

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

**PROTÓTIPO DE UM SOFTWARE DE APOIO A
CONSTRUÇÃO DE INTERFACES PADRÃO WINDOWS.**

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

EDUARDO COMIN

BLUMENAU, NOVEMBRO/2000

2000/2-22

PROTÓTIPO DE UM SOFTWARE DE APOIO A CONSTRUÇÃO DE INTERFACES PADRÃO WINDOWS.

EDUARDO COMIN

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. Dalton Solano dos Reis — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

Prof. Dalton Solano dos Reis — Orientador na FURB

Prof. Carlos E. Negrão Bizzotto

Prof. Everaldo Artur Grahl

AGRADECIMENTOS

Ao meu orientador, prof. Dalton Solano dos Reis, pela atenção e dedicação dispensada a este trabalho.

A minha namorada e aos amigos e colegas pelo incentivo.

Principalmente aos meus pais que não mediram esforços para que eu alcançasse meu objetivo.

E a todos que de alguma maneira contribuíram com este trabalho.

DEDICATÓRIA

Dedico este trabalho à
toda minha família e amigos que
sempre me apoiaram e incentivaram.

SUMÁRIO

| | |
|--|-------------|
| AGRADECIMENTOS | iii |
| DEDICATÓRIA | iv |
| LISTA DE FIGURAS | viii |
| LISTA DE QUADROS | ix |
| RESUMO | x |
| ABSTRACT | xi |
| 1 INTRODUÇÃO | 1 |
| 1.1 MOTIVAÇÃO | 2 |
| 1.2 OBJETIVOS | 2 |
| 1.3 ORGANIZAÇÃO DO TEXTO | 2 |
| 2 ERGONOMIA | 4 |
| 2.1 ABORDAGEM ERGONÔMICA | 5 |
| 2.2 CRITÉRIOS ERGONÔMICOS | 6 |
| 2.2.1 CONDUÇÃO | 6 |
| 2.2.2 CARGA DE TRABALHO | 7 |
| 2.2.3 CONTROLE EXPLÍCITO | 7 |
| 2.2.4 ADAPTAÇÃO | 8 |
| 2.2.5 HOMOGENEIDADE E COERÊNCIA | 8 |
| 2.2.6 SIGNIFICÂNCIA DE CÓDIGOS E DENOMINAÇÕES | 8 |
| 2.2.7 GESTÃO DE ERROS | 9 |
| 2.2.8 COMPATIBILIDADE | 9 |
| 2.3 DIFICULDADE NA UTILIZAÇÃO DOS CONHECIMENTOS ERGONÔMICOS | 9 |
| 2.4 DEFICIÊNCIAS APONTADAS PELA ERGONOMIA | 10 |

| | | |
|---------|---|-----------|
| 2.5 | CICLO ERGONÔMICO X CICLO TRADICIONAL DE DESENVOLVIMENTO DE SOFTWARES | 10 |
| 2.6 | SISTEMAS ERGONÔMICOS | 12 |
| 3 | <i>INTERFACE</i> | 14 |
| 3.1 | CONCEITOS | 14 |
| 3.2 | A IMPORTÂNCIA DA INTERFACE HOMEM-COMPUTADOR (HCI) | 15 |
| 3.2.1 | DIÁLOGO HOMEM-MÁQUINA | 16 |
| 3.3 | PROBLEMAS NA INTERAÇÃO HOMEM-COMPUTADOR | 17 |
| 3.4 | NÍVEIS DE INTERAÇÃO DA INTERFACE COM O USUÁRIO | 17 |
| 3.5 | CARACTERÍSTICAS | 18 |
| 3.6 | OBJETIVOS DO DESENVOLVIMENTO | 18 |
| 3.7 | METÁFORAS PARA INTERFACES | 19 |
| 3.8 | TIPOS DE USUÁRIO | 20 |
| 3.9 | RECOMENDAÇÕES PARA O PROJETO DE INTERFACES | 21 |
| 3.9.1 | ESTILOS DE INTERAÇÃO | 22 |
| 3.9.1.1 | MENU | 22 |
| 3.9.1.2 | FORMULÁRIO PARA PREENCHIMENTO DE CAMPOS | 23 |
| 3.9.1.3 | LINGUAGEM DE COMANDO | 23 |
| 3.9.1.4 | LINGUAGEM NATURAL | 24 |
| 3.9.1.5 | CONTROLE DE MANIPULAÇÃO DIRETA | 24 |
| 3.9.1.6 | COMPARAÇÃO ENTRE OS ESTILOS DE INTERAÇÃO | 24 |
| 3.9.2 | RECOMENDAÇÕES PARA A ORGANIZAÇÃO DE INFORMAÇÕES | 26 |
| 3.9.3 | RECOMENDAÇÕES SOBRE A FORMA | 30 |
| 3.9.4 | RECOMENDAÇÕES PARA UTILIZAÇÃO DAS CORES | 32 |
| 3.9.5 | RECOMENDAÇÕES QUANTO AO TEXTO | 34 |
| 3.9.6 | RECOMENDAÇÕES PARA SAÍDA DE INFORMAÇÕES | 37 |
| 4 | <i>DESENVOLVIMENTO DO PROTÓTIPO</i> | 39 |
| 4.1 | ESPECIFICAÇÃO DO PROTÓTIPO | 39 |

| | | |
|------------|-----------------------------------|-----------|
| 4.2 | IMPLEMENTAÇÃO DO PROTÓTIPO | 40 |
| 4.3 | FUNCIONAMENTO DO PROTÓTIPO | 42 |
| 5 | CONCLUSÃO | 48 |
| 5.1 | CONSIDERAÇÕES FINAIS | 48 |
| 5.2 | EXTENSÕES | 48 |
| | REFERÊNCIAS BIBLIOGRÁFICAS | 49 |

LISTA DE FIGURAS

| | |
|--|----|
| <i>Figura 1- Ciclo Tradicional X Ciclo Ergonômico</i> | 12 |
| <i>Figura 2 - Diagrama de Contexto</i> | 39 |
| <i>Figura 3 - MER</i> | 39 |
| <i>Figura 4 - Janela Principal</i> | 43 |
| <i>Figura 5 - Menu Principal</i> | 44 |
| <i>Figura 6 - Formulário de cadastro de categoria</i> | 44 |
| <i>Figura 7 - Formulário para cadastro de recomendação</i> | 45 |
| <i>Figura 8 - Barra de ferramentas</i> | 46 |
| <i>Figura 9 - Visualizar Recomendação</i> | 46 |
| <i>Figura 10 – Tela Exercício</i> | 47 |
| <i>Figura 11 – Tela Sobre</i> | 47 |

LISTA DE QUADROS

| | |
|---|----|
| <i>Quadro 1 - Comparação entre Estilos de interação</i> | 25 |
| <i>Quadro 2 - Tabela Recomendação</i> | 40 |
| <i>Quadro 3 - Tabela Categoria</i> | 40 |
| <i>Quadro 4 - Associação do nodo aos componentes</i> | 41 |
| <i>Quadro 5 - Adicionar Nodo</i> | 42 |

RESUMO

Este trabalho de conclusão de curso visa um estudo sobre as recomendações para o desenvolvimento de projetos de interface, objetivando o desenvolvimento de um protótipo de um software de apoio a construção de interfaces padrão *Windows*, segundo as recomendações estudadas. Pretende-se que o protótipo possa ser utilizado por programadores como ferramenta de auxílio para a construção de interfaces para seus produtos.

ABSTRACT

This work of course conclusion seeks a study about the recommendations for the development of interface projects, objectifying the development of a prototype of a support software the construction of interfaces pattern Windows, according to the studied recommendations. It is intended that the prototype can be used by programmers as tool of aid for the construction of interfaces for its products.

1 INTRODUÇÃO

O software como elemento de ligação entre o *Hardware* e o problema a ser solucionado evoluiu de uma interação totalmente ligada à lógica de funcionamento da máquina para o uso de interfaces gráficas amigáveis. Com isso surgem alguns problemas, pois grande parte dos programadores não estão preparados para essa nova realidade, estimulando desta forma, pesquisas que possam solucioná-los.

Segundo [MIN1994], o crescimento da popularidade de ambientes gráficos como *X windows* da *UNIX*, *OS/2*, o *Macintosh*, e é claro o mais popular de todos, o *Microsoft Windows*, forçou muitos programadores a mover-se de campos familiares do *design* estruturado e programação estruturada, para o campo pouco familiar do *design* interativo.

Houve um certo tempo em que o uso de *Graphical User Interface* (GUI) eram recursos utilizados por poucos programadores que levavam dias para conseguir produzir muito pouco. Hoje em dia, com as ferramentas existentes no mercado, como: *Visual Basic*, *Borland Delphi*, *JBuider* e outras, permitem que se construa, em questão de minutos, um programa com GUI.

Talvez por essa tamanha facilidade com que se pode utilizar recursos GUI, é que estão aparecendo os novos problemas. Um programador que utilize um excesso de botões, barras de rolagem, janelas e menus, por exemplo, desenvolve normalmente um programa no qual o usuário se sente totalmente perdido, ficando sem nenhuma referência sobre como utilizar o software ([MIN1994]).

Segundo [REI1999], a ergonomia e o *design* trazem contribuições para o processo de desenvolvimento de interfaces em softwares interativos. A contribuição da ergonomia se deve ao seu poder multidisciplinar de adaptar o ambiente de trabalho (software) ao comportamento do homem. O *design*, colabora no sentido de facilitar o diálogo com o usuário através da disposição dos elementos gráficos da tela, tomando como base os princípios de *design*. É válido lembrar que apesar de existir todo um embasamento em torno do *design* e ergonomia, a geração da boa interface também considera o bom senso por parte do projetista.

Não existe nenhuma forma “certa” de se construir uma interface gráfica ou aplicativo com interface gráfica. O que se tem são recomendações sobre as partes que compõem uma interface gráfica, por exemplo, botões, janelas, cores etc. ([MIN1994]).

Em um ambiente gráfico espera-se ver as coisas como encontram-se no mundo real. Neste sentido uma ferramenta de desenho deve se parecer com um lápis, uma de *zoom* deve se parecer com uma lente de aumento e assim sucessivamente. Para a interação com o software tornar-se intuitiva, o usuário não deve sentir dificuldades em compreender uma função de uma dada ferramenta, pois quando isto ocorre há uma perda de praticidade da interface ([HEC1993]). Por exemplo, tem-se um botão com uma figura que não tem nada a ver com a função deste no software.

1.1 MOTIVAÇÃO

Este trabalho tem como motivação, justificando seu desenvolvimento, os seguintes;

- a) Impacto da ergonomia sobre a produtividade;
- b) Importância de se estudar as recomendações ergonômicas, de forma a melhorar a qualidade de interação do software.

1.2 OBJETIVOS

O objetivo principal deste trabalho é que a partir do estudo das recomendações acima citadas, seja possível a construção de um protótipo de um software de apoio que possa mostrar, se possível, de forma gráfica, erros e acertos em exemplos, para que este possa vir a se tornar uma ferramenta de auxílio a programadores para construção de interfaces padrão *Windows*.

1.3 ORGANIZAÇÃO DO TEXTO

No primeiro capítulo será mostrada a Introdução, o objetivo e a organização do mesmo.

O segundo capítulo traz uma visão geral da ergonomia, seus critérios, suas dificuldades na aplicação e uma comparação com outro ciclo de desenvolvimento.

No terceiro capítulo fala-se da interface, conceitos, características e principalmente um conjunto de recomendações para projeto de interfaces, elaborado por vários autores.

Detalhes do desenvolvimento do protótipo, como especificação, implementação e funcionamento, estão descritos no quarto capítulo.

