

XIV Conferencia Latinoamericana de Informática
17avas. Jornadas Argentinas de Informática
e Investigación Operativa.
Buenos Aires, Setiembre 1988.

FERRAMENTAS PARA EDIÇÃO DE TEXTO
EM UM
AMBIENTE OPERACIONAL AMIGÁVEL

- Ana Carla Bon

COBRA - Computadores e Sistemas Brasileiros S/A.
Av. Comandante Guarannys, 447
Jacarepaguá
Cep: 22785 - Rio de Janeiro - RJ
Brasil

- Luiz Carlos Montez Monte

Universidade Federal do Rio de Janeiro
Cidade Universitária - Centro de Tecnologia
COPPE - Engenharia de Sistemas e Computadores
Bloco H - Sala 319
Ilha do Fundão
Cep: 21945 - Rio de Janeiro - RJ
Brasil
Telefone: 280-8832 - Ramal: 419

RESUMO

Esse trabalho desenvolveu o EdiTex, que é um conjunto de ferramentas de edição de texto para um ambiente operacional amigável, contruídos a partir da identificação do texto como objeto fundamental de interface com o usuário.

Essas ferramentas são um subconjunto do projeto FIU (14) - Ferramentas de Interface com o Usuário - desenvolvido na Cobra Computadores S.A., resultado da proposta de se suprir a carência de um ambiente amigável no SOX (15), que é um sistema operacional compatível com o Unix.

Uma das principais características do FIU é, através do EdiTex, fornecer edição de texto sofisticada embutida no sistema, disponível a qualquer aplicação. O Editex provê amplos recursos para que sejam atendidas de forma abrangente, as diversas necessidades de edição e exibição de texto na interface homem-máquina, como: a criação de diálogos textuais, a exibição de legendas e o desenvolvimento de aplicações voltadas exclusivamente ao processamento de texto.

Inédito no Brasil, o projeto FIU baseou-se em extensa pesquisa sobre o estado da arte, tanto na área de fatores humanos, quanto em Engenharia de Software, utilizando no seu desenvolvimento metodologias de projeto de comunicação com o usuário e definindo sua arquitetura orientada a objetos.

I. INTRODUÇÃO

Texto é, na maioria das vezes, indispensável na interação com o usuário. Algumas aplicações, tais como processadores de palavras, são especializadas em texto; outras, como planilhas ou aplicativos de exibição de relatórios, não são especialistas em texto, mas utilizam-no de forma intensa. Até mesmo aplicações gráficas utilizam texto, embora de forma esporádica. De forma geral, no entanto, as aplicações necessitam fornecer e obter informações textuais dos usuários.

Para o usuário é imprescindível que o texto seja editado da mesma maneira, não importando em que aplicação ou onde ele apareça. Por outro lado, para o programador de aplicações é útil que existam ferramentas dedicadas à exibição e edição de texto.

A implementação do conjunto de ferramentas de edição de texto, EdiTex, teve como objetivo prover recursos para atender as mais diversas necessidades de edição e exibição de texto dentro de um ambiente operacional amigável.

Pretende-se que toda aplicação, desenvolvida para o SOX, que utilize, de alguma forma, texto (seja na mais básica exibição de informação, seja criando softwares sofisticados de processamento de texto) obtenha os principais recursos necessários para sua implementação no que se refere a texto, dentro do conjunto de ferramentas especificadas.

O objetivo do EdiTex, portanto, não é implementar aplicações como editores de texto, mas, entre outros, permitir a redução do custo do desenvolvimento dessas aplicações e garantir uma interface com o usuário consistente com o ambiente no qual estão inseridas.

II. DESCRIÇÃO GERAL

II.1 - Metodologias Empregadas

Grandes esforços têm sido realizados, tanto na área de Fatores

Humanos quanto em Engenharia de Software, no sentido de oferecer critérios e métodos a serem seguidos na especificação de projetos de interfaces interativas.

Uma das propostas desse trabalho foi elaborar um desenvolvimento orientado à qualidade do produto, utilizando-se conhecimentos adquiridos dessas áreas.

O desenvolvimento do EdiTex foi realizado em três fases: a fase da Aquisição e Definição de Requisitos, a Fase do Projeto e a Implementação (fig II.1).

Segundo o grupo de desenvolvimento do Star (44), o primeiro passo a ser dado antes do projeto bem sucedido de qualquer interface com o usuário deve ser a "Definição de Requisitos", também denominada de "Análise das Tarefas". Esse passo envolve a análise da área da aplicação e de seus usuários, estabelecendo "quem são os seus usuários, quais os seus objetivos em realizar determinadas tarefas, quais informações eles geram e que métodos empregam".

A fase de Definição dos Requisitos é, em resumo, um levantamento de dados, que é concluído com a elaboração de um documento denominado Especificação de Requisitos de Software (ERS), utilizado como base para as decisões nas fases seguintes de projeto e implementação do software.

