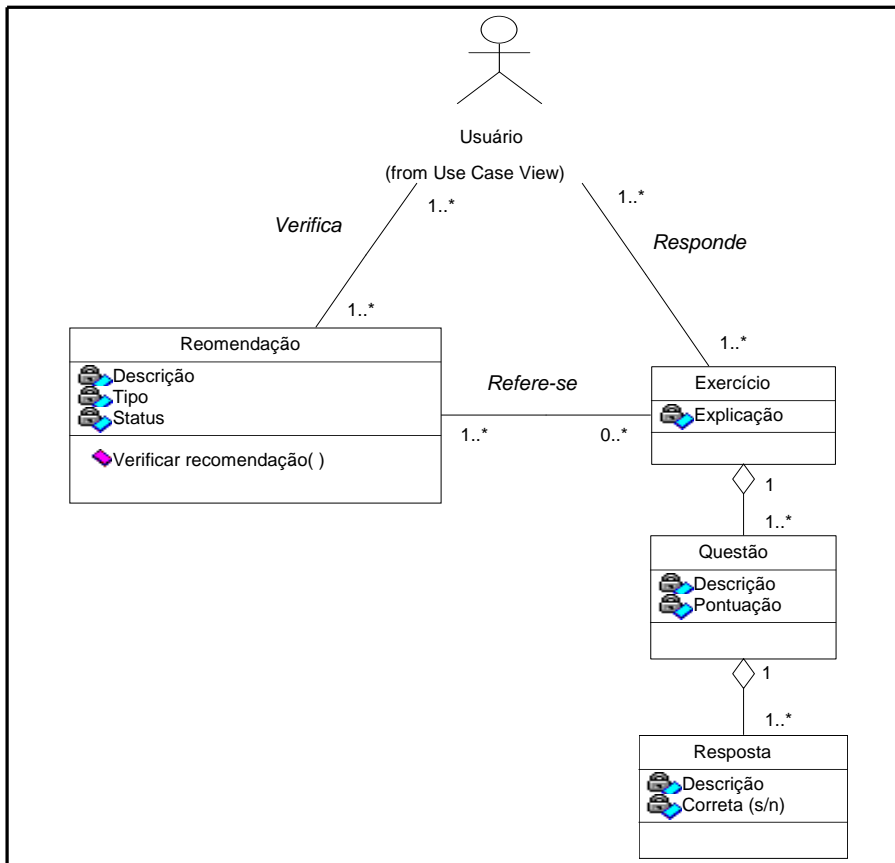
O Projeto Navegacional é mostrado na Figura 5.2. Convém explicar que as setas que saem do menu principal, indicam o fluxo de navegabilidade entre as telas. Já na seta que parte da opção sair, no retângulo da direita, indica que quando esta opção for selecionada, todas as outras ficarão inativas, pois o programa será encerrado, por isso ela está tracejada.

Para complementar o projeto navegacional, é exposto ainda DHF - Diagrama Hierárquico Funcional, com conceitos extraídos de Martin (1991) e também um fluxograma completo do sistema, abrangendo todas as telas existentes no tutorial. Assim, Na Figura 5.3 tem-se o DHF das principais funções do protótipo, sendo sua seqüência mostrada na Figura 5.4.

No caso do fluxograma, devido a sua extensão, foi necessário dividir em três partes: na Figura 5.5, com o início e as opções básicas do tutorial, na Figura 5.6, com os itens Disposição, Formas e Cores e, por fim, na Figura 5.7, explorando os tópicos Texto, Saídas e as opções do menu tópicos.

O Projeto da Interface Abstrata é mostrado na Figura 5.8. Nele são explorados os dois exercícios do tutorial, pois são os que mais se encaixam a este tópico da metodologia.

FIGURA 5.1 – PROJETO CONCEITUAL



No projeto mostrado acima, a classe *Recomendação* refere-se às recomendações para a correta construção de interfaces. Estas recomendações têm uma descrição, um tipo (forma, texto, cores, etc.) e ainda um status (se já foi consultada ou não).

A classe *Exercício* possui um atributo que é a explicação de como resolve-lo. Agregado a esta classe, tem-se a classe *Questão*, com atributos descrição e pontuação e, por fim, agregado à classe *Questão*, aparece a classe *Resposta*, também com descrição e com um atributo indicando se ela está correta ou não.

O “ator” *Usuário*, como o nome indica, representa o usuário do sistema em si, com relacionamentos que mostram que ele verifica as recomendações e responde aos exercícios. Convém, porém, ressaltar que este ator não faz parte da implementação e foi colocado no diagrama para melhor representar o projeto conceitual.