Finalmente no quinto capítulo tem-se as conclusões, considerações finais e sugestões.

2 ERGONOMIA

Para [STE1994], define-se ergonomia como o conjunto de conhecimentos científicos relativos ao homem e necessários a concepção de instrumentos, máquinas e dispositivos que possam ser utilizados com o máximo de conforto, segurança e eficiência.

Segundo [JUN1995], de uma forma genérica, a ergonomia pode ser definida como “a adaptação do trabalho ao homem”. Esta definição, apesar de sintetizar o espírito da ergonomia, não fornece maiores informações sobre os meios empregados para levar adiante seus objetivos.

De uma forma mais precisa, a ergonomia pode ser definida com base na participação de diferentes áreas da ciência, trabalhando de maneira cooperativa na construção de instrumentos adequados ao homem quando da execução de suas tarefas ([JUN1995]).

A ergonomia divide-se em duas categorias básicas (PALMER 1976 *apud* [JUN1995]):

- **Ergonomia de concepção:** preocupa-se com o projeto dos componentes da interação homem-máquina, nas fases anteriores à construção da ferramenta, máquina ou execução da tarefa.
- **Ergonomia corretiva:** relaciona-se com a modificação dos componentes da interface homem-máquina, de forma a corrigir desvios, ou inadequações, existentes nas máquinas, ferramentas, ou tarefas.

Na informática, a ergonomia tem contribuído de várias maneiras para obtenção dos melhores resultados, desde a concepção ou correção dos postos de trabalho, atuando nas relações físicas entre o operador e a máquina até a concepção ergonômica de interface ([STE1994]). Esse último tipo de intervenção resulta do surgimento da ergonomia de *software*, cuja atuação se dá a nível da interface homem-computador.

A área de estudos conhecida como ergonomia de software surgiu nos anos 80, quando apresentou-se de maneira mais séria o problema da qualidade da interação homem-máquina (BARTHET 1988 *apud* [JUN1995]). A construção de um software que respeite as características de cada tipo de usuário e tarefa é o seu objetivo primordial.

A ergonomia da informática evoluiu muito ao longo dos últimos anos. Tornou-se uma disciplina importante e reconhecida, por razões econômicas (baixa dos custos, ampliação do parque de informática), razões sociológicas (os usuários não são somente profissionais da informática) e razões tecnológicas (cada vez mais informática interativa, distinção entre a categoria aplicação e a categoria interface, e instrumentos de prototipagem) (SCAPIN 1988 *apud* [STE1994]).

2.1 ABORDAGEM ERGONÔMICA

No que tange à interface homem-máquina, deve-se ainda destacar uma deficiência geralmente existente quanto aos métodos de sua concepção. Isto se deve ao fato de que estes métodos se baseiam na tarefa prescrita e não na tarefa que realmente ocorre no processo interativo do homem com a máquina. Assim, os sistemas fazem praticamente todas as funções imagináveis mas deixam a desejar no aspecto interativo ([REI1999]).

Nesse sentido, é que, para (CYBIS 1994 *apud* [REI1999]), no desenvolvimento de sistemas onde é forte a intervenção do usuário, não pode-se desconsiderar a abordagem ergonômica do mesmo, levando em conta dois aspectos: análise ergonômica e ciclo de protótipos (prototipação).

Na análise ergonômica existem duas etapas: a análise da tarefa e a análise da atividade. A primeira etapa trata da lógica de funcionamento das atividades, onde são abordados aspectos que descrevem formalmente como o usuário deve realizar suas tarefas. A segunda etapa trata da lógica de utilização, onde é feita uma análise do trabalho que está sendo realizado e descreve como realmente as pessoas efetuam suas atividades na prática ([REI1999]).

A prototipação se caracteriza por elaborar um ciclo de concepção rápida de sistemas em nível de interfaces, avaliação das versões e ajuste das mesmas, até a satisfação do usuário. Este método evidencia a participação ativa do usuário, através de avaliações empíricas e analíticas ([REI1999]). As primeiras se dão pela observação do usuário interagindo com o sistema, onde são realizados testes de adaptação e facilidades de uso, extraíndo-se informações sobre seu desempenho; no entanto, esta atividade requer a participação de ergonomistas e psicólogos, para a interpretação dos dados colhidos. As últimas se baseiam em

conhecimentos práticos resultantes da aplicação de teorias das ciências cognitivas às questões de projeto e avaliação destas interfaces; tais teorias dão origem aos critérios e recomendações ergonômicas analisados no capítulo 3.

2.2 CRITÉRIOS ERGONÔMICOS

No projeto de interfaces, pode-se fazer uso de critérios ergonômicos a serem considerados visando obter um produto com boa interatividade com o usuário. Tais critérios são caminhos e direções que podem ser seguidos na concepção de uma interface ergonômica. A seguir, analisar-se-á uma lista de critérios ergonômicos, conforme colocado por (BASTIEN1991 *apud* [REI1999]).

2.2.1 CONDUÇÃO

A condução pode ser entendida como sendo os meios que a interface possui no sentido de aconselhar, orientar, informar e conduzir o usuário na interação.

Um software com boa condução garante ao usuário informações sobre sua localização no sistema e a seqüência de ações para sua interação perfeita. A facilidade de aprendizado e de utilização que são conseqüências de uma boa condução, permitem diminuir o número de insucessos na interação.

A condução é formada por alguns sub-critérios:

- Presteza - corresponde às informações que permitem ao usuário identificar o estado ou contexto no qual ele se encontra, bem como às ferramentas de ajuda e seu modo de acesso. Uma interface com boa presteza guia o usuário no aplicativo lhe informando onde ele se encontra, o que ele fez para chegar onde está e para onde ele pode ir com segurança;
- Agrupamento/distinção de itens - este sub-critério diz respeito à maneira que os itens de informação estão organizados visualmente na interface. A compreensão de uma tela pelo usuário depende, dentre outras coisas, da ordenação, do posicionamento, e da distinção dos objetos apresentados. Esta organização das informações apresentadas através do agrupamento de itens semelhantes e itens distintos levam a uma melhor condução;

- *Feed back* imediato - é o retorno que o software dá aos tipos de ações que o usuário efetua. O mesmo precisa estar informado dos resultados relacionados com o processamento do sistema que ele interage;
- Legibilidade - é o critério relacionado com características que facilitam a leitura da informação na tela (cor da letra, fundo, espaçamento, tipo de letras, etc.). A *performance* na interação aumenta quando as características perceptivas do homem são levadas em conta.

2.2.2 CARGA DE TRABALHO

O critério carga de trabalho está relacionado com a forma de apresentação das informações para o usuário, provocando seu cansaço.

Os elementos de interface que tem um papel importante na redução da carga cognitiva e perceptiva do usuário, e no aumento da eficiência do diálogo estão relacionados com a carga de trabalho. Dentro da carga de trabalho tem-se a brevidade, que é a tentativa de diminuir a carga de trabalho de leitura e dos passos executados para se atingir os objetivos. A brevidade pode ser caracterizada pelas ações mínimas e a concisão. As ações mínimas se referem à carga de trabalho para se atingir objetivos específicos. A concisão se refere à carga para o nível perceptivo e mnemônico com relação aos elementos individuais das telas de entrada e saída de informações.

2.2.3 CONTROLE EXPLÍCITO

Este critério está ligado ao controle que os usuários devem ter sobre o processamento de suas tarefas pelo sistema, bem como o processamento explícito pelo sistema dessas tarefas. Os usuários devem ter controle total da situação na interação com o sistema. O controle explícito é dividido em dois sub-critérios:

- Controle do usuário - os usuários devem sempre ter controle na interação com o sistema. Estes, a qualquer momento, podem tomar decisões, como cancelar o processamento, suspender, continuar, interromper, etc;
- Ações explícitas - o computador deve somente executar as ações solicitadas pelos usuários.

2.2.4 ADAPTAÇÃO

A adaptação se refere à capacidade da interface de reagir segundo o contexto (modo de interação) e segundo as necessidades e preferências do usuário.

Uma interface não pode agradar todos os usuários mas pode se adaptar ao mesmo, ou seja, devem existir várias maneiras do usuário executar as tarefas, podendo ele escolher a que mais lhe agrada. Podem ser considerados dois sub-critérios:

- Flexibilidade - corresponde tanto ao número de diferentes meios à disposição do usuário para alcançar um objetivo, quanto à capacidade de personalização da interface, levando em conta as exigências do usuário;
- Experiência do usuário - deve-se considerar o grau de experiência que o usuário possui. Aquele que não tem experiências possui necessidades diferentes dos experientes.

2.2.5 HOMOGENEIDADE E COERÊNCIA

Este critério aborda a maneira coerente e homogênea com que os objetos (menus, mensagens, atalhos, etc.) são colocados para o usuário.

A forma de apresentação deve ser padronizada e conservada em contextos iguais. A falta de homogeneidade pode aumentar a busca de uma informação ou ação na interface.

2.2.6 SIGNIFICÂNCIA DE CÓDIGOS E DENOMINAÇÕES

Refere-se à maneira em que as ações são codificadas ou denominadas na interface. A nomenclatura para execução das tarefas deve ser sugestiva com relação ao tipo de tarefa a ser executada.

A interação e memorização das ações pelo usuário ficam muito facilitadas quando existe uma codificação e denominação expressivas, ou seja, que lembrem a ação correspondente.

2.2.7 GESTÃO DE ERROS

É a maneira de evitar ou de diminuir a ocorrência de erros de interação. Os sub-critérios são:

- Proteção contra erros - formas ou mecanismos de detectar quando um erro ocorreu;
- Qualidade das mensagens - a informação fornecida ao usuário sobre o erro e como corrigi-lo devem ser exatas;
- Correção de erros - informação dada ao usuário de como ele pode corrigir seus erros.

2.2.8 COMPATIBILIDADE

A compatibilidade é o acordo entre as características do usuário com suas tarefas a serem realizadas, e a organização das saídas, entradas e do diálogo de uma aplicação.

Em outras palavras, é a conformidade entre o usuário e as tarefas a serem realizadas através da interface.

2.3 DIFICULDADE NA UTILIZAÇÃO DOS CONHECIMENTOS ERGONÔMICOS

Segundo [STE1994], a dificuldade para a aplicação da ergonomia, ocorre porque as suas contribuições não são rapidamente assimiláveis, pois fazem referência à uma vasta área de conhecimento, e a uma metodologia específica. Isto tem exigido, a participação de programadores e de analistas em cursos de treinamento em ergonomia. Da mesma maneira, que os ergonomistas estão tendo que se especializar nas questões de informática.

Muitos programadores e analistas consideram que o estudo ergonômico freia o ritmo de desenvolvimento de seus sistemas, isto porque ocorrem modificações nos planos e metodologias já estabelecidas, e a necessidade de um trabalho de equipe com freqüentes reuniões de coordenação e validação da tarefa ([STE1994]).

Como uma tentativa para solucionar o problema de forma geral sugere-se a concepção de um ambiente inteligente de apoio as atividades de concepção de projeto e avaliação de interfaces com o usuário com base no conhecimento ergonômico.

2.4 DEFICIÊNCIAS APONTADAS PELA ERGONOMIA

Segundo (CYBIS 1994 *apud* [STE1994]), as deficiências de concepção podem extrapolar a capacidade de adaptação do usuário, conduzindo o problema no uso dos sistemas que vão desde sua má utilização até a modificação da tarefa, causando frustração, desinteresse e altas taxas de erro.