Após a fase de Definição de Requisitos, iniciou-se a fase de Projeto que foi dividida em três subfases: o Projeto Externo, no qual foram definidas as Interfaces do EdiTex, isto é, interface com o usuário-operador e com o usuário-programador; o Projeto Arquitetural onde foi realizado a decomposição do EdiTex em módulos; e o Projeto Detalhado, no qual foram elaborados os algoritmos e definidas as estruturas de cada módulo.

Finalmente, a última fase do desenvolvimento do projeto foi a implementação na linguagem de programação "C" do que foi estabelecido nas fases anteriores. A figura II.1 mostra um diagrama do ciclo de desenvolvimento do projeto, empregado conforme des-

critico acima.

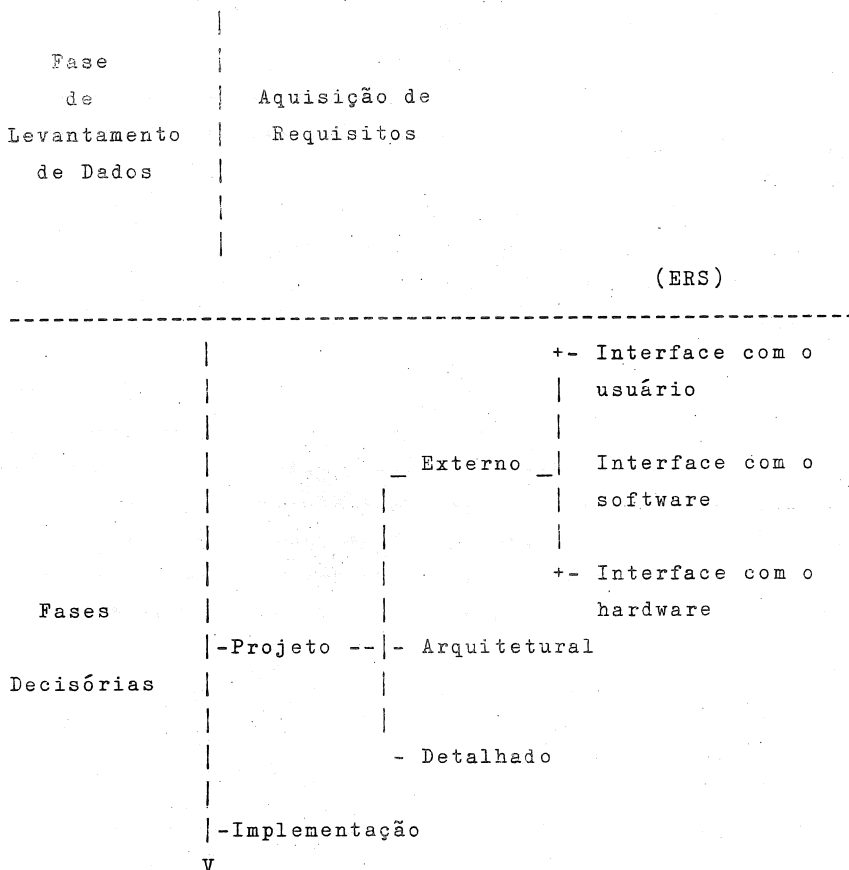


fig II.1 : Fases do Projeto EdiTex

Duas metodologias foram empregadas na realização dessas fase: a Metodologia de Projeto Orientada a Objetos e a Metodologia de Especificação de Interfaces Interativas.

A Metodologia Orientada a Objetos permite que, num dado nível da solução do projeto, seja identificado os objetos e definidas as operações apropriadas a esses objetos. Embora mais dirigida à fase de Projeto Arquitetural e Detalhado, pois como definiu BOOCH (10), ela é uma metodologia que "fornece critérios para a decomposição de um sistema em módulos", ela foi aplicada desde as primeiras fases do desenvolvimento do EdiTex: as entidades

do mundo real e os comportamentos dos usuários sobre essas entidades foram mapeadas em objetos e operações sobre esses objetos.

A abordagem orientada a objetos atrasa, ao máximo possível, o surgimento de processos e dados (só presente na fase de implementação), fazendo com que o mapeamento do mundo real para os objetos existentes nas especificações e nos projetos seja imediato. A aplicação do uso dessa metodologia torna mais fácil a manutenção e alteração do sistema.

Na elaboração do Projeto Externo foi utilizada a Metodologia de Especificação para Interfaces Interativas como descrita por FOLEY et alii (20), que propõem a descrição da linguagem de interação com o usuário-operador em quatro passos: conceitual, semântico detalhado, sintático e léxico.

A vantagem dessa abordagem "top-down", é que durante a especificação da linguagem de interação, é possível abstrair-se, em um dado passo, das decisões que só precisam ser tomadas nos passos seguintes. Por exemplo, durante a definição da semântica da linguagem de interface com o usuário, não é necessário considerar características relacionadas a dispositivos de hardware, que só afetam o projeto léxico.

II.2 - Perspectiva do Sistema

II.2.1 - Ambiente do Editex

O Editex é um subconjunto do FIU - Ferramentas de Interface com o Usuário. O FIU é "a parte do SOX responsável pela comunicação de dados entre os usuários e os processos (aplicações)"(COBRA(14)).