FIGURA 5.2 – PROJETO NAVEGACIONAL

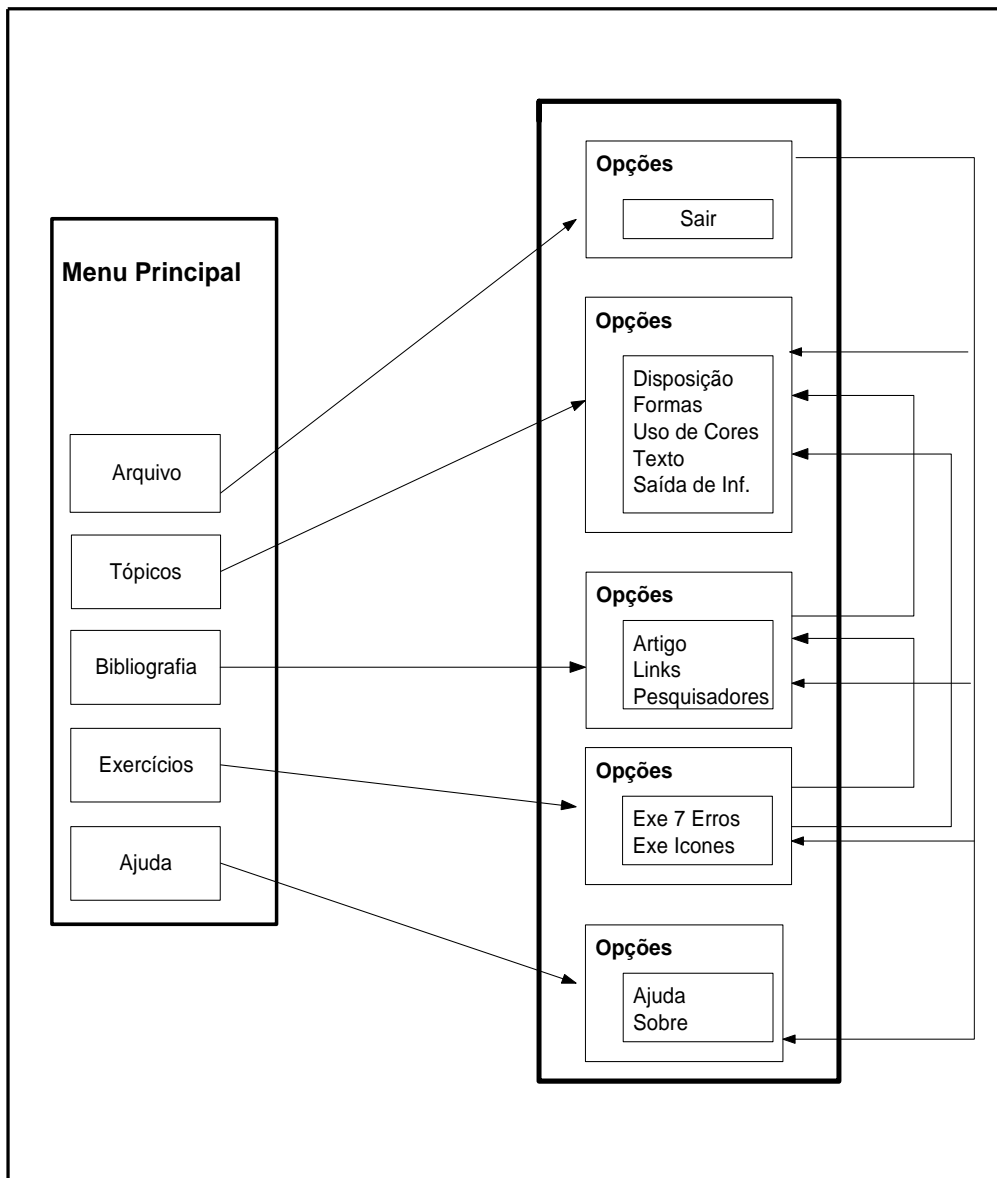


FIGURA 5.3 – DIAGRAMA HIERÁRQUICO FUNCIONAL 1

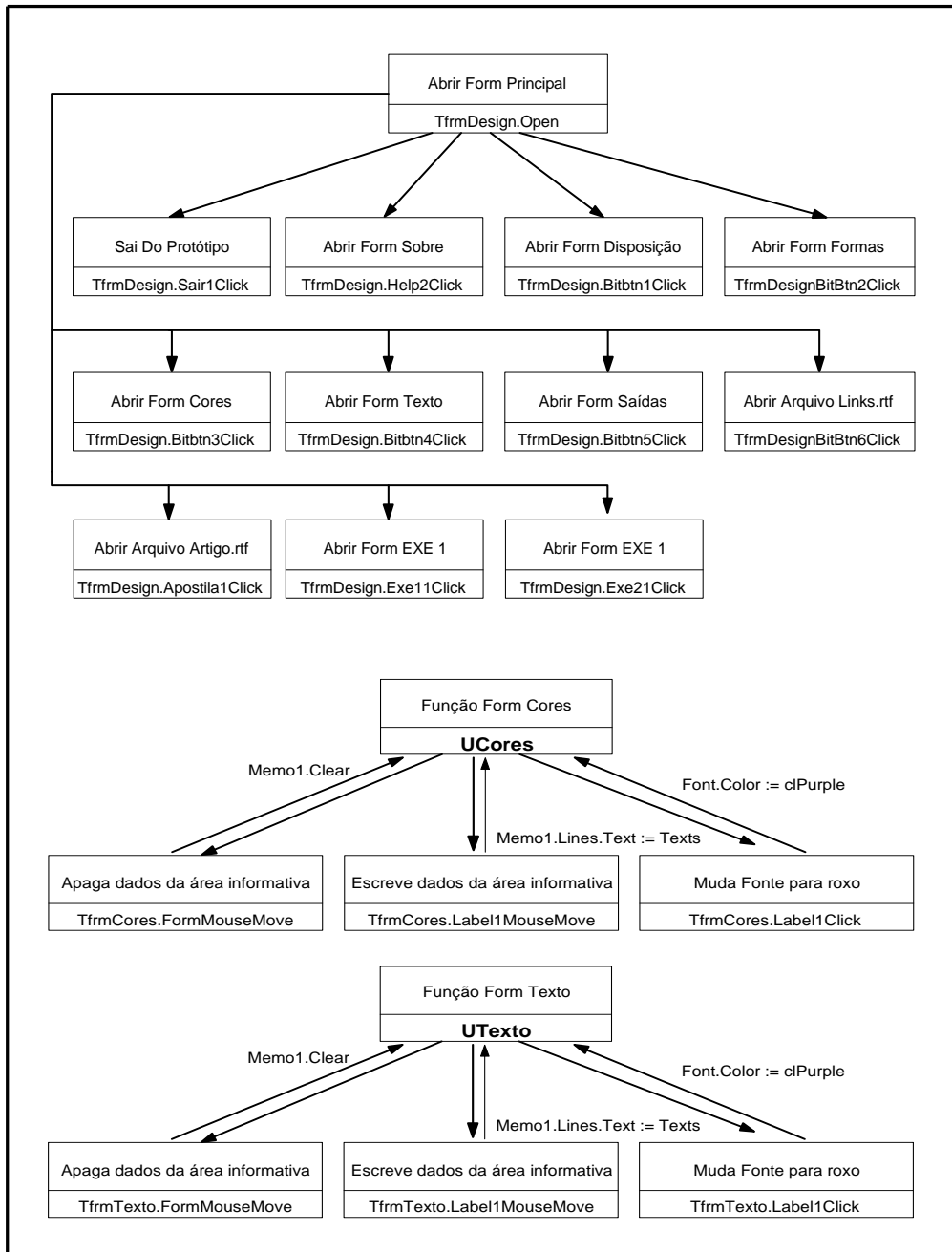