A ergonomia metodologicamente pode fazer frente a um ou outro grupo de deficiências do desenvolvimento dos sistemas ([STE1994]). Tais como:

- Concepção segundo a orientação funcional em detrimento daquela operacional;
- Concepção segundo o critério de desempenho do sistema ao invés de objetivos do usuário;
- Fornecer todas as informações e funções disponíveis e imagináveis ao invés de somente as essenciais e necessárias;
- Não prever erros humanos.

2.5 CICLO ERGONÔMICO X CICLO TRADICIONAL DE DESENVOLVIMENTO DE SOFTWARES

A Figura 1 apresenta o quadro elaborado por (VALLENTIN1987 *apud* [RIG1993]), comparando os passos percorridos na geração de softwares segundo o ciclo de desenvolvimento tradicional de software e o ciclo ergonômico. Mostra-se com clareza a diferença nas abordagens adotadas desde a etapa de projeto preliminar até o uso da análise do trabalho como elemento de entrada para manutenção.

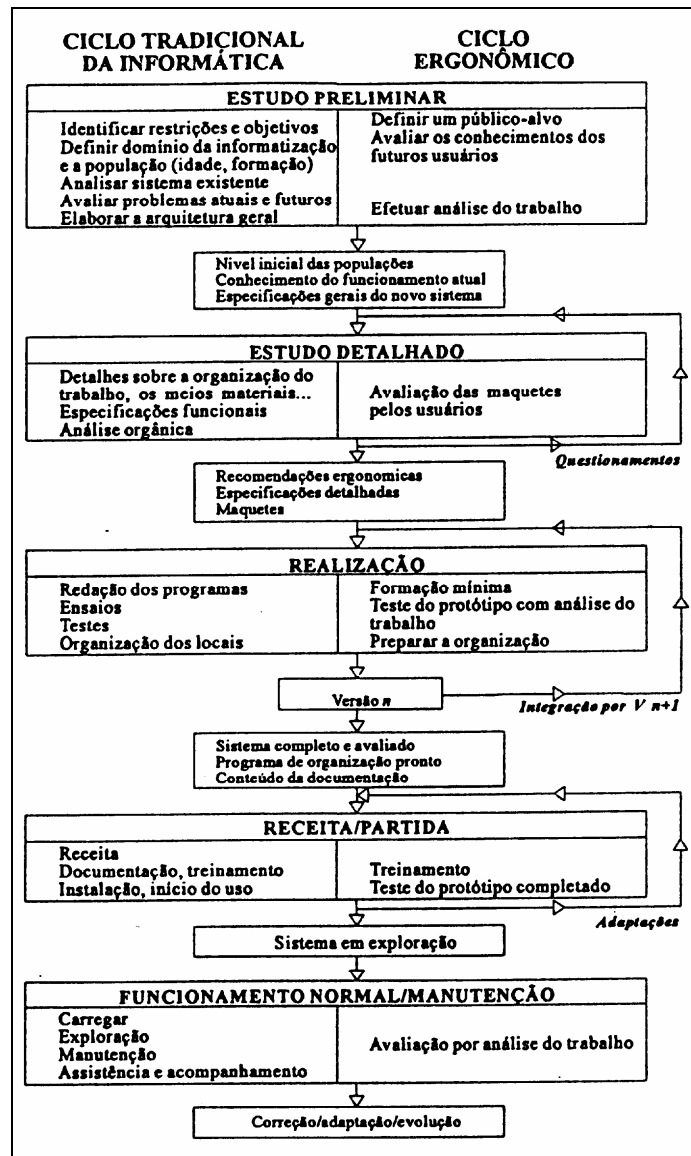
Segundo [RIG1993], o ciclo tradicional de desenvolvimento de softwares teve sua origem no período em que o processamento de dados era realizado exclusivamente em máquinas de grande porte (*mainframe*), com as quais apenas profissionais altamente especializados (analistas, programadores e operadores) entravam em acordo direto. Os demais participantes nas realizações das tarefas informatizadas forneciam os dados através de formulários, não atuando diretamente nos centros de processamentos de dados.

Os esforços para o aperfeiçoamento da análise e da programação naquele período se voltavam para a melhor exploração da capacidade do *hardware* e para o aumento de velocidade no processamento, fator preponderante nos custos ([STE1994]).

Segundo [STE1994], métodos tradicionais são os que moldam a realização das tarefas às características e potencialidades dos equipamentos e que consideram apenas superficialmente as características dos recursos humanos envolvidos na implementação do novo sistema durante a fase de projeto lógico. A ergonomia privilegia o conhecimento do usuário e o da tarefa como elementos básicos.

O método tradicional para desenvolvimento de sistemas e a abordagem ergonômica tem como diferença básica o elemento no qual se apóiam para geração de *softwares*. Enquanto, o método tradicional se apóia no sistema de informações e na estrutura organizacional, a abordagem ergonômica tem como elemento inicial para o desenvolvimento do sistema o estudo do usuário e sua tarefa.

Figura 1- Ciclo Tradicional X Ciclo Ergonômico



Fonte: [RIG1993]

2.6 SISTEMAS ERGONÔMICOS

As abordagens adotadas pela ergonomia tradicional não são adequadas para problemas com ênfase em tecnologia de informação. Os modernos sistemas de computação deveriam ajudar o usuário a resolver problemas, tomar decisões, planejar e outras atividades cognitivas ([STE1994]). Os softwares ergonômicos devem observar um conjunto de variedades de conhecimento.

Os objetivos e experiências no campo de softwares ergonômicos, conforme (FISCHER 1988 *apud* [STE1994]) podem ser resumidos nas seguintes séries de proposições:

- a) Uma das coisas mais importantes que um computador precisa fazer, portanto, é tornar as informações disponíveis resumindo-as para o usuário e apresentá-las em uma forma sensível;
- b) Os projetistas dos sistemas de computação precisam conhecer os diferentes tipos de memória humana. Como memória de curto termo, memória sensorial e memória de longo termo;
- c) Os projetistas deveriam explorar a habilidade dos seres humanos de processar informações visuais eficientemente. A técnica de apresentação visual (gráficos, cores, janelas, etc.) abrem possibilidade nesta área;
- d) Os usuários deveriam ser capazes de construir para si mesmo um modelo adequado de funcionalidade do sistema (como ele trabalha e o que está pretendendo fazer);
- e) Os sistemas de computação deveriam ser adaptáveis à diferentes usuários. Pois cada usuário tem um diferente nível de familiaridade com o sistema bem como experiência no seu campo;
- f) Diferentes sistemas de aplicação que são executados por um único usuário deveriam ter um padrão de interface com o usuário. A padronização de interfaces com o usuário é essencial se os sistemas de computação são para ser usados eficientemente e se é para haver uma livre transferência de conhecimento sobre como executar diferentes tarefas;
- g) As interfaces com o usuário precisam ser desenvolvidas por especialistas e conforme exigências requeridas. A interface com o usuário tem efeito decisivo na qualidade do trabalho o qual pode ser alcançado com o sistema. Ela também deveria ser padronizada para diferentes aplicações.

Segundo [STE1994], a ergonomia de *software* indica que projetar um sistema implica conceber o *software* dotado de inteligência suficiente para interagir com o usuário e entendê-lo em um aspecto amplo de opções de mesmo colóquio estruturado.

3 INTERFACE

Segundo [STE1994], a interface homem-máquina é de extrema importância, pois condiciona o desempenho do operador e, na medida em que estabelece como e quais as informações estarão a ele disponíveis, influencia fortemente na realização da tarefa. A interface é um trabalho mútuo, onde a máquina e o homem devem estar em sintonia. A construção desta interface é um desafio porque o ser humano possui particularidades que podem gerar uma grande variabilidade no sistema de produção automatizadas.

O uso de interfaces interativas visa auxiliar os usuários não especialistas, via recursos de ajuda e outras características essenciais. Mas por outro lado, o especialista não pode ser esquecido, pois é um crítico em potencial e se negará a usar sistemas que tornem seu trabalho desagradável [STE1994]. Por isso, o desenvolvimento de interfaces se tornou um desafio para os engenheiros de software e projetistas de sistemas.

A interface com o usuário é a parte mais importante e crítica de um programa aplicativo. É a única parte que os usuários finais vêem, pois não podem percorrer o interior do pacote para admirarem a eficiência do projeto interno, tudo o que vêem é o que esta na sua frente, na tela. Se o que se projeta é algo desajustado, lento, insensível ou difícil de aprender, o usuário terá todo o direito de acusar o aplicativo como sendo de má qualidade. Para que os objetivos previstos sejam alcançados deve haver uma maior participação do usuário na construção de sistemas.

3.1 CONCEITOS

Com o advento de novas tecnologias, em especial com a automatização de vários processos de trabalho, se torna importante a observação do relacionamento do homem com a máquina, já que o desempenho de ambos está interligado, e na saúde física e mental do operador depende muito desta interação e de como ela é desenvolvida ao longo da rotina de trabalho [STE1994].

Para [PRES1995], a interface é a porta de entrada do software ao qual estamos interagindo. Para que o trabalhador tenha uma boa interação com o sistema produtivo e conseqüentemente, com o trabalho é preciso que leve em consideração as condições de

trabalho em relação as exigências das tarefas e aos modos operatórios (REBELLO 1993 *apud* [STE1994]).

Uma interface homem-máquina engloba os aspectos dos sistemas informatizados que influenciam a participação do usuário em sua tarefas (SCAPIN 1986 *apud* [STE1994]). Nela estão incluídos os dispositivos de *hardware*, os programas de entrada e saídas, a arquitetura do diálogo, os manuais, os cursos de treinamento e o suporte (CYBIS 1991 *apud* [STE1994]).

3.2 A IMPORTÂNCIA DA INTERFACE HOMEM-COMPUTADOR (HCI)

Segundo [NIK1994], a importância da interface homem-computador tem origem em um princípio básico: o desempenho do sistema associado ao bem estar do usuário no trabalho informatizado. A interface é considerada por [RIG1993] como um elemento de vital importância na concepção de *softwares*, pois esta condiciona o desempenho do operador e, na medida em que estabelece como e quais informações estarão disponíveis, influencia claramente a realização da tarefa. A interação homem-computador pode ser considerada como uma verdadeira conversação, onde cada participante tem seus próprios objetivos (SCAPIN 1988 *apud* [NIK1994]).

Cada vez menos o usuário final precisa saber sobre linguagem de máquina, funcionamento de computadores, linguagens de programação, etc. A tendência é a máquina adaptar-se ao seu usuário (seja este o programador ou o cliente). Portanto, necessita-se desenvolver interfaces inteligentes, independentes do funcionamento dos computadores [STE1994].

Entretanto, (SCAPIN 1988 *apud* [NIK1994]), adverte para o fato de o computador ser diferente das demais máquinas que o homem utiliza. Mesmo sendo um equipamento com o qual o homem se relaciona fisicamente, através da manipulação do teclado ou *mouse*, o computador contém programas que representam (ou deveriam representar), uma extensão do cérebro humano, relacionando-se com os aspectos cognitivos do homem - como a percepção, a memória e a tomada de decisões -, enquanto as demais máquinas atuam apenas como extensões ampliadoras da ação física do homem. Para (BARTHET 1988 *apud* [NIK1994]), o fato de o usuário dialogar diretamente com o software de aplicação, e não com quem o

concebe, devendo portanto, ser corretamente associada a lógica geral do sistema de informações a ser levantada pelo analista com a lógica de utilização do sistema no posto de trabalho do usuário.