A interface FIU-aplicação é realizada por meio de chamadas e retornos de funções do FIU e somente utilizando essas funções é que a aplicação se comunica com o usuário.

Cada usuário envia dados e controles para o FIU pelo teclado ou

pelo mouse do seu terminal e recebe informações pela tela desse terminal .

Além de fornecer um conjunto de ferramentas para a programação da interface com o usuário dos aplicativos, o FIU "induz a uma padronização dessas interfaces" propiciando a construção de um ambiente amigável no SOX.

Entende-se por amigável, um software de fácil uso, cuja principal característica é comunicar-se bem (HECKEL (24)). Um conjunto de aplicações amigáveis, integradas entre si a partir de uma consistência nas suas interfaces com o usuário, formam um ambiente amigável.

Para que um ambiente seja amigável, é indispensável que exista um aplicativo responsável pela administração do sistema, cuja função seja permitir ao usuário a execução fácil de suas tarefas e a utilização de todos os recursos da máquina. Não é possível pensar em ambiente amigável nos sistemas compatíveis com o Unix, enquanto a comunicação usuário-sistema for realizada por meio da Shell.

Além disso, devem ser oferecidas ferramentas de propósito geral que visem atender às necessidades de usuários distintos. Um conjunto mínimo dessas ferramentas deve conter um processador de texto, um processador gráfico e alguns utilitários acessórios a outras tarefas como uma máquina de calcular e um bloco de rascunho.

II.2.2 - FIU

O FIU tem como principal característica "promover a utilização de técnicas avançadas de comunicação com o usuário, independentemente da existência de facilidades gráficas" (Monte(36)). Embora atualmente o FIU esteja sendo utilizado em um ambiente semi-gráfico, poucas alterações serão necessárias para evolução a um ambiente gráfico.

A utilização do mouse é opcional, podendo o operador utilizar

somente o teclado. As funções do FIU manipulam principalmente, texto, janelas, diálogos, cardápios e predefinidos. A seguir estarão descritos os significados destes componentes do FIU (COBRA (14)).

- Janelas

Visualmente uma janela é uma área retangular da tela na qual são exibidas informações e que pode sobrepor total ou parcialmente, ou ser sobreposta por outras janelas. Funcionalmente, uma janela é o meio pelo qual uma aplicação transmite e recebe dados do usuário.

A utilização de janelas sobrepostas no FIU baseou-se no conceito criado pelo Smalltalk-80 (22) e utilizados em diversos sistemas de interface surgidos posteriormente como o Star (44), Lisa (48), Macintosh (2) e o Amiga (46). Uma única aplicação, ou aplicações distintas, podem possuir simultaneamente um número qualquer de janelas na tela. Porém, em um dado instante, apenas uma janela pode estar selecionada, significando que somente essa janela está apta para receber dados do usuário. A facilidade de aplicações diferentes poderem possuir janelas na tela simultaneamente, está presente em vários sistemas de interface como o MS-Windows (34), X-Windows (35), Unix PC (4) e o Amiga. O Macintosh, não permite essa facilidade.

Existem vários tipos padronizados de janelas, cada um sendo adequado ao tipo de informação tratada, existindo desde janelas para exibição de mensagens de erro até janelas sofisticadas para edição de texto.

Cada janela tem, associada a ela, um ambiente de desenho, que é o conjunto completo das informações necessárias para as operações de desenho da interface com o usuário. Um ambiente de desenho contém, entre outras coisas: as dimensões de uma janela, sua posição na tela, o padrão de preenchimento da tela, os tipos e estilos de caracteres no texto. Utilizando conceitos gráficos como sistemas de coordenadas, um ambiente de desenho permite que todos os desenhos e cálculos de uma janela sejam fei-

tos baseados no próprio sistema de coordenadas locais dessa janela. Também são utilizadas técnicas gráficas de "clipping" (recorte) para determinar as regiões de uma janela que precisam ser redesenhadas num dado momento.

