

**ANDREY RICARDO PIMENTEL
JOÃO ALBERTO FABRO
PAULO JAIME**

SLIDE. UM SISTEMA DE HIPERTEXTO E SUA IMPLEMENTAÇÃO

Trabalho apresentado como requisito parcial
para a obtenção de crédito na disciplina de
Trabalho de Graduação II do curso de
Bacharelado em Informática da Universidade
Federal do Paraná
Orientadora: Prof. Cristina Duarte Murta

CURITIBA 1993

SUMÁRIO

SUMÁRIO	I
1. INTRODUÇÃO	1
2. CONCEITOS	3
2.1 NODOS	4
2.2 LIGAÇÃO OU ELO	5
2.3 ÂNCORA	6
2.4 CLASSIFICAÇÃO DE HIPERTEXTOS	6
2.4.1 Sistemas Macro-Literários	6
2.4.2 Sistemas para Exploração de Problemas	7
2.4.3 Sistemas Folheadores (<i>Browsing Systems</i>)	7
2.4.4 Sistemas de Uso Geral	7
2.5 UTILIZAÇÃO DE HIPERTEXTOS	8
2.6 CARACTERÍSTICAS COMUNS ENTRE SISTEMAS DE HIPERTEXTO	9
2.7 VANTAGENS E DESVANTAGENS DO HIPERTEXTO	10
2.8 CARACTERÍSTICAS AVANÇADAS PARA SISTEMAS DE HIPERTEXTO	12
2.8.1 Métodos de Pesquisa em redes de Informação	12
2.8.2 Composições aumentando o modelo de nodos e elos	13
2.8.3 Estruturas que possibilitem a mudança da estrutura da informação	13
2.8.4 Redes hipermídia com poder de processamento.	14
2.8.5 Utilização de Versões	14
2.8.6 Suporte para ambientes de trabalho colaborativo	15
2.8.7 Extensibilidade e Configurabilidade	16
3. SLIDE - UM SISTEMA DE HIPERTEXTO	16
3.1. ESPECIFICAÇÃO DO SISTEMA.	17
3.2 PROJETO DO SISTEMA.	18

3.2.1 Metodologia Orientada a Objetos.....	19
3.2.2 Definição das Classes.	20
3.2.3 Definição das Responsabilidades	23
3.2.4 Definição dos Colaboradores	23
3.2.5. Estruturas de Dados.	24
3.3. IMPLEMENTAÇÃO DO SISTEMA.....	25
3.3.1. A biblioteca de Classes CLASSLIB.....	26
3.3.2. A Implementação dos Objetos do Sistema SLiDe.....	26
3.3.3. A Implementação das Ferramentas do Sistema SLiDe.	27
4. - PROJETO DA INTERFACE.....	28
4.1. DEFINIÇÃO DA POPULAÇÃO USUÁRIA.	28
4.2 DESCRIÇÃO DA INTERAÇÃO COM O USUÁRIO.	29
4.3. DESCRIÇÃO DA INTERFACE.....	30
4.4 CENÁRIOS.	32
4.5 IMPLEMENTAÇÃO DA INTERFACE.....	36
4.5.1 Mecanismo de mensagens	37
4.5.2. Objeto Aplicação.....	38
4.5.3. Objeto Janela	39
4.6 PORTABILIDADE DO SISTEMA	40
5.CONCLUSÃO.....	40
REFERÊNCIAS BIBLIOGRÁFICAS	41

LISTA DE FIGURAS

FIGURA 3.1: CARTÕES CRC DAS CLASSES DO SISTEMA.	22
--	----

FIGURA 3.2: DIAGRAMA DE CLASSES DO SISTEMA SLIDE.....	24
FIGURA 3.3:DIAGRAMA DA ESTRUTURA DE DADOS UTILIZADA	25
FIG 4.1. TELA MOSTRANDO UM NODO	32
FIG 4.2. TELA MOSTRANDO UM NODO	33
FIG 4.3. TELA MOSTRANDO O MENU "ARQUIVO" ABERTO	33
FIG 4.4. TELA MOSTRANDO O MENU "EDITAR" ABERTO.....	34
FIG 4.5. TELA MOSTRANDO O MENU "NAVEGAÇÃO" ABERTO NA POSIÇÃO NORMAL	34
FIG 4.6. TELA MOSTRANDO A TELA DE DIÁLOGO REFERENTE AO COMANDO "TÓPICOS".....	35
FIG 4.7. TELA MOSTRANDO O MENU "AJUDA" ABERTO	35
FIG 4.8. TELA MOSTRANDO O MENU NAVEGAÇÃO ABERTO EM QUALQUER POSIÇÃO DA TELA	36
FIGURA 4.9: HIERARQUIA DO OBJETO HIPERTEXTO	38
FIGURA 4.10: HIERARQUIA DO OBJETO GERENCIADOR DE TELA.....	39

1. INTRODUÇÃO

A necessidade de recuperar informações rápida e facilmente é uma das maiores preocupações da ciência da computação. Devido a naturalidade no uso da informação e maior eficiência na sua recuperação, sistemas de hipertexto estão sendo cada vez mais utilizados e pesquisados.