FIGURA 5.4 – DIAGRAMA HIERÁRQUICO FUNCIONAL 2

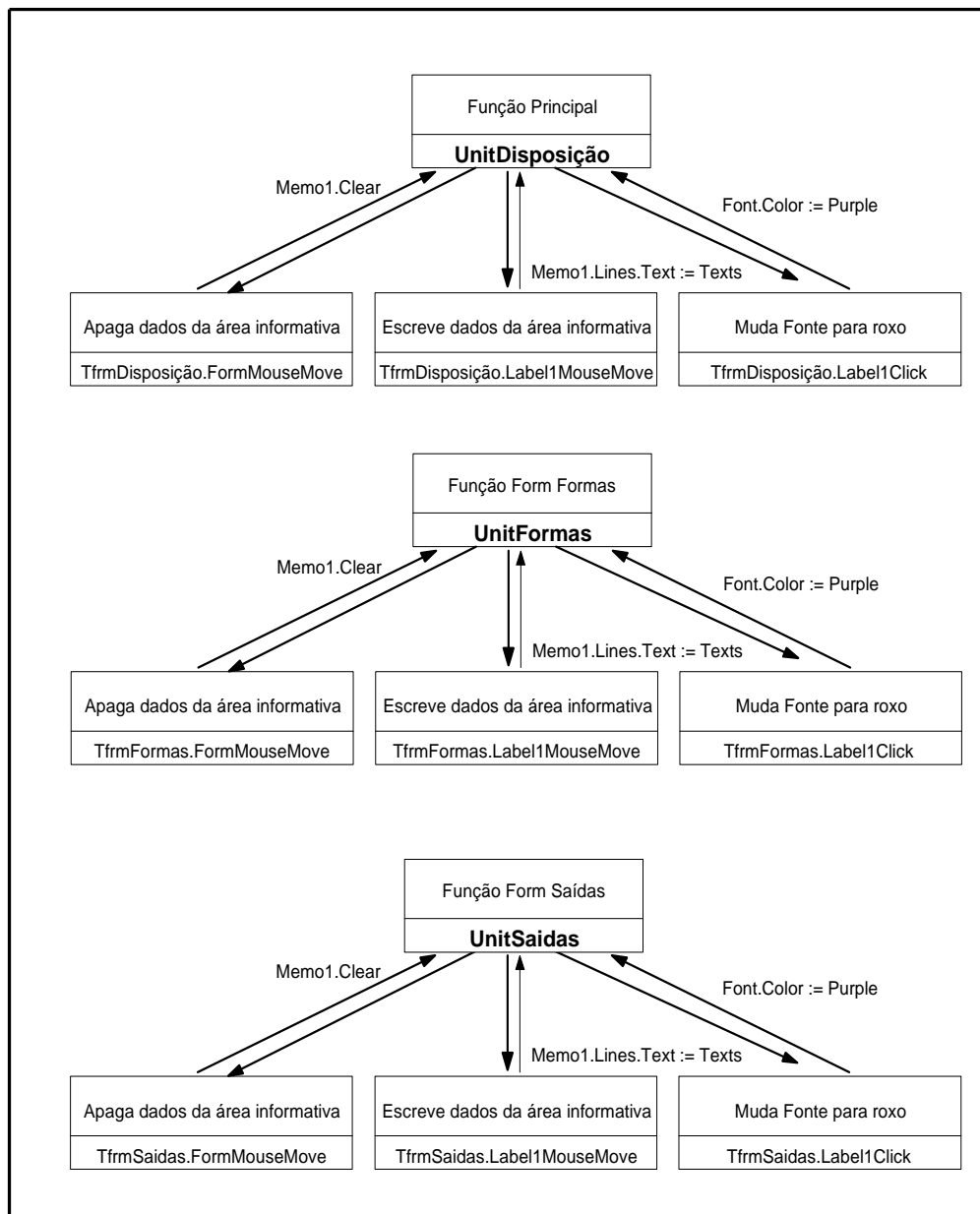
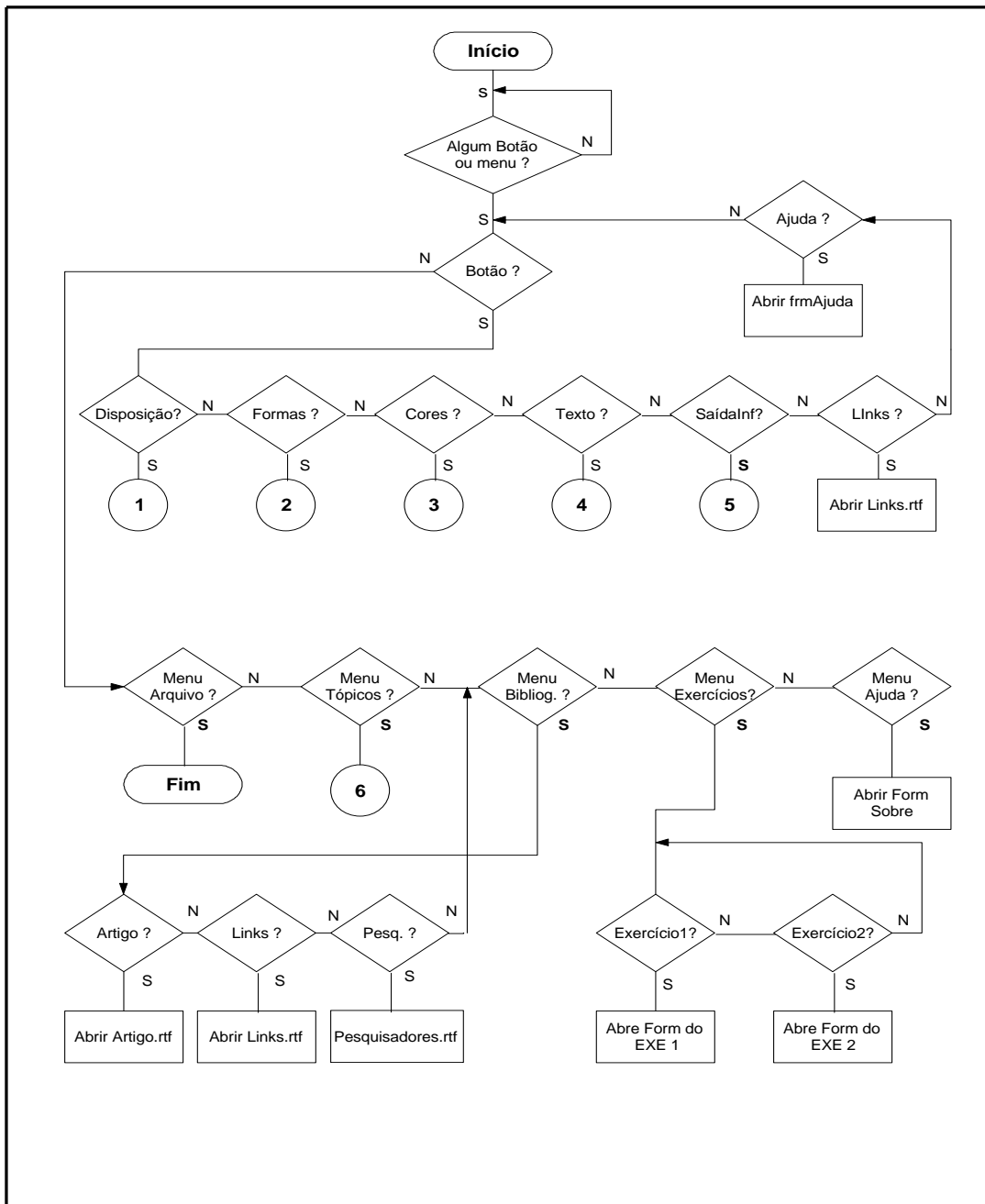
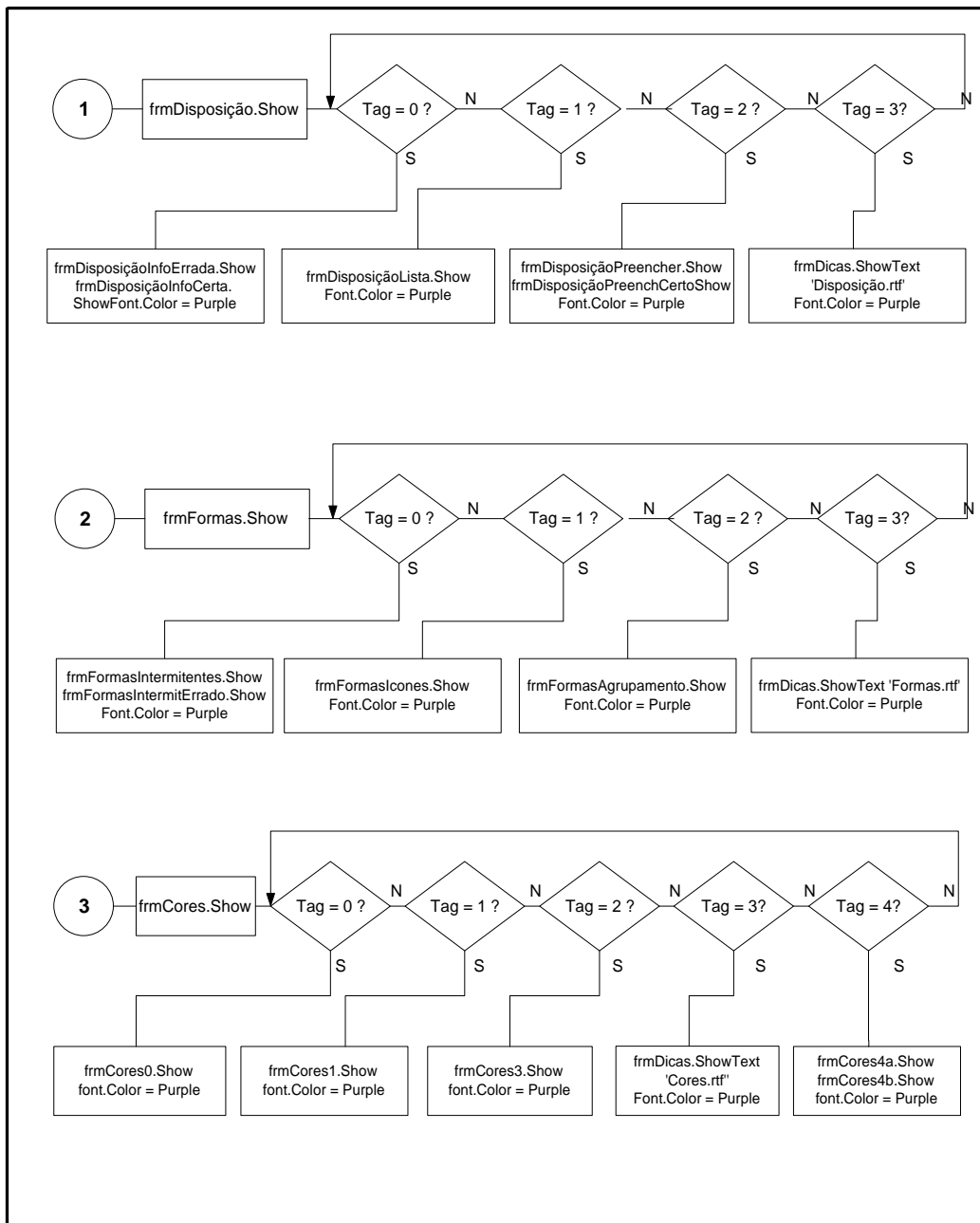