- Caixas de Diálogo

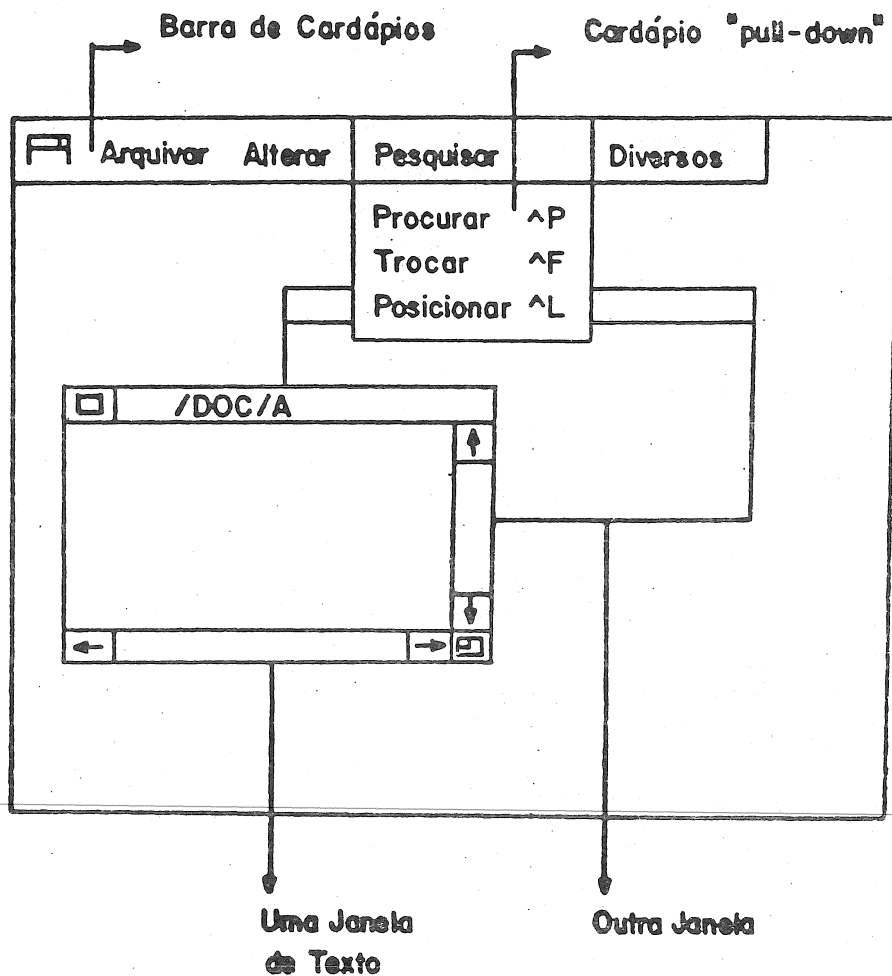
Uma Caixa de Diálogo é um tipo de janela utilizado para que uma aplicação obtenha informações adicionais do usuário a fim de poder executar um determinado comando. Caixas de Diálogo também são utilizadas para fornecer informações importantes, tais como mensagens de erro ou o acompanhamento de uma tarefa.

A interação com o usuário realizada por meio de uma Caixa de Diálogo denomina-se diálogo. Existem três tipos de diálogo, que diferem quanto ao nível de restrição imposta às ações do usuário: Diálogos Imodais - não restringem qualquer ação do usuário, permitindo-o interagir com outras janelas enquanto a Caixa de Diálogo permanece na tela; Diálogos Modais - limitam a ação do usuário a responder o diálogo ou descartá-lo, não permitindo qualquer interação com outras janelas; e Alerta - que só permite ser descartado, uma vez que não é pedida nenhuma informação ao usuário. Diálogos modais e imodais são utilizados para coletar informações do usuário, enquanto Alertas são utilizados para informar algo ao usuário como, por exemplo, mensagens de erro.

Dentro de uma Caixa de Diálogo, existem dispositivos padronizados por meio dos quais o usuário fornece dados à aplicação. São eles: Caixas de Texto, para o usuário preencher dados textuais; Caixas de Marcas para o usuário determinar opções não exclusivas; Botões de Rádio, para o usuário escolher uma opção dentre um conjunto de opções mutuamente exclusivas; e Teclas de Luz, para comandar ações.

- Cardápios

Um cardápio é uma lista de opções da qual o usuário pode reali-



FIU: A Tela de um Aplicativo

zar uma escolha. O FIU determina a existência de uma barra de cardápios sempre visível, posicionada na primeira linha da tela. Essa barra de cardápios, como no Macintosh, contém títulos de cardápios, que ao serem selecionados, desenrolam a lista de opções daquele cardápio ("pull-down menu"). A barra de cardápios possibilita ao usuário listar, a qualquer momento, todos os comandos de uma determinada aplicação e verificar quais estão disponíveis para serem executados.

- Predefinidos

Predefinido "é uma descrição de entidade previamente gerada e mantida em um arquivo sob controle do FIU" (COBRA(14)). O FIU utiliza predefinidos para armazenar formas de janelas, cardápios, caixas de diálogo, entre outras.

Predefinidos propiciam a separação entre o código da aplicação e o conjunto de dados que constituem a interface com o usuário. Isso permite que a aplicação experimente diferentes formatos de informação sem alterar o código do seu programa.

Arquivos de dados também podem estar associados a predefinidos e isso tem implicação importante no desenvolvimento de processadores de texto porque permite que o texto seja gravado, associado aos seus dados de formatação. O FIU fornece um compilador de predefinidos.

II.3 - Usuários do Editex

A palavra "usuário" tem sido empregada na redação desse trabalho com o único sentido de caracterizar as pessoas que operam o computador. No entanto, existe uma outra classe distinta de usuários do Editex.

Esta classe é formada pelos projetistas e programadores de interfaces com o usuário. De uma forma geral, o conjunto das ferramentas de edição deve ser apropriado ao desenvolvimento de qualquer aplicação que utilize texto.

A segunda classe é constituída pelos usuários operadores da máquina. Representantes dessa classe tanto podem ser profissionais leigos em computação, que utilizam um microcomputador pessoal como apoio às suas atividades, quanto profissionais da área de informática. Em ambos os casos, esses profissionais necessitam de formas eficientes de comunicação com o sistema, o que inclui ferramentas adequadas para a edição de texto.