Assim, segundo [NIK1994], a implantação de sistemas informatizados no ponto de vista da interface homem-computador, pode ser relacionada como uma das causas da “barreira” que vem sendo imposta às pessoas sem formação específica, dificultando assim o acesso aos benefícios da informática. As possíveis causas para esses obstáculos, segundo (SCAPIN 1988 *apud* [NIK1994]):

- Conhecimento incompleto das tarefas e dos usuários do trabalho a ser informatizado;
- Falta de métodos apropriados para a concepção e avaliação de interfaces.

3.2.1 DIÁLOGO HOMEM-MÁQUINA

Segundo [STE1994], o diálogo, ou seja, a comunicação nos dois sentidos, entre o usuário e o sistema podem ser funcionalmente decompostos nas entradas, ou seja, a comunicação do usuário com o computador, e as saídas, a comunicação do computador com o usuário.

Qualquer sistema interativo tem obrigatoriamente, alguma forma de estabelecer o diálogo com o usuário. Quanto maior a participação do usuário, maior a possibilidade do sistema estar adaptado.

A característica fundamental de tais sistemas, é que o processamento seja controlado por escolhas do usuário em momentos convenientes.

De acordo com [STE1994], a forma deste diálogo é de importância vital, para que o projeto final não resulte em um sistema incompatível com as necessidades dos usuários. O usuário de um sistema interativo nem sempre está acostumado ou familiarizado com o uso de computadores. O importante para este usuário é trabalhar com uma ferramenta, que auxilie na solução do seu problema, que é o programa aplicativo.

3.3 PROBLEMAS NA INTERAÇÃO HOMEM-COMPUTADOR

Segundo [STE1994], os problemas na interação homem computador identificados através de pesquisa e relacionados por alguns estudiosos ([SCA1986], [CYB1991]) no assunto são:

- Falta de conhecimentos preliminares das tarefas e dos usuários por parte dos projetistas;
- Sendo o projetista do sistema especialista em programação e pouco ou nada entender da área em que o aplicativo vai funcionar;
- Utilização de jargões profissionais e termos estrangeiros ou desconhecidos na área do aplicativo;
- Falta de proteção às funções que acarretam grandes modificações no estado do sistema;
- Ausência de resposta do sistema para algumas ações dos usuários;
- Falta de guia e orientação;
- Conhecimento incompleto sobre a tarefa e os usuários do trabalho a ser informatizado;
- A ausência de métodos e ferramentas lógicas específicas para a concepção e avaliação de interfaces com o usuário;
- Considerar o computador como um fim em si mesmo.

3.4 NÍVEIS DE INTERAÇÃO DA INTERFACE COM O USUÁRIO

Segundo [STE1994], os níveis de abstração servem como forma de estruturação da interface e como critério de repartição de seus elementos intrínsecos. Devem ser considerados pelos projetistas, pois ajudam na qualidade dos sistemas interativos.

Os níveis de abstração da interação do usuário com o *software*, segundo [RIG1993], são quatro:

- Conceitual: pela lógica de funcionamento da máquina que define a tarefa através do conjunto de objetivos, métodos e informações que o usuário deve utilizar;

- Semântico: pela lógica de utilização do usuário que define o efeito das funções sobre os objetos correspondentes, ou seja, os significados dos vocabulários, abreviaturas, etc;
- Sintático: pela gramática do diálogo, a seqüência dos comandos e seus parâmetros, os agrupamentos, as possibilidades de *feedback* e de ajuda a detecção de erros;
- Léxico: pelas características individuais das entradas e saídas, como a programação visual das telas e *ícones*, códigos de cores, teclas de funções, etc.

3.5 CARACTERÍSTICAS

Sobre os itens que caracterizam os sistemas interativos, (SEGALIN 1989 *apud* [STE1994]) publicou uma lista, que serve como sugestão ao projetista de sistemas interativos e certamente não é suficiente para resolver todos os problemas. Os itens são:

- Usar concisamente linguagem “natural”, evitando códigos e permitindo abreviações;
- Usar entradas pequenas, facilitando a correção de erros;
- Permitir entradas simples para usuários novatos e entradas múltiplas para usuários experientes;
- Permitir que o usuário controle o tamanho das mensagens de erro;
- Emitir mensagens de erro polidas, sugestivas e informativas;
- Oferecer ajuda ao usuário em dificuldades ou quando ele solicitar;
- Usar uma linguagem de comandos simples e consistente;
- Fazer com que sob todos os aspectos, o sistema pareça estar sob controle do usuário;
- Evitar redundância nos diálogos;
- Adaptar o sistema às habilidades do usuário.

3.6 OBJETIVOS DO DESENVOLVIMENTO

Existem três objetivos no desenvolvimento de interfaces interativas, que são altamente independentes e, considerados pela maioria dos projetistas, segundo (SEGALIN 1989 *apud* [STE1994]) são eles:

- Simplicidade: um sistema interativo é simples se contém poucos comandos e, se os comandos tem uma estrutura consistente. A saída deve ser legível e as mensagens

de erro devem ter formato uniforme. A estrutura de comando deve ser compatível com o domínio do problema e com a seqüência de processos imaginada pelo usuário;

- **Potencialidade:** um dos maiores atrativos no uso do computador é a potencialidade que ele oferece. Um bom sistema deve ser dotado de comandos poderosos que permitam ao usuário alcançar facilmente seus objetivos. Os comandos devem facilitar o processo de resolução de problemas. Usuários experientes devem ter comandos especialmente poderosos a sua disposição, podendo criar macros e extensões de sistemas;
- **Satisfação do Usuário:** a satisfação do usuário independente da eficiência do sistema. Um sistema pode ser eficiente, mas desagradável de usar.

Segundo [STE1994], para o usuário final, o mais importante, além do poder, é o tempo de resposta as suas solicitações, não importando se isto esta sendo feito de forma eficiente ou não. Evidentemente, não se deve tomar isso como regra na hora de se projetar um sistema, pois o consumo excessivo de memória do computador, por exemplo, nem sempre é justificado pela velocidade com que as tarefas são executadas. O ideal é que se consigam projetar sistemas que atendam as necessidades do usuário final, proporcionando-lhe satisfação, dentro de uma política que evite desperdício de recursos.

3.7 METÁFORAS PARA INTERFACES

Para [JUN1995], pode-se definir uma metáfora, na área de interfaces, como sendo a transposição da imagem ou conceito do mundo real para um modelo em computador. Este modelo não deve necessariamente ser uma representação literal; ele deve sobretudo carregar as funcionalidades.

Uma das primeiras empresas a perceber o potencial de se projetar interfaces que tenham semelhança com o mundo concreto com as quais as pessoas estão familiarizadas foi a Xerox. Em vez de desenvolver metáforas verbais como uma forma de ajudar os usuários a compreender a interface, eles foram um passo além e projetaram uma metáfora para a interface que estava baseada no escritório. O resultado foi a interface do usuário com o *Star* (Sistema Operacional desenvolvido pela Xerox). O aspecto central da metáfora da interface foi criar contrapartidas eletrônicas para os objetos de um escritório. Isto significou representar

os objetos na tela através de ícones. Foram incluídos papéis, pastas, arquivos, e caixas de entrada e saída. A metáfora genérica que era apresentada na tela era a de um tampo de mesa (um *desktop*), que se assemelhava a mesa de um escritório típica.

Os usuários interagem com os *ícones*. Assim como uma pessoa abre, fecha, copia e joga fora arquivos de papéis no mundo físico, a interface foi projetada de forma que ações equivalentes pudessem ser realizadas sobre as versões eletrônicas. O truque foi projetar um dispositivo de entrada que permitisse a realização destas ações eletrônicas intuitivamente. Assim o *mouse* foi desenvolvido para permitir ações equivalentes à manipulação física dos documentos, que era obtida via “*clicar*”, “*apontar*”, “*selecionar*”, “*mover*” e “*arrastar*”.

A metáfora auxilia o usuário a montar uma imagem da funcionalidade geral existente no *software*, relacionando-a ao mundo real [JUN1995].

Alguns exemplos ilustram a utilização de metáforas [HEC1993]:

- O modelo da interface dos ambientes gráficos MS-Windows e Macintosh utiliza-se de forma intensiva da metáfora da escrivaninha, tendo como elementos as pastas de trabalhos e os arquivos;
- A estrutura do sistema de arquivos do DOS, com o uso de diretórios e arquivos;
- O conceito de ponteiros;
- Os sistemas de correio eletrônico.

[HEC1993] apresenta algumas considerações quanto ao uso e escolha de metáforas no projeto de interface:

- A metáfora escolhida deve apresentar o máximo possível de estrutura, isto é, deve prover uma larga gama de possibilidades;
- A metáfora deve ser relevante no contexto do problema;
- Ela deve ser facilmente representável;
- Os usuários devem se identificar com a mesma;
- A metáfora deve ser extensível.

3.8 TIPOS DE USUÁRIO

Segundo [JUN1995], de uma forma geral podem ser divididos em três grupos segundo o nível de conhecimento que possuem e a forma como exploram a execução das tarefas:

- Usuários Novatos: possuem pouca intimidade com o software, necessitando de um maior nível de auxílio para execução de suas tarefas. Exploram o software de maneira mais dirigida;
- Usuários Intermediários: sabem como executar as tarefas básicas no software e possuem um nível razoável de automatismos já criados. Começam a explorar as funcionalidades adicionais do software;
- Usuários Experientes: dominam as funcionalidades do software e baseiam sua ação grandemente em automatismos.

A frequência de utilização tem impacto nesta classificação pois usuários intermediários ou experientes podem necessitar das funcionalidades destinadas aos usuários novatos quando da execução de tarefas que não são frequentemente utilizadas.

3.9 RECOMENDAÇÕES PARA O PROJETO DE INTERFACES

Segundo [REI1999], a ergonomia e o *design* trazem contribuições para o processo de desenvolvimento de interfaces em softwares interativos. A primeira, contribui devido ao seu poder multidisciplinar de adaptar o ambiente de trabalho (software) ao comportamento do homem, usando os critérios ergonômicos. O último, colabora no sentido de facilitar o diálogo com o usuário através da disposição interfigural e intrafigural dos elementos gráficos da tela, por meio dos princípios de *design*. É válido lembrar que apesar de existir todo um embasamento em torno do *design* e ergonomia, a geração da boa interface também considera o bom senso por parte do projetista.

Deve-se salientar que, além das recomendações citadas na seção 3.9.2, deve-se optar pelo estilo de interação que prevalecerá no software que será projetado.

Segundo [REI1999], como referência à apresentação das recomendações, foram utilizados os autores: Cybis [CYB 1994], Dillon [DIL 1992], Easterby [EAS 1970], Fukuda [FUK 1990], Gould [GOU 1985], Hodgson [HOD 1985], Langen [LAN 1990], Marcus [MAR 1990], Morland (1983), Santos (1992), Shneiderman [SHN 1987], Vallentin [VAL 1987] e Yamamoto [YAM 1992].

Estas recomendações serão citadas a partir dos autores acima mencionados com posteriores observações no sentido de facilitar seu entendimento e possível aplicação prática [REI1999].

3.9.1 ESTILOS DE INTERAÇÃO

Geralmente, a interface de um software está calcada em um estilo interativo, que se caracteriza por ser mais eficiente ou não, de acordo com o perfil do usuário e da tarefa a ser realizada. Nessa perspectiva, a escolha de um estilo em detrimento de outro, é um dos fatores que influenciam diretamente no sucesso da interação [REI1999].

Visando situar o leitor nos vários estilos de interação, de modo a facilitar a melhor compreensão das recomendações e propor uma visão crítica na possível escolha de um estilo, a seguir far-se-á uma descrição dos mesmos com base em estudos de Odebrecht [ODE1994].