FIGURA 5.5 – FLUXOGRAMA PRINCIPAL



**FIGURA 5.6 – FLUXOGRAMA DISPOSIÇÃO / FORMAS / CORES**



**FIGURA 5.7 – FLUXOGRAMA TEXTO / SAÍDAS / MENU**

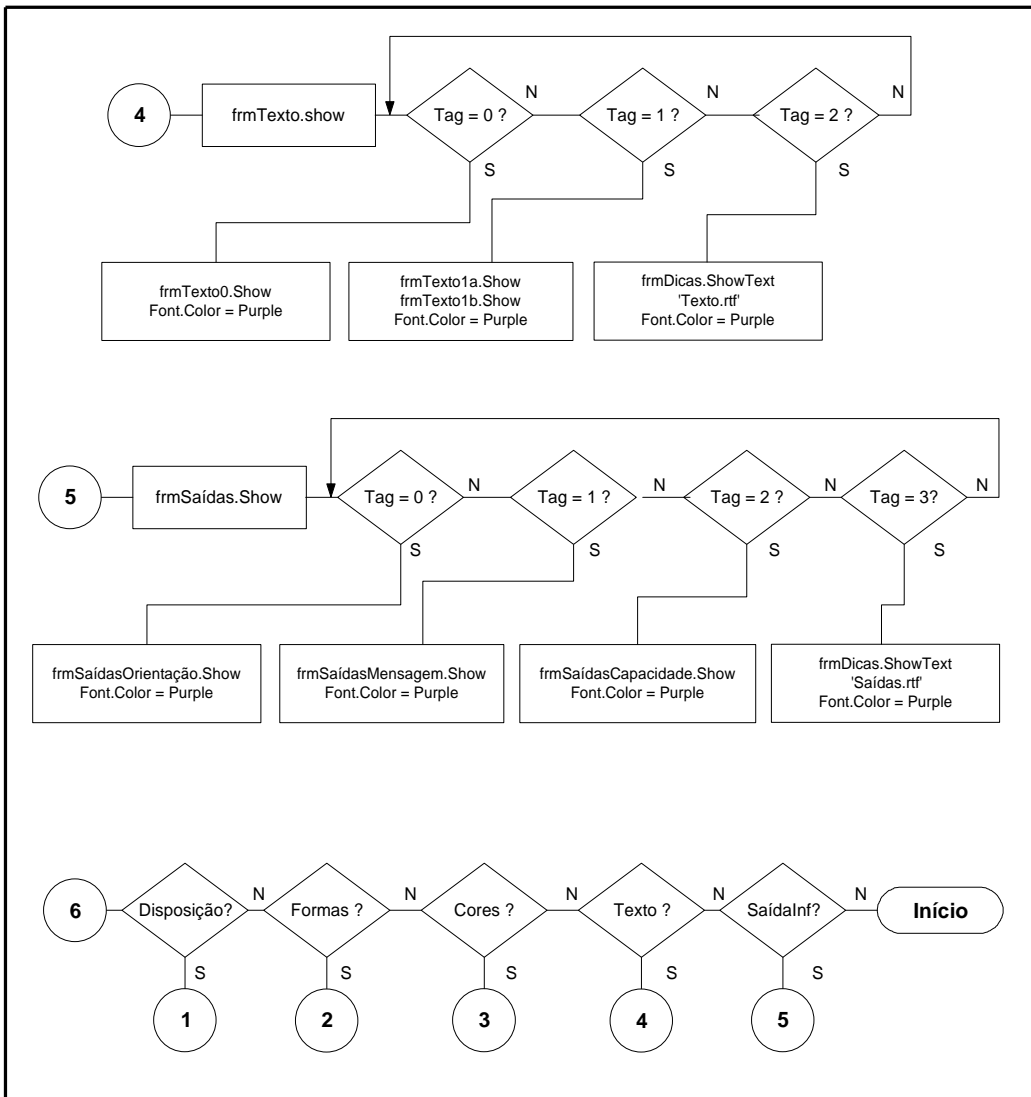
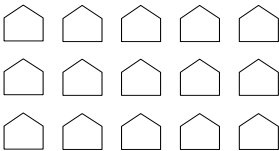


FIGURA 5.8 – PROJETO DA INTERFACE ABSTRATA

<b>Exercício 1 - Jogo dos sete erros</b>	
Enunciado do exercício:	
<input type="checkbox"/> Alternativa 1 <input type="checkbox"/> Alternativa 2 <input type="checkbox"/> Alternativa 3 <input type="checkbox"/> Alternativa 4 <input type="checkbox"/> Alternativa 5 <input type="checkbox"/> Alternativa 6 <input type="checkbox"/> Alternativa 7 <input type="checkbox"/> Alternativa 8 <input type="checkbox"/> Alternativa 9 <input type="checkbox"/> Alternativa 10 <input type="checkbox"/> Alternativa 11 <input type="checkbox"/> Alternativa 12 <input type="checkbox"/> Alternativa 13 <input type="checkbox"/> Alternativa 14	<b>Verificar Pontuação</b>  <b>Nota:</b> <input type="text"/> <b>Acertos:</b> <input type="text"/>
<b>Exercício 2 - Teste escolha dos ícones</b>	
Enunciado do exercício:	
<b>Ícones:</b>	
Função 1 Função 2 Função 3 Função 4 Função 5 Função 6 Função 7 Função 8	<b>Verificar Pontuação</b>  <b>Nota:</b> <input type="text"/> <b>Acertos:</b> <input type="text"/>

## 6 IMPLEMENTAÇÃO

Neste capítulo são descritas, nesta ordem, a técnicas e ferramentas utilizadas, mencionando a metodologia, a linguagem e o ambiente de programação. A seguir, a operacionalidade da implementação, apresentando as principais telas do programa. Por fim, é exposta a análise do protótipo, mostrando resultados obtidos nos testes efetuados pelos usuários.