II.4 - Visão Global

Essa seção fornece uma visão global da funcionalidade desejada para o Editex. O Editex - conjunto de ferramentas para edição e exibição de texto - deve fornecer ferramentas para a programação das partes textuais das interfaces com o usuário das aplicações e gerenciar as ações dos usuários sobre essas partes, durante a execução das aplicações. Estas ferramentas devem servir para a manipulação (tanto do programador quanto do operador) de três classes de objetos textuais: linhas de texto não editáveis, campos de texto e janelas especializadas em texto.

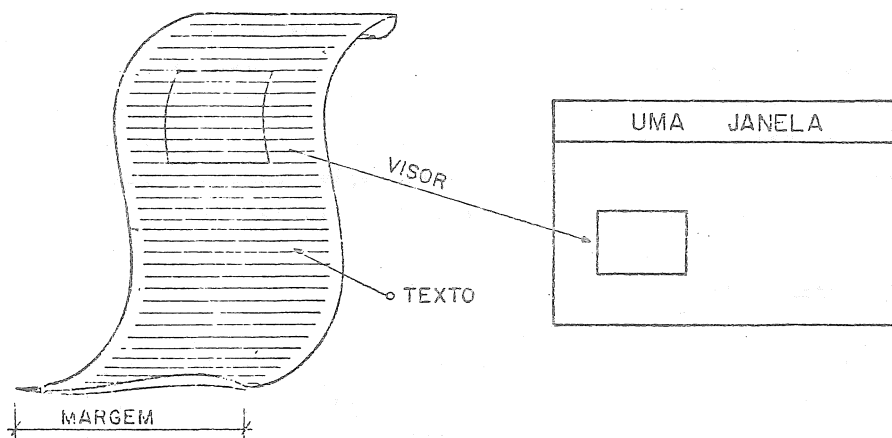
1 - Linha de texto não editável

Uma linha de texto não editável é constituída por uma sequência de um ou mais caracteres. Linhas de texto podem ser exibidas na tela, ou seja, os caracteres que compõem a linha podem ser desenhados na tela de acordo com determinadas características de desenho.

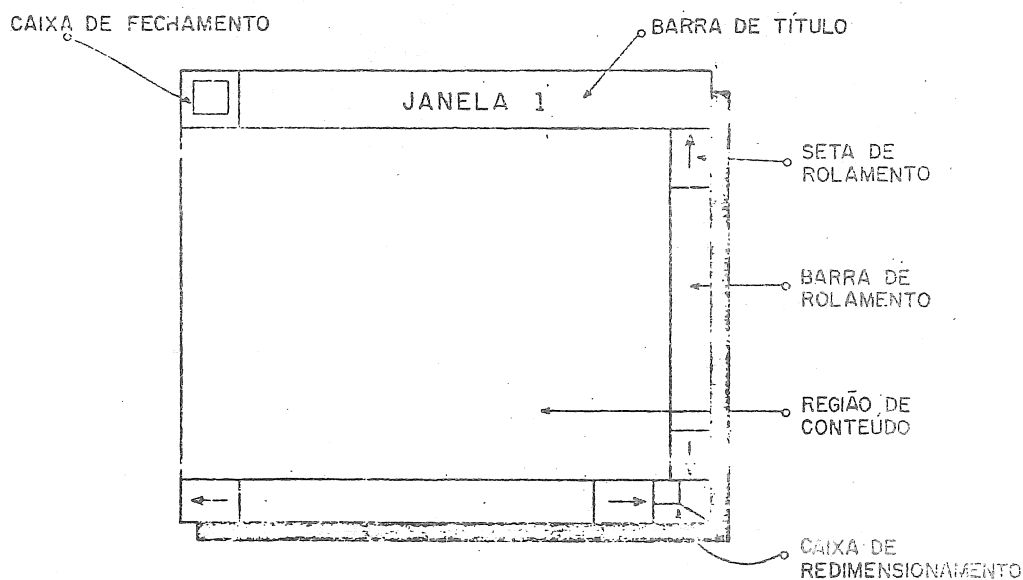
2 - Campos de Texto

Um Campo de Texto é um texto exibido formatado e que pode ser visualizado totalmente ou parcialmente em uma janela na tela através de um campo retangular denominado visor. Operações sobre o texto do Campos de Texto podem ser divididas em dois grandes grupos:

- operações de alteração de conteúdo do texto, tais como inserir novos caracteres ou copiar um trecho do texto para outra parte do mesmo texto.
- operações de definição da formatação do texto, tais como:



Objeto campos de texto



Objeto janela de texto

margens, alinhamento às margens, tabulação, estilo e tamanho dos caracteres.

3 - Janelas especializadas em texto

Uma Janela especializada em texto é um tipo de janela cuja região de conteúdo exibe um determinado texto. Operações possíveis de serem realizadas em uma janela especializada de texto incluem:

- todas as operações inerentes a janelas no FIU, tais como movimentação e redimensionamento da janela na tela;
- todas as operações de alteração e formatação do texto;
- operações específicas de janela de texto tais como: controle da visualização do texto dentro da janela e subdivisão da janela.