3.9.1.1 MENU

Este estilo caracteriza-se por uma lista que possui um número limitado de opções, cujo diálogo é controlado pelo usuário, que pode escolher uma destas opções. Geralmente, estas opções estão associadas a números, letras, palavras ou frases que identificam e invocam as funções correspondentes a determinada opção.

O menu é usado preferencialmente nos ambientes em que o usuário não sabe quais opções estão disponíveis para o acesso. Pelo fato do menu deixar bem claro quais opções estão disponíveis, ele torna-se adequado a usuários inexperientes. No entanto, o menu poderá tornar-se cansativo a medida que o usuário adquirir mais intimidade com a interface com o qual interage. Uma outra desvantagem, é que se a hierarquia de menus possuir muitos níveis, poderá trazer lentidão na escolha da opção, além de ocupar boa parte da tela.

Dentre os vários tipos de menus, convém destacar os de seleção simples (textos), onde possui somente duas opções; e os menus de seleção múltipla, onde poderá se escolher mais de dois itens. Pesquisas demonstram que o tempo de escolha de uma opção de uma lista com mais de seis itens, cresce enormemente. Isto ocorre pelo fato de a memória de curto termo ter capacidade limitada, uma vez que normalmente suporta não mais que sete itens.

Das diversas maneiras que se possui para acessar uma opção no menu, a mais utilizada e a que surte mais efeito, é o uso do mouse, e também o posicionamento do cursor através de uma barra que destaca a opção apontada.

3.9.1.2 FORMULÁRIO PARA PREENCHIMENTO DE CAMPOS

Neste estilo de interação o usuário é quem digita as instruções ou dados, preenchendo um campo ou utilizando uma linguagem formal. A linguagem para esta interação é clara e concisa, já que é predefinida e está limitada a uma aplicação em particular. Existe uma desvantagem para usuários novatos, pois precisam aprender esta linguagem que está presente no processo interativo, apesar de diminuir em muito a taxa de erros na interação porque o usuário normalmente tem certeza do que está escrevendo.

Este tipo de interação é semelhante à linguagem natural. A diferença está no fato da linguagem natural possuir uma sintaxe, semântica e estrutura léxica bem definida, mas em muitos casos é ambígua. A linguagem utilizada para preencher os campos é particular do domínio da aplicação.

3.9.1.3 LINGUAGEM DE COMANDO

É comumente utilizado em linguagens de programação e sistemas operacionais. Para utilizar esta linguagem, o usuário necessita um certo tempo para aprender como a comunicação se dará. É preferida por usuários experientes em informática. O processo de interação ocorre através de comandos digitados ou por teclas de função que ativam estes comandos via teclado.

Como vantagens deste estilo temos: a flexibilidade, a rapidez na execução de tarefas complexas e o apoio à iniciativa do usuário. Por outro lado as desvantagens abrangem: a dificuldade de memorização, fraqueza no tratamento de erros, além de requerer muito treinamento e experiência por parte do usuário.

3.9.1.4 LINGUAGEM NATURAL

A linguagem natural tenta colocar no diálogo homem-computador uma linguagem utilizada no cotidiano do usuário para se comunicar com as pessoas. Aqui o usuário não necessita aprender nada de novo sendo uma vantagem para usuários *ad-hoc*.

O grande problema da linguagem natural é de que existe ambigüidades na sua utilização. Um exemplo de ambigüidade pode ser verificado na seguinte oração: "Ela tem vestido azul". Aqui não se sabe se ela tem um vestido de cor azul ou se ela freqüentemente usa cores azuis. Para retirar a ambigüidade é necessário avaliar todo um contexto.

O campo de linguagem natural é uma área de estudos da Inteligência Artificial na tentativa de criar transparência total entre a máquina e o homem.

3.9.1.5 CONTROLE DE MANIPULAÇÃO DIRETA

O controle de manipulação direta se caracteriza por elaborar uma representação do domínio da aplicação através da utilização de objetos gráficos. O usuário através da utilização de mouses pode acessar estes componentes gráficos, que formam metáforas do mundo real, e implementam funções requisitadas pelo usuário. A interface é tratada como um *desktop* (mesa de trabalho) onde existem os ícones e figuras que representam as tarefas a serem executadas quando ativadas. Este estilo tem grandes vantagens como necessitar pouco treinamento devido ao seu caráter visual, além de ser de fácil memorização e aprendizado. O padrão do sistema operacional baseado em janelas exemplifica este tipo de interface, e normalmente necessita de mouse e vídeo gráfico.

Dentre os estilos de interação apresentados este parece o mais acessível tanto para usuários experientes como para novatos, além de permitir uma boa aproximação entre o mundo real e o computacional através de metáforas, mediante a utilização dos elementos gráficos.

3.9.1.6 COMPARAÇÃO ENTRE OS ESTILOS DE INTERAÇÃO

Segundo [RIG1993], as características das informações a serem apresentadas nas telas dos *softwares* são diretamente afetadas – e muitas vezes ditadas – pelos estilos de interação escolhidos para a realização do diálogo.

O quadro elaborado por (SHNEIDERMAN 1988 *apud* [RIG1993]) (ver Quadro 1) mostra um comparativo entre os estilos de interação, apresentando com clareza as vantagens e desvantagens de cada um deles.

Quadro 1 - Comparação entre Estilos de interação

| ESTILOS | VANTAGENS | DESVANTAGENS |
|------------------------|--|---|
| Seleção de Menus | Diminui treinamento e reduz digitação. Estrutura o processo de decisão, permite uso de ferramentas de gerenciamento diálogo. Facilidade para suportar tratamento de erros. | Perigo de haver muitos menus. Pode fazer usuários frequentes ficarem mais lentos. Requer espaço na tela. Requer rápida ' <i>display rate</i> ' |
| Preencimento de campos | Simplifica a entrada de dados. Requer treinamento simples. Mostra o contexto da atividade. Permite uso de ferramenta de gerenciamento da forma. | Consome espaço na tela. Requer habilidade para digitação. Erros digitação. |
| Linguagem de comando | É flexível. Apóia a iniciativa do usuário. É simpático ao "vigor" do usuário. Potencialmente rápido para tarefas complexas. Capacidade para suportar macros. | Dificuldade de retenção. Fraco no tratamento de erros. Requer treinamento substancial. Requer memorização substancial. |
| Linguagem natural | Alivia a necessidade do aprendizado da sintaxe. Pode não indicar o contexto, imprevisível. | Requer diálogos claros. Pode requerer mais digitação. |
| Manipulação Direta | Apresenta a tarefa visualmente. Fácil aprendizado. Fácil memorização. Erros podem ser evitados. Encoraja exploração e fornece alta satisfação subjetiva. | Pode requerer display gráfico. Pode requerer dispositivos para apontar (<i>mouse</i> , etc). Mais esforço de programação para aperfeiçoar ferramentas. Pode ser mais difícil escrever macros. |

Fonte Adaptada: [RIG1993]

3.9.2 RECOMENDAÇÕES PARA A ORGANIZAÇÃO DE INFORMAÇÕES

No âmbito das recomendações, estas se referem a melhor maneira de se organizar a informação na tela para que seja mais compreensível a quem interage. A organização da informação está relacionada com o tipo de informação que estará disponível a cada momento, bem como de que maneira esta ficará disponível, contribuindo para a boa interação.

Para a organização das informações tem-se as seguintes recomendações:

- Definir o que, quando e como apresentar a informação. O usuário não deve precisar buscar informações em outra tela para realizar uma tarefa. As informações exibidas devem ser apenas as necessárias para a realização desta tarefa. É inútil colocar em uma interface diversas funções que, naquele contexto, não interessam o usuário, confundindo o mesmo e preenchendo sua capacidade de memória com informações desnecessárias.
- Organizar a informação por níveis de importância para a realização da transação. Utilizar critérios para os diferentes níveis, como por exemplo: importante, secundário e periférico. É importante disponibilizar nas telas as informações realmente relevantes naquele momento, e as de menos importância deixar pouco visível.
- Apresentar os dados de forma diretamente utilizável pelo usuário. Evitar que o usuário tenha que realizar conversões, traduções ou interpretações. O usuário não pode estar sujeito a buscas constantes na sua memória para fazer conversões de códigos, nomes e números correspondentes ao que ele realmente deseja. Por exemplo, ao colocar-se um código numérico para uma cidade e durante a interação necessitar-se, continuamente, saber este código. Este fato irá trazer desconforto na utilização, pois o usuário é obrigado a se lembrar qual o código corresponde a cada cidade em determinado momento.
- Oferecer diferentes modos de apresentação da informação para a escolha do usuário. É importante dar possibilidade ao usuário para a escolha de uma interface que melhor lhe atenda, ou seja, usuário novato tem necessidades diferentes que um usuário experiente. A interface deve prever vários níveis de usuários. Por exemplo,

para usuários novatos utilizar uma interface com condução por exemplos, já os experientes permitir a utilização de macro funções.

- Adotar princípios lógicos para ordenar listas. Quando tem-se uma lista de palavras na tela é importante adotar um padrão de ordenação que seja melhor compreendido pelo usuário. Por exemplo, a enumeração de unidades de uma empresa deve começar pela matriz e não pela unidade que está em primeiro numa ordenação alfabética. A ordenação alfabética somente deverá ser utilizada quando não for possível a adoção de outro princípio lógico.
- Fornecer documentação em tela sobre o que, onde e como fazer para prosseguir a transação. Não se pode deixar o usuário perdido em hipótese alguma. Ele sempre deve saber onde está e como fará para prosseguir na interação. Esta recomendação faz forte uso do critério ergonômico chamado condução, como visto na seção 2.2.1.
- Procurar homogeneidade entre as telas. Utilizar os mesmos princípios organizacionais, as mesmas formas e cores e a mesma disposição espacial dos elementos em toda a interface. O usuário deve sempre seguir o mesmo padrão de interação, evitando a quebra no ritmo de trabalho. Não obstante, esta quebra pode ser utilizada com o objetivo de chamar a atenção num determinado ponto da interface.
- Preencher no máximo 25% da tela com elementos gráficos e textuais. A aglomeração da informação aumenta exponencialmente o tempo de busca e a dificuldade na realização da tarefa. Se a tela está muito cheia fica bastante difícil encontrar o que se procura.
- Minimizar o número de divisões principais da tela. Agrupar as informações em blocos lógicos e graficamente hierarquizados facilita a ação e a memorização. Como já foi verificado, nossa memória de curto termo tem normalmente a capacidade limitada a sete itens.
- Usar preferencialmente *display* positivo, ou seja, informações em preto ou em cores, sobre o fundo claro. Com *display* positivo, 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. É importante que o *display* positivo

permaneça em todo processo de interação, evitando a alteração contínua no foco de visão.

- Começar a tela com título ou cabeçalho que descreva rapidamente o conteúdo ou propósito da tela. Uma hierarquia de telas com seus respectivos títulos traz uma melhor definição dos mesmos, contribuindo para homogeneidade e clareza.
- Usar no máximo 30 tipos de ícones e 15 formas geométricas diferentes no conjunto das telas do software. Se o uso de componentes gráficos for demasiado fica difícil saber qual tarefa está associada a cada componente. O ambiente Windows possui um lembrete que aparece quando o mouse incide sobre um ícone.
- Usar o menor número de colunas possíveis na composição das telas e janelas. Estudos indicam que o limite de capacidade do usuário suporta com tranquilidade até quatro colunas com doze linhas cada, para as telas apresentadas.
- Programar o cursor para estar automaticamente posicionado na primeira casa da área para a entrada de dados. O usuário não tem que ir com o cursor até a posição onde deverá digitar alguma entrada de informação. O sistema deve colocar o cursor no local certo, evitando a perda de tempo.
- Usar simetria para denotar formalidade, estabilidade e ausência de movimento. Os arranjos simétricos são facilmente decifrados pelos usuários, mas são pouco atraentes devido a monotonia que causa durante a interação.
- Usar composição assimétrica para denotar informalidade. Os arranjos assimétricos são mais informais e despertam o interesse do usuário. Quando se deseja chamar a atenção do usuário para determinada característica, como foi visto anteriormente, se usa a assimetria.
- Aproximar componentes entre si para que formem grupos. Elementos que estejam muito próximos entre si perdem sua identidade e são percebidos como se o grupo fosse um só elemento. Esta recomendação pode reduzir o número de componentes na tela, tornando-a visualmente mais limpa. Um exemplo é a barra de ferramentas do *Windows*, onde tem-se os objetos reunidos num canto da tela, podendo estes acionar ações correlacionadas entre si.