### 6.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

Como já foi mencionado, a metodologia de desenvolvimento utilizada foi a OOHDM, sendo que para a construção e implementação do tutorial propriamente dito, utilizou-se a linguagem *Object Pascal*, na ferramenta Borland Delphi 5.

Convém salientar que, apesar de ter sido especificado um diagrama de classes no projeto conceitual, a implementação propriamente dita não foi construída sob o paradigma objeto orientado. No caso do tutorial desenvolvido, foi utilizada programação estruturada.

### 6.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO

Para demonstrar a funcionalidade da implementação, serão expostas as principais telas do tutorial, a fim de comprovar a eficácia do mesmo.

A figura 6.1 mostra a tela de abertura, na qual o usuário poderá escolher dentre os tópicos a serem estudados: Disposição da informação, Formas, Uso de Cores, Texto e Saída da Informação, podendo ainda clicar na aba exercícios.

FIGURA 6.1 – TELA INICIAL



A medida que o usuário clica nos botões, telas respectivas aos assuntos vão se abrindo, de forma que ele vai tomando conhecimento dos diversos conceitos e técnicas para a construção correta de interfaces gráficas com o usuário.

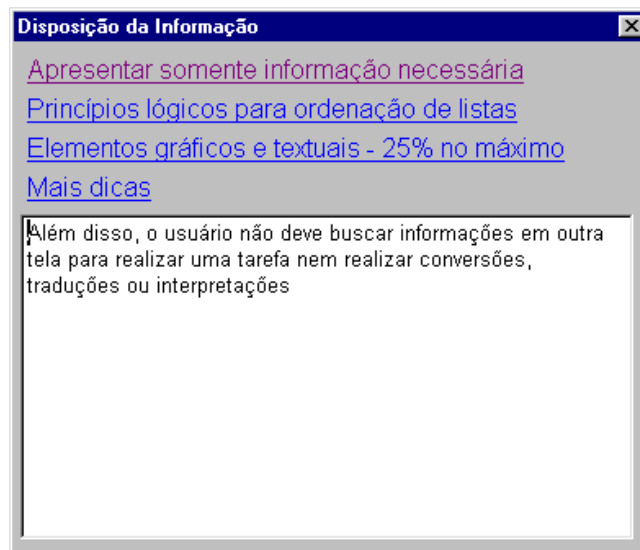
Ao clicar no primeiro tópico, Disposição da Informação, a janela mostrada na figura 6.2 aparecerá. Nesta tela, existem quatro frases, sendo que cada linha sublinhada é um “link” para uma outra janela. As janelas que vêm após o clique em uma destas linhas, mostram informações mais específicas sobre o determinado tópico e ainda mostram exemplos práticos da recomendação abordada.

Formatado

**FIGURA 6.2 - DISPOSIÇÃO DA INFORMAÇÃO**

Formatado

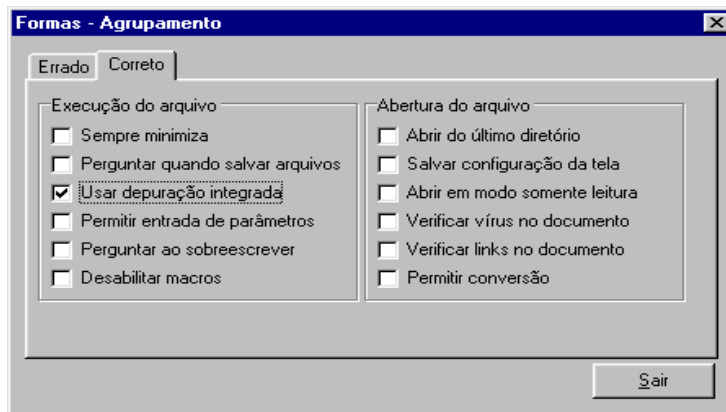
Formatado



Por conveniência, serão mostradas a seguir somente algumas telas do tutorial, haja vista a existência de um grande número delas no trabalho.

No tópico formas, é mostrado uma maneira correta de agrupar ícones correlatos, conforme Figura 6.3.

FIGURA 6.3 - FORMA - AGRUPAMENTO



Com relação ao uso de cores, foi selecionado um exemplo extraído do tutorial. Nele, pode-se observar, de acordo com a seqüência (Figura 6.4, Figura 6.5 e Figura 6.6), a forma de como a cor pode ser usada para reforçar os fatos. À medida que a temperatura vai aumentando, a cor na caixa muda de azul para branco e, por fim, quando a temperatura é máxima, a caixa é pintada de vermelho.

FIGURA 6.4 - ERRADO/CERTO - FRIA

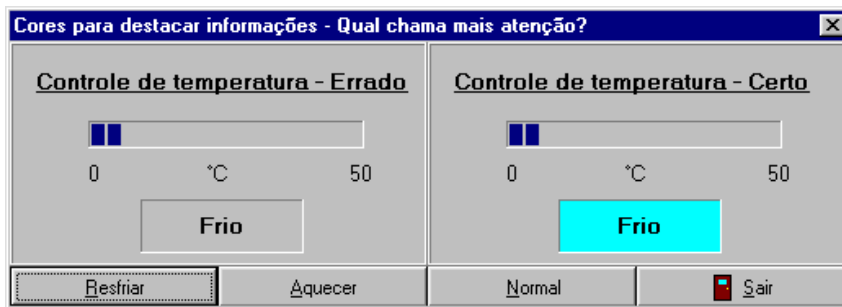


FIGURA 6.5 - ERRADO/CERTO - NORMAL

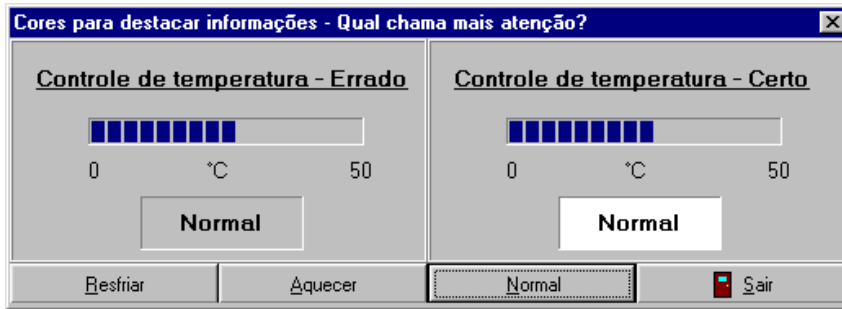
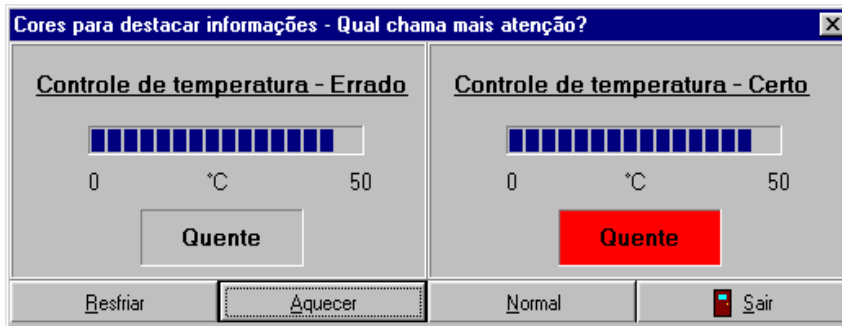


FIGURA 6.6 - ERRADO/CERTO - NORMAL



No quadro 6.1, é ilustrado a parte do código fonte responsável pela geração das telas apresentadas na página anterior.