II.5. - Atributos Desejáveis

Com o objetivo de se obter uma boa interface com o usuário, foram definidos previamente os atributos que o sistema deveria atender. Os atributos selecionados são os propostos pela literatura pesquisada em Interfaces com o Usuário, Fatores Humanos e Processamento de Texto.

II.5.1 - Consistência:

Esse atributo determina, de uma forma geral, que qualquer software escrito para a mesma máquina deve possuir a mesma aparência, isto é, que o usuário não precise, para cada nova aplicação utilizada, reaprender uma nova interface (APPLE (2)). Além disso, o conhecimento que o operador adquire de uma parte do sistema o ajuda em outras partes (MARCUS (31), SIMPSON (42)).

Em particular, isso significa que o usuário deve ter a mesma maneira de editar um texto, seja qual for a aplicação que ele esteja utilizando. O EdiTex oferece facilidades das mais simples às mais sofisticadas para exibir e editar texto. Uma aplicação que não seja voltada para texto e que não necessariamente utilize todas essas facilidades, deve garantir que o conjunto

mínimo oferecido seja compatível com as do conjunto completo. Por exemplo, o usuário deve ter a mesma maneira de retirar do texto o último caractere digitado; porém, não necessariamente poderá em todas as aplicações selecionar palavras.

II.5.2 - Capacidade de fornecer respostas rápidas

A capacidade de fornecer respostas rápidas significa que resultados imediatos devem estar associados às ações do usuário para que ele possa avaliar a ação realizada e seguir em frente (APPLE (2)).

BERTINO (9) diz que é de "vital importância que uma interface seja projetada de forma que o usuário nunca fique pensando se o seu comando (ou dado) enviado, foi recebido". Ele classifica dois tipos de realimentação necessárias em um sistema de interface. O primeiro tipo fornece a realimentação das ações do usuário e incluem eco, destaque na tela de objetos e cartões selecionados, movimentos do cursor e mensagens de erro. O segundo tipo fornece realimentação do progresso das ações do sistema, especialmente operações que consumam muito tempo, exibindo alguma indicação de que o que foi pedido está sendo realizado.

EMBLEY(16) et alii citam tempos de respostas aceitáveis para determinadas operações de edição: características de eco - menor do que 0.1 segundos ; pedidos conversacionais - menor do que 2 segundos ; procura página por página - menor do que 0.5 segundos para que apareçam as primeiras linhas da próxima página ; iniciar ou sair de edição - menor do que 15 segundos

II.5.3 - Permissividade

Significa que o sistema deve permitir ao usuário fazer qualquer coisa que seja razoável, isto é, que seja o usuário, e não o programa, quem determina o que deve ser feito. O sistema deve produzir ao usuário a sensação do domínio da situação e consequentemente de segurança em relação ao computador (APPLE (2)). Para HECKEL (24), uma interface deve ser permissiva; ele sugere

re: "dê o controle ao usuário".

Na edição de texto, o usuário deve ter flexibilidade para percorrer seu texto e selecionar parte dele sem se restringir a uma função particular ou às suas opções.

A melhor maneira de um sistema se tornar permissivo é evitando modos (TESLER(43), APPLE(2)). Um modo, ou um estado, pode ser definido como uma parte de uma aplicação onde o usuário deve, formalmente, entrar e sair e que, enquanto permanece ativo, restringe a realização de operações.

II.5.4 - Multimodalidade

A comunicação do sistema de interface com o usuário deve permitir interação em vários níveis de experiência do usuário de forma simultânea, aceitando comandos equivalentes com sintaxes de pouca complexidade e sintaxes mais complexas (APPLE(2), BERTINO(9), HECKEL(24)).

Isso permite que usuários pouco experientes consigam realizar as mesmas tarefas que um usuário experiente, tendo apenas que executar mais ações para obter o resultado esperado. Por outro lado, os usuários já familiarizados com o sistema podem executar suas ações mais rapidamente.

II.5.5 - Indulgência

Esse atributo implica em que um sistema deve, além de minimizar as chances do usuário cometer erros, limitar as conseqüências dos erros e também fornecer maneiras rápidas e simples do usuário se recuperar de situações causadas por erros.

Diagnósticos de erros devem ser claros para evitar que o usuário fique fazendo tentativas para corrigir os erros. (BERTINO(9)). Na edição de texto, conforme citado por EMBLEY et alii(18), "mesmo para usuários especialistas, de 5 a 30 por cento do tempo real de edição podem ser geralmente atribuídos à erros e correção de erros".

II.5.6 - Fácil de Aprender e Usar

"Um sistema que não tenta aliviar o medo, apreensão e incerteza que os usuários sentem quando estão usando o sistema, merecerá muito pouco interesse" (BERTINO(9)).