- Usar similaridade para formação de grupos. Elementos similares em tamanho, forma, cor, aparência, direção, valor ou velocidade tendem a se ligar perceptivamente, formando grupos. Por exemplo, os comandos para ativar determinada função pode ter uma característica pertinente a um grupo específico.
- Usar contraste para criar distinção e prevenir ambigüidades. O mais primário elemento de detecção de um objeto é a diferença entre ele e seu entorno. 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.
- Usar direção como codificação por distinção para que dois elementos iguais transmitam informações diferentes. Um exemplo é a utilização de setas iguais em direções diferentes para denotar caminhos ou atividades diferentes. O desenho de um disquete com uma seta saindo de um *drive* (unidade de disco flexível) indica uma cópia de segurança, já um o desenho de um disquete com uma seta entrando no *drive* indica uma restauração da cópia.
- Usar direção associada à repetição para denotar movimento. Um símbolo que indica direcionamento repetido várias vezes, transmite ao usuário a sensação de movimento. Esta recomendação é baseada no princípio de *design* chamado de repetição.
- Usar anomalia para atrair a atenção do usuário para determinado ponto da tela. O uso de uma anomalia para quebrar um padrão interfacial é necessária para chamar a atenção a um determinado ponto da tela ou indicar uma situação de alerta.
- Personalizar itens pela atribuição de comprimentos característicos. Por exemplo, utilizar comprimentos maiores para itens de maior relevância no acesso às funções. Um outro exemplo é o uso de um comprimento padrão a um grupo de itens que acessam funções afins.
- Usar inclinação característica para codificar itens por grupamento e distingui-los dos demais. A inclinação de alguns itens na tela provoca a formação de grupos de itens que facilitam na caracterização e identificação dos mesmos dentro de uma interface.

- Usar inclinação para criar ênfase em um ponto específico da tela. Nos textos, o uso de caracteres itálicos cria ênfase. Este tipo de inclinação não deve ser constante na interface, pois perde seu poder de enfatizar pontos importantes. A ênfase se cria devido a presença de um comportamento anômalo nas letras ou objetos da tela.
- Usar cor para distinção rápida entre itens. As cores são rapidamente percebidas pelos usuários. A diferenciação da cor entre os elementos e seu entorno deve ter contraste suficiente para não ocorrerem ambigüidades e difícil visibilidade. A cor também pode ser utilizada para formar agrupamento de itens.
- Usar piscagem para codificar com alto grau de distinção. A piscagem tem o poder de chamar total atenção do usuário devido a 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.

3.9.3 RECOMENDAÇÕES SOBRE A FORMA

Este tipo de recomendação diz respeito tanto à geração de componentes (ícones, desenhos, figuras, etc.) para transmitir a informação adequada, quanto à estrutura formal e espacial das telas. Estas recomendações são fortemente baseadas nos princípios de *design*, abordados anteriormente, e são colocadas da seguinte maneira:

- 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.
- Relacionar gráfica e funcionalmente as telas com os formulários de apoio. Se os formulários de apoio, ou seja, formulários onde estão escritas as informações a serem digitadas, tiverem o mesmo padrão das telas, o processo de digitação será muito facilitado. É uma questão de compatibilidade.
- Usar elementos gráficos com movimento em tempo real para saídas que exijam longos tempos de espera. O movimento real de ícones ou termômetros, para indicar a porcentagem de processamento executada, situa o usuário no processo interativo, dando um *feed-back* em operações demoradas.

- Usar ícones e símbolos para identificar ações concretas. Os ícones ou elementos gráficos devem aparecer como metáforas das ações a serem realizadas na prática, ou seja, devem ser expressivas ao contexto do usuário.
- Usar descrição textual associada ao ícone ou símbolo quando necessário. A descrição textual se faz necessária quando o ícone ou elemento gráfico não é claro o suficiente. A descrição deverá vir logo abaixo do ícone que é identificado.
- Evitar a possibilidade de ocorrerem ambigüidades na leitura do ícone ou símbolo. Os símbolos que formam os ícones de uma determinada interface têm um significado especial para o ambiente em que o software está inserido. É necessário ter cultura no tipo de implementação que o sistema sugere. Também, necessário se faz que os símbolos utilizados tenham significados únicos no contexto do software, evitando as ambigüidades.
- Delimitar claramente o contorno e as divisões internas do ícone ou símbolo. Deve-se deixar bem claro onde começa um ícone e termina outro, evitando confusões na identificação e aumento do tempo de escolha. Quando um ícone não possui contorno ele pode ser delimitado através do contraste do seu interior com o seu arredor.
- Imprimir unidade ao conjunto dos elementos que compõem os ícones ou símbolos. O ícone sempre tem que ser percebido como um elemento único que acessa uma função que é caracterizada simbolicamente pelo desenho da figura impressa no ícone. Se ele for percebido como um grupo de elementos pode haver problemas de associar este ícone a uma função específica.
- Imprimir estabilidade aos ícones ou símbolos. Um ícone deve ser bem desenhado para impedir a dupla interpretação ocasionada por problemas de construção do mesmo. Além de ser bem projetado, o ícone deve ter sempre o mesmo significado durante a interação.
- Realizar testes com usuários típicos assim que possível. A prototipação de sistemas de informação tem papel importante para construção de um padrão interfacial que obedeça os requisitos do usuário.

- Criar preferencialmente ícones e símbolos simétricos. A simetria como um dos princípios de *design* já vistos, pode trazer para os ícones estabilidade e legibilidade, além de contribuir no balanço visual da tela. Simetria é a disposição de elementos similares em torno do eixo central e de equilíbrio da interface.
- Usar tamanho e escala para sugerir tridimensionalidade nos ícones e símbolos. Este tipo de recomendação é baseada no princípio de tamanho e escala, 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.
- 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.
- As características do monitor devem ser consideradas, sobretudo se for *touch-screen*, os ícones devem ter dimensões compatíveis aos dedos dos usuários.

3.9.4 RECOMENDAÇÕES PARA UTILIZAÇÃO DAS CORES

É importante considerar o ajuste de um esquema cromático que traga adequação ao componente humano, devido a utilização abundante de vídeos coloridos atualmente.

Tem-se as seguintes recomendações pertinentes às cores:

- Criar e aplicar cores em etapas separadas. Criar as telas no modo monocromático, cuidando da codificação da informação pelo seu conteúdo e forma, e só depois adicionar cuidadosamente cores onde elas possam ser úteis ao usuário. A cor não deve desviar a atenção do que realmente se quer mostrar.
- Usar no máximo sete e no mínimo três cores de maneira planejada para cada tela. Por exemplo, cores diferentes para o menu, título, ilustrações, mensagens de erro e fundo de tela. Como a cor tem o poder de agrupar por distinção, havendo muitas cores se perceberá vários grupos distintos e dificultará na memorização. Segundo (MARCUS 1992 *apud* [REI1999]), apesar das placas de vídeo atuais dispor de uma resolução com 16 milhões de cores, a mente humana só é capaz de discriminar 7,5 milhões de cores.

- 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. A capacidade de percepção humana trata estas características.
- Imprimir simplicidade e clareza no esquema cromático. 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.
- Usar combinações legíveis para textos ou figuras e fundos. As combinações: preto-branco, amarelo-preto, branco-vermelho, preto-amarelo, cian-magenta, amarelo-marrom, cian-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 contraste, harmonia e anomalia.
- 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.
- 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 está presente em 8% dos homens na Europa. Para auxiliar neste caso é recomendável utilizar fundo verde e letras vermelhas sobre ele, permitindo a percepção de texto e fundo pelos daltônicos. Idosos e usuários que operam terminais por longo período percebem melhor cores saturadas, ou seja, com bastante pigmentação.
- Considerar os níveis culturais e profissionais na codificação cromática. As conotações das cores podem variar fortemente entre culturas e entre profissões: 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.

- Considerar as variações interpessoais nos conceitos estéticos dos usuários. 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.
- Considerar restrições do ambiente e dos equipamentos na codificação cromática. É importante saber que a iluminação do ambiente pode interferir na percepção das cores. Além do ambiente os monitores de vídeo perdem sua capacidade de coloração com o uso através do tempo.
- Para as telas a serem impressas a cores, considerar as restrições técnicas à reprodução das cores. Quando se deseja imprimir telas do software deve-se considerar que a resolução dos monitores de vídeo é, geralmente, maior que a das impressoras usadas. Assim, a qualidade de impressão não será a mesma da tela.
- O uso de combinações de certas cores podem criar pós-imagens (fantasmas) no contorno da figura. Um exemplo disso são as combinações verde-vermelho e azul-vermelho.
- Considerar o contexto profissional na codificação cromática. 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: discricção.

3.9.5 RECOMENDAÇÕES QUANTO AO TEXTO

As recomendações dos textos se caracterizam por abordar questões como o tamanho do texto, a forma como ele é disponibilizado, espaçamento, tamanho de fonte, etc. É a maneira de trazer coerência aos elementos textuais que se apresentam na interface.

- Eliminar amenidades sociais nos comandos e menus. O uso de expressões como "por favor", "bom dia", "para incluir pressionar o botão inclusão", etc., contribuem para diminuir a clareza de interação, além de provocar o descontentamento na interação por parte de usuários experientes.
- Imprimir legibilidade ao texto. Deve-se utilizar sempre que possível letras maiúsculas e minúsculas. As maiúsculas devem ser utilizadas em palavras que se

deseja dar ênfase. Para garantir compatibilidade com os formulários de dados, se deve colocar letras escuras sobre fundo claro. Usar no máximo quatro a três tamanhos de letras e três fontes distintas nos textos a serem apresentados. A legibilidade do texto é um problema do princípio de *design* chamado harmonia.

- 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.
- Evitar o uso de textos em maiúsculas. Quando se utiliza o texto com maiúscula, o espaçamento entre linhas fica menor, dificultando a leitura. Se o texto é escrito todo em maiúsculo a estratégia de criar ênfase para alguns pontos da tela fica prejudicada, pois o maiúsculo é comum e não provoca anomalia.
- Usar piscagem nos textos para mostrar erros ou iluminar pontos. A piscagem, fundamentada no critério anomalia, 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.
- 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 pois, segundo [RIG 1993] “a capacidade da memória de curto termo para reter informações textuais é limitada a 5 caracteres alfabéticos”.
- 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.
- Imprimir identidade e legibilidade ao título. O título deve ser, preferencialmente, centralizado ou ajustado pela 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.