### QUADRO 6.1 – FONTE DOS FORMS DA EXPRESSÃO DA COR

```

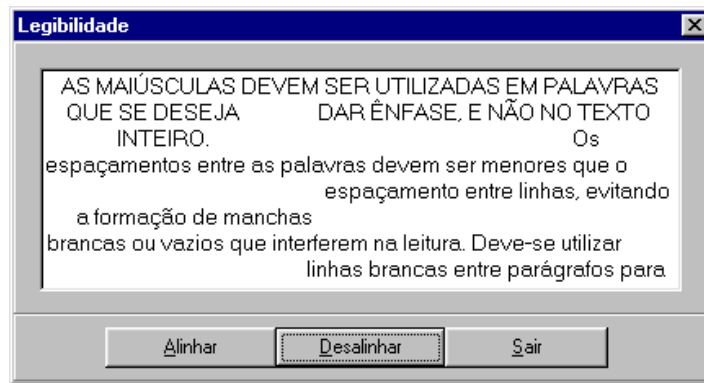
// Esta procedure simula o efeito da cor para mostrar
// a variação de temperatura
procedure TfrmCores1.btnAumentaClick(Sender: TObject);
begin
  if ProgressBar1.Position <= 10 then
    begin
      ProgressBar1.Position:= ProgressBar1.Position + 1;
      ProgressBar2.Position:= ProgressBar2.Position + 1;
      case ProgressBar1.Position of
        0..3: begin //Com progress bar até 3, nome=frio e cor azul
          pnStatusCerto.Caption:= 'Frio';
          pnStatusErrado.Caption:= 'Frio';
          pnStatusCerto.Color:= clAqua;
        end;
        4..6: begin //Com progress bar até 6, nome=normale cor branca
          pnStatusCerto.Caption:= 'Normal';
          pnStatusErrado.Caption:= 'Normal';
          pnStatusCerto.color:= clWhite;
        end;
        7..10: begin //Com progress bar até 10, nome=quente ,cor verm.
          pnStatusCerto.Caption:= 'Quente';
          pnStatusErrado.Caption:= 'Quente';
          pnStatusCerto.Color:= clRed;
        end;
      end;
    end;
  end;
end;

procedure TfrmCores1.btnDiminuiClick(Sender: TObject);
begin
  if ProgressBar1.Position > 0 then
    begin
      ProgressBar1.Position:= ProgressBar1.Position - 1;
      ProgressBar2.Position:= ProgressBar2.Position - 1;
      case ProgressBar1.Position of
        0..3: begin // Com progress bar até 3, nome=frio e cor azul
          pnStatusCerto.Caption:= 'Frio';
          pnStatusErrado.Caption:= 'Frio';
          pnStatusCerto.Color:= clAqua;
        end;
        4..6: begin //Com progress bar até 6, nome=normal, cor branca
          pnStatusCerto.Caption:= 'Normal';
          pnStatusErrado.Caption:= 'Normal';
          pnStatusCerto.Color:= clWhite;
        end;
        7..10: begin //Com progress bar até 10, nome=normal cor verm
          pnStatusCerto.Caption:= 'Quente';
          pnStatusErrado.Caption:= 'Quente';
          pnStatusCerto.Color:= clRed;
        end;
      end;
    end;
  end;
end;

```

Com relação ao texto, um dos exemplos do tutorial, refere-se ao desalinhamento encontrado em alguns softwares, conforme mostrado na Figura 6.7.

**FIGURA 6.7 - TEXTO DESALINHADO.**



A Figura 6.8 apresenta um exemplo de recomendação para a saída de informações, onde, em sua forma correta, existe uma barra de progresso e ainda uma ampulheta para dar a impressão de tempo decorrido.

**FIGURA 6.8 - ORIENTAÇÃO AO USUÁRIO**



O código utilizado para criar o exemplo mostrado na Figura 6.8 pode ser observado no Quadro 6.2. Nesse quadro é mostrado o código relativo ao processamento sem orientação e processamento com movimento real (barra sem ampulheta). No Quadro 6.3, são ilustradas as *procedures* de processamento com movimento só ilusório e real/ilusório.

**QUADRO 6.2 – FONTE SEM ORIENTAÇÃO E MOVIMENTO REAL**

```
// Processa sem orientação
procedure TfrmSaidasOrientacao.Button3Click(Sender: TObject);
```

```

begin
  Timer1.Enabled := True; {Timer=5000}
  Label1.Caption := 'Processando...'
end;
procedure TfrmSaidasOrientacao.Timer1Timer(Sender: TObject); //fim do timer
begin
  Timer1.Enabled := False; {Timer=5000}
  Label1.Caption := 'Processado';
end;
procedure TfrmSaidasOrientacao.Button4Click(Sender: TObject); //cancela
begin
  Timer1.Enabled := False;
  Label1.Caption := 'Processo cancelado';
end;

// Processa movimento real (barra sem ampulheta)
procedure TfrmSaidasOrientacao.Button1Click(Sender: TObject);
begin
  frmSaidasOrientacao.Label3.Visible := true;
  frmSaidasOrientacao.Label3.Caption := 'Processando...';
  Timer2.Enabled := True; {Timer=500}
  ProgressBar1.Position := 0;
end;
procedure TfrmSaidasOrientacao.Button2Click(Sender: TObject); //cancela
begin
  Timer2.Enabled := False;
end;
procedure TfrmSaidasOrientacao.Timer2Timer(Sender: TObject); //fim do timer
begin
  ProgressBar1.Position := ProgressBar1.Position + 5;
  if ProgressBar1.Position >= 100 then
  begin
    Timer2.Enabled := False;
    frmSaidasOrientacao.Label3.Visible := true;
    frmSaidasOrientacao.Label3.Caption := 'Processo Concluído';
  end;
end;
end;

```

---

### QUADRO 6.3 – FONTE MOVIMENTO ILUSÓRIO E REAL/ILUSÓRIO

```

// Processa moviemento ilusório (só ampulheta)
procedure TfrmSaidasOrientacao.Button5Click(Sender: TObject);
begin
  Timer3.Enabled := True; {Timer=5000}
  Label2.Caption := 'Processando...';
  Screen.Cursor := crHourglass; {ampulheta}
end;
procedure TfrmSaidasOrientacao.Button6Click(Sender: TObject);
begin
  Timer3.Enabled := False;
  Label2.Caption := 'Processo cancelado';
  Screen.Cursor := crDefault;
end;

```

```

end;
procedure TfrmSaidasOrientacao.Timer3Timer(Sender: TObject); //fim do timer
begin
    Timer3.Enabled := False;
    Label2.Caption := 'Processado';
    Screen.Cursor := crDefault;
end;

// Processa moviento real e ilusório (barra e amplulheta)

procedure TfrmSaidasOrientacao.Button7Click(Sender: TObject);
begin
    frmSaidasOrientacao.Label4.Visible := true;
    frmSaidasOrientacao.Label4.Caption := 'Processando...';
    Timer4.Enabled := True; {Timer=5000}
    ProgressBar3.Position := 0;
    Screen.Cursor := crHourglass; {amplulheta}
end;
procedure TfrmSaidasOrientacao.Button8Click(Sender: TObject);
begin
    Timer4.Enabled := False;
    Screen.Cursor := crDefault;
end;
procedure TfrmSaidasOrientacao.Timer4Timer(Sender: TObject); //fim do
timer
begin
    ProgressBar3.Position := ProgressBar3.Position + 5;
    if ProgressBar3.Position >= 100 then
    begin
        frmSaidasOrientacao.Label4.Visible := true;
        frmSaidasOrientacao.Label4.Caption := 'Processo Concluído';
        Timer4.Enabled := False;
        Screen.Cursor := crDefault;
    end;
end;
end.