Alguns princípios a serem seguidos para se alcançar um sistema que tenha uma interface fácil de aprender e usar são:

- Mantenha o sistema simples ao invés de complexo (SIMPSON(42), HECKEL(24)).
- Comandos para o sistema devem ser intuitivos, isto é, eles devem descrever de forma sucinta e clara a ação que eles executam. HECKEL(24) sugere também a comunicação por metáforas.
- Deve ser minimizada a demanda da memória humana (SIMPSON(40)). Se para executar uma função, forem necessários diversas informações o sistema deve se encarregar de requisitá-las ao usuário, não obrigando-o a memorizá-las.
- Devem ser mantidos os comandos mais freqüentes na forma mais simples; comandos não freqüentes podem ser complexos (EMBLEY et alii(18)).
- O usuário deve, a qualquer momento, poder visualizar todo o conjunto de funções que o sistema oferece.
- O sistema deve fornecer um mecanismo eficiente de ajuda que auxilie, de forma clara e rápida, o usuário a entender algum aspecto do sistema (HECKEL (24), BERTINO (9), MEYROWITZ et alii (32)).
- O usuário deve ter formas de desfazer ações, seja para experimentar outras soluções, seja para se recuperar de erros (BERTINO(9), MEYROWITZ et alii (32)).

III. CONCLUSÃO

A evolução da Informática que levou o homem a conviver intensamente no seu cotidiano com o computador, vem direcionando o desenvolvimento de software no sentido da definição de interfaces mais amigáveis. Passou-se a considerar conceitos sobre o comportamento humano e aplicá-los nos sistemas de interfaces de

forma a produzir no homem a sensação de segurança e domínio em relação ao computador.

Reconhecendo a comunicação textual como um dos elementos básicos de um sistema de interface com o usuário, desenvolveu-se um conjunto de ferramentas para edição e exibição de texto - o EdiTex.

Atualmente o EdiTex é amplamente utilizado pelos demais módulos do FIU, além de já estar inserido em um sistema produtivo, tendo sido utilizado no projeto de um editor de texto experimental e no Projeto SIR - Sistema Integrado Relacional (COBRA (15)). Ambos os projetos utilizam intensamente Janelas de Texto editáveis ou não: o editor permite que sejam criadas até oito janelas de texto editáveis na tela, cada uma contendo um texto com formatos totalmente diversos; o SIR utiliza Janelas de Texto para que o usuário edite consultas a bancos de dados e para exibir relatórios. A maioria das Caixas de Diálogos criadas por esses projetos, contém diversos dispositivos de Caixas de Texto.

A aplicação da metodologia orientada a objetos, em particular na fase de Projeto Interno, tem proporcionado diversos benefícios pois permite que melhorias e alterações realizadas ao longo do desenvolvimento das ferramentas fiquem sempre restritas ao módulo afetado, sem impactar os demais módulos que o utilizam.

O objetivo do EdiTex, de atender as mais diversas necessidades de edição de texto em um ambiente operacional amigável, vem sendo alcançado. Em especial, a validade da proposta de se oferecer uma automatização da edição através de uma janela especializada, já foi comprovada.

Comparando-se as facilidades de edição de texto oferecidas pelos sistemas de interfaces pesquisados e o EdiTex, verifica-se que esse é um dos que mais recursos oferece para o desenvolvimento de programas aplicativos que utilizem de alguma forma texto. O sistema de interface, até então mais completo em rela-

ção a texto era o Macintosh, que embora ofereça através do seu "Toolbox" um imenso recurso gráfico na edição de texto, não oferece facilidades tais como tabulação, alinhamento às duas margens, janelas automáticas, e só permite um único estilo e conjunto de caracteres no texto. A aplicação que desejar esse recursos tem que elas mesmas programarem, sem utilizar o "Toolbox". Outro sistema que oferece grandes facilidades de edição é o X-Windows, que pelas propostas apresentadas no X-Toolkit, contém também janelas especializadas em texto, mas não oferece muitas facilidades de formatação de texto.

Seguindo o modelo de edição do Smalltalk-80, baseado na imodalidade, o EdiTex consegue oferecer ao usuário-operador uma grande facilidade nas suas tarefas de edição, ao mesmo tempo que integrado ao restante do FIU, pode proporcionar a construção de um ambiente amigável para o SOX.

REFERÊNCIAS BIBLIOGRÁFICAS

- (1) APPLE COMPUTER INC., Mac Write, 1984.
- (2) APPLE COMPUTER INC., Inside Macintosh, vol I,II e III, Addison-Wesley Publishing Company, Inc., 1985.
- (3) ARCA, J.A, Practical WordStar Uses, Sybex Inc, 1983.
- (4) AT&T, AT&T UNIX PC Interface Specification, 1985.
- (5) AT&T, The VI User's Handbook, 1984.
- (6) AT&T, System V Interface Definitions, Issue 2, Vol. II, 1986.
- (7) AT&T, UNIX User Reference Manual - System V - release 2.0, 1984.
- (8) AVANT, H., "The AT&T UNIX PC: A UNIX System for the Masses?", UNIX/WORLD, pp. 56-66, Setembro 1985.
- (9) BERTINO, E., "Design Issues in Interactive User Interfaces", Interfaces in Computing, pp 37-53, Março 1985.
- (10) BOOCH, G., Software Engineering with ADA, The Benjamin/Cummings Publishing Company, Inc, 1983.
- (11) BYLES T., "One Approach to the human - computer interface", IEEE CG&A, pp. 245-251, Dezembro 1984.
- (12) BYERS T.J., "TopView - IBMs long awaited multitasking

program makes its debut", BYTE, pp. 245-251, Outubro 1985.