- Com relação aos comandos requeridos na interface, usar termos consistentes, específicos, descritivos, distintivos entre si e familiares para o usuário. 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. O uso de termos específicos para os comandos informa melhor sobre seu conteúdo, se tornando inequívoco.
- 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”. Deve-se evitar a apresentação de informações desnecessárias para o usuário, porque além de aumentar a demora na escolha de uma opção, preenche a memória com informações irrelevantes.
- 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.
- 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.
- Com relação à abreviatura, adotar estratégia clara para abreviar e mantê-la em toda a interface. Existem as seguintes estratégias de abreviatura: Truncagem simples: usar da primeira até a *n-ésima* letra que não coincidam com outras palavras já truncadas. Por exemplo telef (telefone) e teleg (telégrafo). Corte de vogais com truncagem: eliminar as vogais da palavra e utilizar a truncagem, por exemplo: prgs. (programas). Primeira e última letras: abreviar uma palavra utilizando somente a primeira e última letras, exemplo: ST (*sort*). Abreviações de outros contextos, por exemplo: BAK (*backup*), PC (*personal computer*). Através do som produzido pela pronúncia da palavra, exemplo: CTRL (*control*).

- Usar corretamente o recurso da abreviatura. Dentro do sistema, as abreviaturas devem seguir uma padrão. As coincidências nas abreviaturas decorrentes do uso de um padrão devem ser resolvidas mediante uma regra específica para tratar estes conflitos ou abreviaturas iguais.
- Quanto às tabelas, usar o mínimo de colunas possível. Como já vista, o limite aceitável é de quatro colunas com doze linhas por tabela.
- 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.

3.9.6 RECOMENDAÇÕES PARA SAÍDA DE INFORMAÇÕES

Estas recomendações consideram aspectos relativos ao modo do sistema interativo retornar as informações que foram requisitadas por um possível usuário. Em softwares interativos é muito importante saber como dar as respostas e em que tempo.

- 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 na interação do que dar uma mensagem mostrando que ele tem um 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.
- Fornecer uma quantidade de respostas compatíveis com a capacidade do usuário. Para (SHNEIDERMAN 1987 *apud* [REI1999]) a capacidade de absorção do usuário está limitada a grupos de cinco à nove informações. Então, é importante considerar este limite quando o sistema dá respostas ao usuário, ou seja, não adianta dar quinze respostas simultâneas na tela para o usuário se o mesmo só irá absorver nove delas, num dado momento.
- Para tempos longos de resposta, orientar o usuário com informações na tela. Algumas maneiras de informar o usuário: Mostrar o tempo estimado que levará o processamento requerido. Elementos gráficos com movimento real para mostrar o que o sistema está fazendo e a porcentagem de processamento efetuada.

Movimento ilusório, visto anteriormente, pode servir para informar o que o sistema está fazendo, num processamento demorado.

- Em se tratando de tempo de resposta ao usuário, respeitar as características do mesmo neste tempo. O tempo de resposta aceito de maneira confortável pelo usuário depende de suas características pessoais, pelo nível de experiência, pelo tipo de atividade a ser realizada, etc. Deve-se considerar esses fatores na determinação de tempos de resposta adequada para quem usa o software. A prototipação é uma maneira de adequar os tempos de respostas aos usuários potenciais.
- 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 costumar com os tempos de resposta das ações efetuadas no software, frustrando as expectativas de espera pelo processamento.

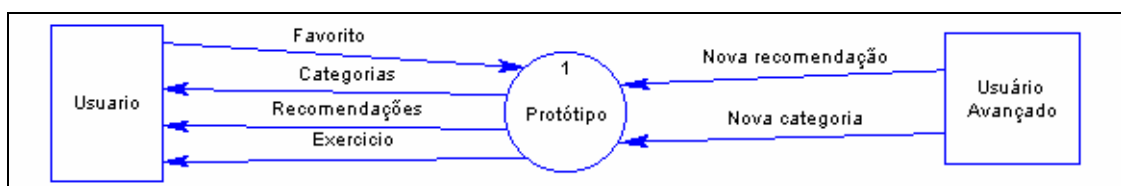
4 DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo, será descrito a especificação, detalhes da implementação, linguagem e ferramentas utilizadas, além da funcionalidade do protótipo.

4.1 ESPECIFICAÇÃO DO PROTÓTIPO

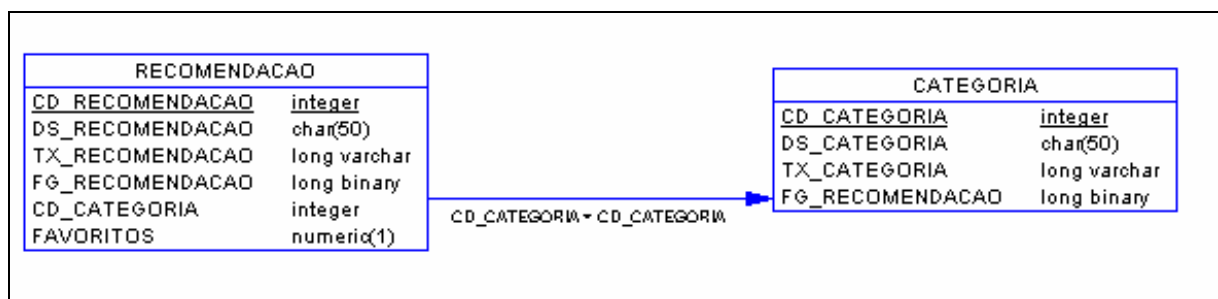
O Diagrama de Contexto é uma representação gráfica do sistema como um todo e conforme o sistema esta assim definido (ver Figura 2).

Figura 2 - Diagrama de Contexto



A Figura 3 mostra o Modelo de Entidade Relacionamento (MER) conforme utilizado no desenvolvimento do protótipo.

Figura 3 - MER



Os Quadros 2 e 3 respectivamente mostram os atributos das tabelas recomendação e de categoria.

Quadro 2 - Tabela Recomendação

| Atributo | Tipo | Descrição |
|-----------------|---------------------|--------------------------------|
| Cd_recomendação | Integer | Código da recomendação |
| Ds_recomendação | Characters | Descrição da recomendação |
| Tx_texto | Arquivo texto (rtf) | Texto da recomendação |
| Fg_figura | Bitmap (bmp) | Imagem exemplo recomendação |
| Favoritos | Boolean | Se está presente nos favoritos |

Quadro 3 - Tabela Categoria

| Atributo | Tipo | Descrição |
|-----------------|---------------------|--------------------------|
| Cd_categoria | Integer | Código da categoria |
| Ds_categoria | Characters | Descrição da categoria |
| Tx_texto | Arquivo texto (rtf) | Texto da Categoria |
| Fg_figura | Bitmap (bmp) | Imagem exemplo Categoria |

4.2 IMPLEMENTAÇÃO DO PROTÓTIPO

Para implementação do Protótipo foi utilizada a ferramenta *Borland Delphi 5*, as tabelas do banco de dados foram feitas no *Database Desktop (Paradox)* e também foi utilizado o *BDE Administrator* para criação de um alias para facilitar o trabalho com os dados que estarão armazenados nas tabelas.

O protótipo é formado por seis formulários, o principal, o formulário para cadastro e alteração de categoria, outro para cadastro e alteração de recomendações, o *Data Module* onde estão os componentes do banco de dados, outro que contém o exercício e finalmente o formulário de informações (sobre).

No formulário principal foram utilizados `panels` para organizar os componentes no formulário, botões do tipo `speedbutton`, o `TImage` para mostrar a figura exemplo, o `richedit` para mostrar textos, um `mainmenu` e dois `pmenu` que são utilizados no `click` com o botão direito do `mouse`. Tem-se ainda o componente `Ttreeview`, o qual representa uma janela que exibe uma lista hierárquica de itens (árvore), como os títulos em um documento, as entradas em um índice, ou os arquivos e diretórios em um disco.

No protótipo a árvore possui três níveis, sendo o primeiro apenas usado como título da árvore, o segundo é usado para apresentar as categorias contidas na respectiva tabela, e finalmente o terceiro nível apresenta as informações da tabela de recomendações. Cada nodo da árvore esta associado ao `TImage` e ao `Richedit` (ver Quadro 4).

Quadro 4 - Associação do nodo aos componentes

```

. . .
case Node.Level of
  1:begin
    if tbCategoria.FindKey([Apont(Node.Data)^]) then
      begin
        quadro_texto.Lines.Assign(tbCategoriaTx_texto);
        imageE.Picture.Assign(tbCategoriaFg_figura);
      end
    else
      Raise Exception.Create('Cetegoria não Cadastrada');
    end;
  . . .

```

É utilizado outro `TreeView` para exibir os itens que estão inseridos nos favoritos. Quando uma categoria ou recomendação é inserida nos favoritos, é alterado o conteúdo do atributo favoritos da tabela recomendação de falso para verdadeiro, sendo que a árvore dos favoritos fica oculta atrás da árvore normal, e só fica visível quando é solicitada.

Já quando é feito um cadastro de uma nova categoria ou recomendação em tempo de execução, usa-se a rotina de inserção de um novo nodo na árvore (ver Quadro 5).

Quadro 5 - Adicionar Nodo

```
procedure TFrmPrincipal.AddNodo;  
{adiciona um nodo na arvore}  
var  
    aux:apont;  
begin  
    new(aux);  
    aux^:= codigo;  
    TreeView1.Items.AddChildObject(pai, nome, aux);  
end;
```

Os botões anterior e próximo guardam o caminho percorrido na árvore através do tipo *Tlist* do *Delphi* (lista dinâmica). Sendo que os botões tem a mesma função quando esta sendo exibida a árvore dos favoritos.

No formulário de cadastro de categoria como diferenciais foram usados *DBEdit* e um *openpicturedialog*.

Em relação ao formulário de categorias, no formulário de recomendações foi acrescentado um *TDBLookupComboBox* para que fosse possível escolher dentro de qual categoria a recomendação estaria cadastrada.

4.3 FUNCIONAMENTO DO PROTÓTIPO

Ao ser iniciado, o protótipo apresenta ao usuário sua janela principal (ver Figura 4), que é composta de um menu principal, uma barra de ferramentas e a parte visual do protótipo que contém a árvore com os itens relacionados em ordem hierárquica, um quadro com a descrição do nodo a ele associado e finalmente a imagem exemplo ou descritiva também associada ao relativo nodo.

Figura 4 - Janela Principal



O protótipo possui um recurso chamado favoritos, que funciona de modo semelhante a este recurso em um *browser*, com a diferença que no protótipo guarda-se uma recomendação e no *browser*, guarda-se um endereço. Quando o usuário solicita a exibição dos favoritos, a árvore normal torna-se oculta, tornando-se visível outra árvore, na qual é mostrada apenas os itens adicionados aos favoritos.

No menu principal (ver Figura 5) tem-se os itens categoria e recomendação, nos quais as opções de cadastrar nova, para cadastrar uma nova categoria ou recomendação, editar, para alterar dados já cadastrados, e remover para remover um dado cadastrado. O item Favoritos tem as opções de adicionar aos favoritos e remover dos favoritos. O item exercício leva a uma tela onde tem-se uma interface com alguns erros para serem detectados pelo usuário, funcionando como um exercício de fixação dos conhecimentos abordados. Os demais itens, sair, encerra a execução do protótipo e o sobre é apenas descritivo, apresenta informação sobre o protótipo.

Figura 5 - Menu Principal



Existem opções para o *click* direito do *mouse* nos nodos da árvore, quando esta sendo exibida a árvore normal pode-se excluir, editar ou adicionar aos favoritos, já quando esta sendo exibido a árvore dos favoritos tem-se a apenas a opção de retirar dos favoritos, pois nos favoritos não é permitido alterar o conteúdo dos dados.