```

---

Na opção **Links**, encontra-se uma lista com *links* de *sites* relacionados ao assunto “Design de Interfaces”, para que o usuário possa aprofundar os seus conhecimentos fazendo uma pesquisa na internet. Esta janela é mostrada na Figura 6.9.

**FIGURA 6.9 - LINKS**



Por fim, clicando na aba Exercícios, o usuário pode escolher entre o exercício 1, no qual são testados seus conhecimentos no âmbito geral do *design* de interfaces. Neste exercício, o usuário interage com uma tela na qual deve observar toda a estrutura da mesma, verificando possíveis erros que serão marcados por ele na tela seguinte, numa espécie de jogo dos 7 erros. A nota será impressa de acordo com a quantidade de erros que ele identificar no *form* apresentado. A figura 6.10 ilustra o exercício 1.

O usuário pode ainda escolher o exercício 2, que trata mais especificamente da correta escolha de ícones para a interface a ser criada. Neste exercício, é preciso que o usuário marque um “*check box*” e, em seguida, clique em um ícone que deve representar a função escolhida. Ao final, deve clicar em “Verificar Pontuação” a fim de conhecer a sua nota. O exercício 2 é ilustrado na figura 6.11.

**FIGURA 6.10 - EXERCÍCIO JOGO DOS 7 ERROS**

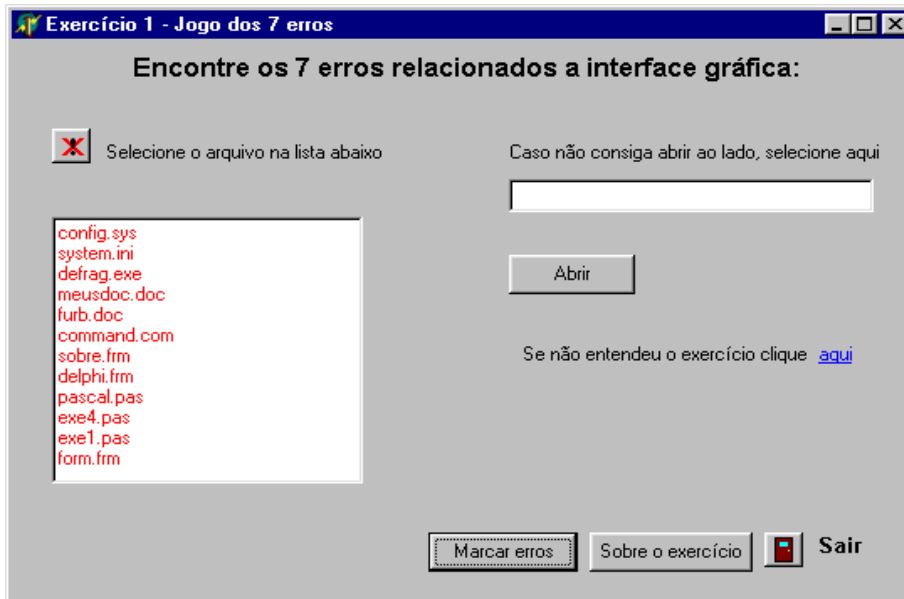
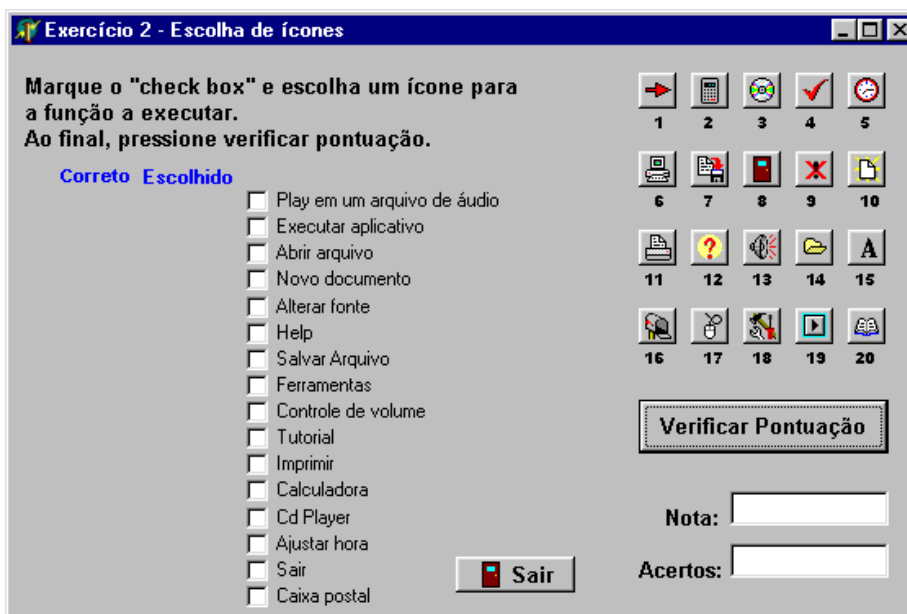


FIGURA 6.11 - EXERCÍCIO ESCOLHA DE ÍCONES



### 6.3 ANÁLISE E VALIDAÇÃO DO TUTORIAL

Para validar o tutorial desenvolvido para este trabalho, foi solicitado a alguns programadores que utilizassem o software, testando e procurando erros a fim de verificar a sua eficiência no que diz respeito às recomendações e até mesmo com relação à disposição das telas do próprio protótipo.

Após esta utilização, os programadores responderam um questionário com perguntas relacionadas à funcionalidade do software. Estas perguntas e as possíveis respostas estão dispostas a seguir.

- 1) O assunto tratado no tutorial e do seu interesse profissional ?
- 2) As telas são dispostas de maneira que torna o aprendizado fácil ?
- 3) É possível fazer uma consulta mais aprofundada nos locais indicados no menu bibliografia ?
- 4) A disposição do conteúdo em cinco tópicos facilita a compreensão?
- 5) A exposição de exemplos práticos ajudou na fixação dos conteúdos mostrados no tutorial ?
- 6) Os exercícios auxiliam na fixação do conteúdo visto no tutorial ?
- 7) Este tutorial se mostra eficiente e vai contribuir para a revisão de seus paradigmas relacionados à construção de interfaces gráficas ?