(13) COBRA COMPUTADORES E SISTEMAS BRASILEIROS, "SIR/AC", Especificação Planejada de Software número 149, Outubro/1987.

(14) COBRA COMPUTADORES E SISTEMAS BRASILEIROS, "FIU - Ferramentas de Interface com o Usuário", Especificação Planejada de Software número 15, Setembro/1986.

(15) COBRA COMPUTADORES E SISTEMAS BRASILEIROS, "SOX", Especificação de Requisitos de Software número 5, junho/1987.

(16) DATA GENERAL CORPORATION, Using CEO Word Processing, Rev. 00, Maio 1982.

(17) EDWARDS J.R. et alii, "The Atari 520ST", BYTE, pp. 89-100, Janeiro 1986.

(18) EMBLEY, D.W. e NAGY, G., "Behavioral aspects of Text Editors", Computing Surveys Vol. 13, no. 1, pp. 33-70, Março 1981.

(19) FINSETH, C.A., "Managing Words - What Capabilities Should You Have With a Text Editor?", BYTE, pp. 302-310, Abril 1982.

(20) FOLEY J.D. et alii, "The Human Factors of Computer Graphics Interaction Techniques", IEEE CG&A, pp. 17-46, Novembro 1984.

(21) FOLEY J.D. et alii, "The human Factors of Graphic Interaction Tasks and Techniques", School of Engineering and Applied Science, Janeiro 1981.

(22) GOLDBERG A. et alii, Smalltalk-80, The language and its implementation, Addison Wesley.

(23) GOOD M., "Etude and the Folklore of User Interface Design", ACM, pp.34-43, 1981.

(24) HECKEL P., The elements of friendly software design, Warner Books Inc, Maio 1984.

(25) HOFFMAN P., Microsoft Word, Made Easy, Osborne McGraw-Hill 1985.

(26) IBM, IBM Personal Computer Professional Editor, Dezembro, 1972.

(27) IEEE, Guide To Software Requirements Specifications, The Institute of Electrical and Electronics Engineers Inc., 1984.

(28) JONG, S., "Designing a Text Editor? The User Comes First", BYTE, pp. 284-300, Abril 1982.

(29) LEMMONS, P., "Beyond the Word Processor", BYTE, pp. 53-56, Janeiro 1984.

- (30) LEWIS, T.G., "Apple Macintosh Software", IEEE Software, pp.89-91, Março 1985.
- (31) MARCUS A., "Corporate Identity for Iconic Interface Design: The Graphic Design Perspective", IEEE CG&A, pp. 24-32, Dezembro 1984.
- (32) MEYROWITZ N. e DAM V.A., "Interactive Editing Systems", Computing Surveys, vol. 3, pp. 321-415.
- (33) MICAL J.R., "Introduction to the Amiga ROM Kernel", BYTE, pp. 116-128, Fevereiro 1986.
- (34) MICROSOFT CORPORATION, Microsoft Windows Users Guide, versão 1.0, 1986.
- (35) MIT PROJECT ATHENA et alii, X Toolkit - A Proposed Architecture Preliminary Draft, Dezembro 1986.
- (36) MONTE, L.C., "Sistemas de Interface com o Usuário", Monografia do Curso Tópicos Especiais em Banco de Dados não Convencionais, COPPE - Engenharia de Sistemas e Computação, Dezembro/1986
- (37) NORMAN, D. A., "Design Rules Based on Analyses of Human Errors", Communications of the ACM, vol. 26, n. 4, pp. 254-256, Abril 1983.
- (38) ROBERT, T.L., "The Evaluation of Text Editors: Metodology and Empirical Results", Communications of the ACM, vol. 26, no.4, pp. 265-283, Abril 1983.
- (39) SCHEIFLER, R. W. e GETTYS, J., "The X-Windows System", ACM Transaction on Graphics, vol.5, n.2, pp 79-109, Abril 1986.
- (40) SIMPSON, H., Design of User Friendly Programs for Small Computers, McGraw-Hill Book Company, 1984.
- (41) SMITH D.C., "Macintosh System Architecture", BYTE, Fevereiro 1984.
- (42) SMITH D.C. et alii, "Designing the Star User Interface", BYTE, pp. 242-282, Abril 1982.
- (43) TESLER L., "The Smalltalk Environment", BYTE, pp. 90-147, Agosto 1981.
- (44) WILLIAMS G et alii, "The AMIGA Personal Computer", BYTE, pp 83-100, Agosto 1985.
- (47) WILLIAMS G., "The AT&T UNIX PC", BYTE, pp 98-105, Maio 1985.
- (48) WILLIAMS G., "The Lisa Computer System", BYTE, pp. 33-53, Fevereiro 1983.