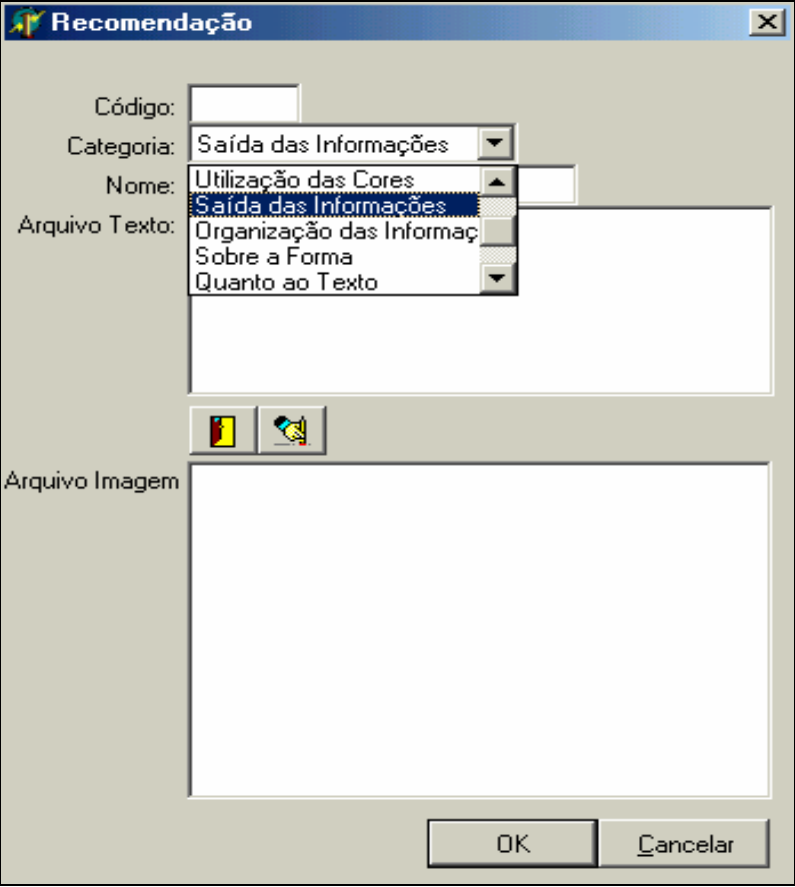
O cadastro de uma nova categoria é feito pelo usuário através do formulário de cadastro, este é carregado quando o usuário o chama através do botão ou do item no menu detalhado acima (ver Figura 6).

Figura 6 - Formulário de cadastro de categoria

The image shows a dialog box titled "Categoria". It contains the following fields: "Código:" with a text box containing the number "3"; "Nome:" with an empty text box; "Arquivo Texto:" with a large empty text area; and "Arquivo Imagem:" with a large empty text area. Above the "Arquivo Imagem:" field are two small icons: a folder and a document. At the bottom of the dialog are "OK" and "Cancelar" buttons.

Para cadastrar uma nova recomendação é necessário que o usuário defina em qual categoria a recomendação estará inserida (ver Figura 7), quanto aos dados necessários, são os mesmos de uma categoria, um figura e uma descrição.

Figura 7 - Formulário para cadastro de recomendação



O formulário, intitulado "Recomendação", possui os seguintes campos e elementos:

- Código:** Campo de texto vazio.
- Categoria:** Menu suspenso com o valor "Saída das Informações".
- Nome:** Campo de texto com o valor "Utilização das Cores".
- Arquivo Texto:** Menu suspenso com o valor "Saída das Informações". O menu está aberto, mostrando as opções: "Utilização das Cores", "Saída das Informações" (destacada), "Organização das Informaç", "Sobre a Forma" e "Quanto ao Texto".
- Arquivo Imagem:** Área com ícones de arquivos de imagem e um campo de texto vazio.
- Botões:** "OK" e "Cancelar" na base direita.

Para editar (alterar os dados) tanto as recomendações quanto as categorias também são usados os formulários de cadastros, o qual abre com as informações do item que se deseja alterar, podendo ser mudado tanto o nome quanto a foto e a descrição.

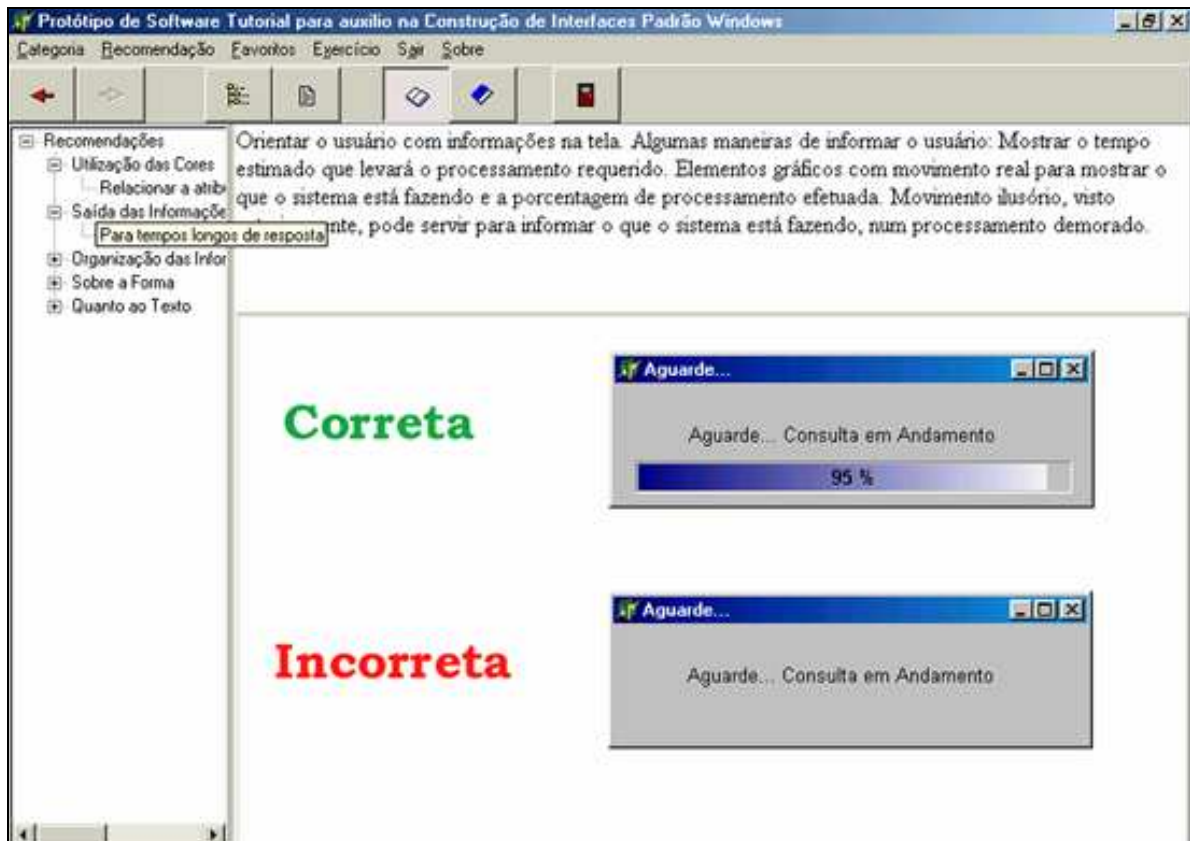
A Barra de ferramentas (Figura 8), mostra em ordem da esquerda para a direita os botões de navegação, que guardam o caminho percorrido na árvore. O botão para inserir nova categoria, seguido pelo de nova recomendação, seguido dos botões de visualização, um para mostrar arvore normal e outro para exibir favoritos, e finalmente um para fechar o protótipo.

Figura 8 - Barra de ferramentas



A Figura 9 mostra como é exibida uma recomendação no protótipo.

Figura 9 - Visualizar Recomendação



A Figura 10 mostra a tela do exercício onde tem-se uma interface exemplo com um determinado número de erros segundo as recomendações estudadas e o usuário com um duplo *click* ou um *click* simples desabilita o componente que contém o erro. O exercício é fixo, ou seja todas as vezes que for executado apresenta os mesmos erros.

Figura 10 – Tela Exercício

Exercício

Existem 3 erros na Interface Faltam 1

Cadastro de Clientes

Insira o nome do cliente: Fulano de tal

Endereço: rua tal

Telefone: 222 6677

E-mail: 222 6677

Cidade: Longe de Tudo

Estado: Santa Catarina - SC

Cancelar OK Para Incluir Cliente Click OK

Finalmente no item do menu sobre tem-se informações descritivas sobre o protótipo (ver Figura 11).

Figura 11 – Tela Sobre



5 CONCLUSÃO

Neste capítulo serão apresentadas conclusões em relação ao trabalho desenvolvido e sugestões para trabalhos futuros.

5.1 CONSIDERAÇÕES FINAIS

Não existe nenhuma regra pré-definida sobre como construir uma interface, o que existem são recomendações, baseadas em aspectos ergonômicos e princípios de *design*, que servem como orientação para o desenvolvimento de uma interface que cada vez mais possa simplificar a interação entre usuário e software.

O protótipo cumpriu com o objetivo proposto, que era a partir do estudo das recomendações viabilizar a implementação de um protótipo de uma ferramenta de auxílio a programadores, com exemplos gráficos de erros e acertos em interfaces padrão windows.

A principal característica do protótipo é a forma dinâmica como os dados podem ser administrados. O usuário tanto pode acrescentar novos dados (categorias ou recomendações) como alterar os dados armazenados.

5.2 EXTENSÕES

Como extensões para trabalhos futuros pode-se sugerir o aumento de exemplos e de exercícios de reconhecimentos de erros em interfaces e conseqüentemente tendo-se mais interatividade com o usuário.

O estudo de formas alternativas de avaliar o conhecimento adquirido pelo usuário com o uso de um tutorial.

Estudar maneiras de explorar uma forma de tutorial que incorpore um desafio para o usuário, como um jogo, por exemplo.

REFERÊNCIAS BIBLIOGRÁFICAS

- [HEC1993] HECKEL, Paul. **Software Amigável: técnicas de projeto de software para uma melhor interface com o usuário**. Trad. de Insight Serviços de Informática. Rio de Janeiro: Campus, 1993.
- [JUN1995] JUNIOR, D. C. C. Helio. **Tutorial de recomendações ergonômicas aplicadas à implementação de software**. Trabalho de Conclusão de Curso, FURB, Blumenau, 1995.
- [MIN1994] MINASI, M. **Segredo de projeto de interface com o usuário**. Rio de Janeiro; Infobook, 1994.
- [NIK1994] NIKEL, Fábio. **A demanda de informação para o desenvolvimento de sistemas com interfaces ergonômicas**. Trabalho de Conclusão de Curso, FURB, Blumenau, 1994.
- [ODE1994] ODEBRECHT, C. **Análise ergonômica da interface software usuário: um estudo de caso do processador de textos Fácil/W com usuários novatos**. Dissertação de Mestrado. UFSC, Florianópolis, 1994.
- [PRE1995] PRESSMAN, R. S. **Engenharia de software**. São Paulo: Makron Books, 1995.
- [REI1999] REIS, Dalton S. **Tópico 3 - Sistemas gráficos: hardware e software**. Endereço eletrônico: www.ipa.furb.rct-sc.br/dalton/DiscipCG/ModuloA/moAtop03.htm. Data da consulta: 01-08-2000.
- [RIG1993] RIGHI, C.A.R. **Aplicações de recomendações ergonômicas para o componente de apresentação da interface de softwares interativos**. Dissertação de Mestrado. Florianópolis, 1994.

- [STE1994] STEIN, E. Sandra. **Ergonomia: auxiliando na interface homem-máquina através da utilização de ferramentas.** Trabalho de Conclusão de Curso, FURB, Blumenau, 1994.