Para o questionário acima, foram colocadas as seguintes possíveis respostas:

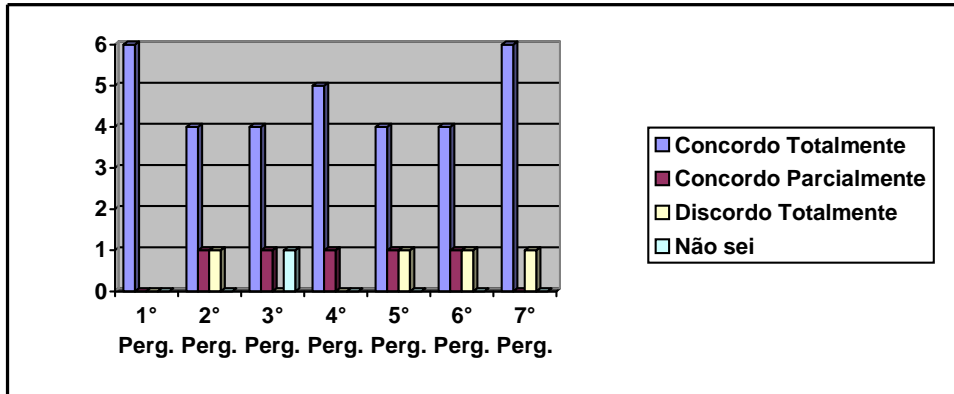
- a) concordo totalmente;
- b) concordo parcialmente;
- c) discordo totalmente;
- d) não sei.

Apesar da amostra ser estatisticamente pequena, o experimento serviu para fazer um breve teste de utilização do tutorial.

Com a apuração do resultado do questionário, constatou-se uma boa aceitação por parte dos entrevistados, pois a maioria mostrou-se satisfeito com a eficácia do tutorial,

respondendo quase que unanimemente às questões cinco e sete, conforme exposto no gráfico da figura 6.12, o que demonstra que o tutorial atingiu o resultado esperado.

**FIGURA 6.12 – RESULTADOS DO TESTE DE VALIDAÇÃO**



7

**Formatados:** Marcadores e numeração

**Excluído:** ¶



## 7 CONCLUSÕES

Por mais que as recomendações e dicas apresentadas possam apontar para uma padronização de interfaces gráficas, limitando a criatividade do desenvolvedor, acredita-se que, na verdade, os estudos estejam sendo direcionados para a padronização de uma linguagem gráfica. Assim como a linguagem simbólica usada pelos surdos-mudos ou como os símbolos de trânsito, largamente conhecidos pela comunidade que os usa, esta linguagem gráfica padrão facilita ao usuário interpretar de imediato informações e funções conhecidas em um programa que acaba de acessar. Isto sem dúvida é um diferencial na hora de escolher um software.

Além da eficiência com que um programa de computador executa suas funções, as interfaces gráficas com o usuário determinam a eficácia com que essas mesmas funções serão executadas.

O trabalho cumpriu com seu objetivo principal, pois foi desenvolvido o tutorial proposto com recomendações sobre os conceitos de desenvolvimento de interfaces gráficas.

Com relação aos objetivos específicos, o tutorial permite uma avaliação dos conhecimentos do usuário através da realização dos exercícios propostos. Está limitado ao número de dois exercícios. É possível fazer a pesquisa das recomendações ergonômicas. Estas recomendações encontram-se no tutorial indexadas por área.

Utilizando o tutorial, o usuário navega por telas que mostram a maneira correta e a errada de construção de interfaces, o que possibilita a visualização das diferenças entre ambas. O tutorial apresenta, ainda, exemplos práticos de recomendações ergonômicas. Do total das recomendações existentes, optou-se por incluir no tutorial apenas aquelas mais usuais e indicadas pela maioria dos autores.

Foi possível especificar o software utilizando a metodologia proposta nos objetivos específicos, assim, o resultado final ficou dentro do esperado.

## 7.1 EXTENSÕES

Como extensão para a implementação de novos trabalhos, sugere-se a ampliação dos exemplos e recomendações, bem como um incremento no número de exercícios propostos.

A título de pesquisa, sugere-se um estudo sobre outras formas de interação homem-computador, como as alternativas para deficientes visuais ou com alguma deficiência motora que impeça estas pessoas de operarem o mouse ou teclado.

Considera-se também produtivo realizar um estudo complementar questionado o nível de satisfação dos usuários de programas de computador com relação à interface gráfica dos programas que eles utilizam, colhendo ainda sugestões para a melhoria dos mesmos.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ARAGÃO, Alfredo Lanari de. **Modelagem com o método OOADM**, Campo Grande, mar. [2002]. Disponível em: <<http://www.ec.ucdb.br/~alanari/engsoft/eClassOOADM.doc>>. Acesso em: 12 ago. 2002.
- AURI, Marcelo. **Exemplo de modelagem utilizando OOADM**, São Paulo, mar. [2002?]. Disponível em: <<http://www.icmc.sc.usp.br/~auri/sce-5811/trabalho5.html>>. Acesso em: 12 ago. 2002.
- BARANAUSKAS, Ma. Cecília C. **Resumo da mesa redonda “IHC: Interação Humano-Computador ou Interação Humana através do Computador?”**, Rio de Janeiro, nov. [1998]. Disponível em: <<http://www.inf.puc-rio.br/~ihc98/>>. Acesso em: 01 ago. 2002.
- CAMILO JÚNIOR, Hélio da Conceição. **Tutorial de recomendações ergonômicas aplicadas à implementação de software**. 1995. 60 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- CANTU, Marco. **Dominando o DELPHI 5**. São Paulo: Makron Books, 2000.
- CARVALHO, José Oscar Fontanini de. **Interfaces para o deficiente visual**, São Paulo, nov. [1999?]. Disponível em: <<http://www.epub.org.br/informed/defic.htm>>. Acesso em: 08 ago. 2002.
- CORNELL, Gary. **DELPHI, segredos e soluções**. São Paulo: Makron Books, 1996.
- LABIUTIL. UFSC. **Ergonomia de Interfaces humano-computador**, Florianópolis, nov. [2000]. Disponível em: <<http://www.labiutil.inf.ufsc.br/apostila/Apost4-4.pdf>>. Acesso em: 08 ago. 2002.
- MARTIN, James. **Técnicas estruturadas e case**. São Paulo: Makron Books, 1991.
- MENDES, Daniela Santiago. **INTERNET: A Nova Mídia de Massa**, Belo Horizonte, set. [2001]. Disponível em: < <http://www.fafich.ufmg.br/~larp/001.htm>> Acesso em: 16 set. 2002.

MINASI, Mark. **Segredos de projeto de interface gráfica com o usuário**: Tradução, Flávio Morgado. Rio de Janeiro: Infobook, 1994.

NIKEL, Fábio. **A demanda de informações para o desenvolvimento de sistemas com interfaces ergonômicas**. 1994. 76 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

PASOLD, Fernando. **Gerador de home pages de apoio ao ensino utilizando princípios de design de interfaces**. 2000. 57 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

POMPERMAIER, Leandro Bento. **Modelagem de Aplicações Hiperídia: Uma experiência com OOHD**M, Porto Alegre, nov. [2000]. Disponível em: <<http://www.inf.ufrgs.br/~pomper/oohdm.html>>. Acesso em: 10 set. 2002.

REIS, Dalton Solano dos. **Design de interfaces**, Blumenau, out. [2000?]. Disponível em: <<http://www.inf.furb.br/~dalton/DiscipiPI/material.htm>>. Acesso em: 15 fev. 2002.

SCHWABE, Daniel. **The object oriented hypermedia design model (OOHDM)**, Rio de Janeiro, [1996a?]. Disponível em: <<http://www.telemidia.puc-rio.br/oohdm/OOHDM.html>>. Acesso em: 12 ago. 2002.

SCHWABE, Daniel. **Systematic hypermedia application design with OOHD**M, Carolina do Norte, [1996b?]. Disponível em: <<http://www.cs.unc.edu/~barman/HT96/P52/section1.html>>. Acesso em: 12 ago. 2002.