O objetivo deste trabalho é descrever as características, o projeto e a implementação de um sistema de hipertexto para uso geral desenvolvido como projeto de iniciação científica da Universidade Federal do Paraná.

A motivação principal do desenvolvimento deste sistema é o uso de várias tecnologias de ponta: metodologias de projeto e programação orientadas a objeto, técnicas de desenvolvimento de interfaces com o usuário mais amigáveis, além do estudo aprofundado da tecnologia ainda emergente dos sistemas de hipertexto.

A idéia de hipertexto é organizar documentos através de associações, baseado no modelo de funcionamento da mente humana. Isto foi primeiramente descrito por Vannevar Bush, em 1945, no artigo *As We May Think* [Bus45]. Neste, ele descreve um sistema de informações denominado *Memex*. O projeto utilizava microfilmes e fotocélulas como meios de armazenamento de informações. Devido à tecnologia existente na época, não houve um aprofundamento maior nesses conceitos.

Posteriormente, Douglas Engelbart, em Stanford, realizou estudos sobre o uso do computador como ferramenta de auxílio no desenvolvimento do raciocínio humano, desenvolvendo novas teorias sobre a interação homem-máquina, onde os principais pontos eram a construção de sistemas mais dinâmicos e flexíveis e uma eficiência maior no armazenamento e recuperação das informações com o mínimo esforço do usuário. Baseado nesses estudos, ele e um grupo de pesquisadores implementaram em 1968 o *NLS (oN Line*

System) [EE68] que era um sistema de informações interligadas através de uma estrutura formada por uma rede de elos.

O *NLS* foi inicialmente utilizado como ferramenta para armazenar projetos, programas, documentos e pesquisas, permitindo o compartilhamento dos dados e comunicação entre os pesquisadores, e foi um dos primeiros programas a fazer uso de *mouse* com ajuda dependente de contexto, entre outras facilidades.

Nessa mesma época, Ted Nelson e seu grupo da Universidade de Brown, desenvolveram um sistema de hipertexto tendo como objetivo um ambiente literário unificado de escopo mundial. Foi onde o termo hipertexto apareceu pela primeira vez. O sistema foi denominado *Xanadu* [Nel81], e possibilitava a edição de textos, criação de conexões entre documentos, pesquisa dos textos interligados, expansão do volume de texto e adição de novas ligações.

A evolução da tecnologia, com o uso de disco ótico e ambientes gráficos, fez surgir sistemas que uniram, além dos textos, gráficos, sons e imagens, dando origem ao que chamamos de hiperídia [Mar92b]. Um dos produtos de hipertexto que contribuiu para a sua divulgação foi o *HyperCard* [Nie90], sistema desenvolvido para o *Macintosh* da *Apple*, aproveitando do poder da sua interface gráfica. O sistema inclui a linguagem *HyperTalk* [Nie90], que permite ao usuário a programação dos eventos que podem ocorrer.

A partir daí surgiram outros sistemas de hipertextos para o ambiente mono-usuário: *Hiperties* [SM86], baseado no Sistema de Enciclopédia Interativa da Universidade de Maryland, EUA, disponível para computadores IBM-PC e estações de trabalho SUN; *Guide* [Her87] da OWL International Incorporation, EUA, com versões para o IBM-PC e Macintosh.

Foram criados também vários sistemas para computadores de grande porte, tais como: *Intermedia* [YHMD88] desenvolvido pela Universidade de Brown, *NoteCards* [Hal88], do Centro de Pesquisa da Xerox, *gIBIS* (*Graphical Issue-Based Information System*)

[BC88] da empresa MCC - Microelectronic and Computer Tecnology Corporation e *KMS* [AMY88], desenvolvido na Universidade de Carnegie-Mellon.

As aplicações comerciais já incorporam o hipertexto nos seus produtos como manuais de referência, documentação e sistemas de ajuda. No nível de pesquisa, estuda-se o seu uso em sistemas para trabalho cooperativo, ambientes de desenvolvimento de software e sistemas de treinamento por computador.

Esta monografia é composta de 5 capítulos. O capítulo 2 descreve os conceitos básicos de hipertexto. O capítulo 3 descreve o projeto e a implementação do Sistema não Linear para organização de Documentos - SLiDe. No capítulo 4 é feita a descrição da interface do sistema e da sua implementação. E o capítulo 5 apresenta as contribuições do trabalho e sugestões para pesquisas futuras.

2. CONCEITOS

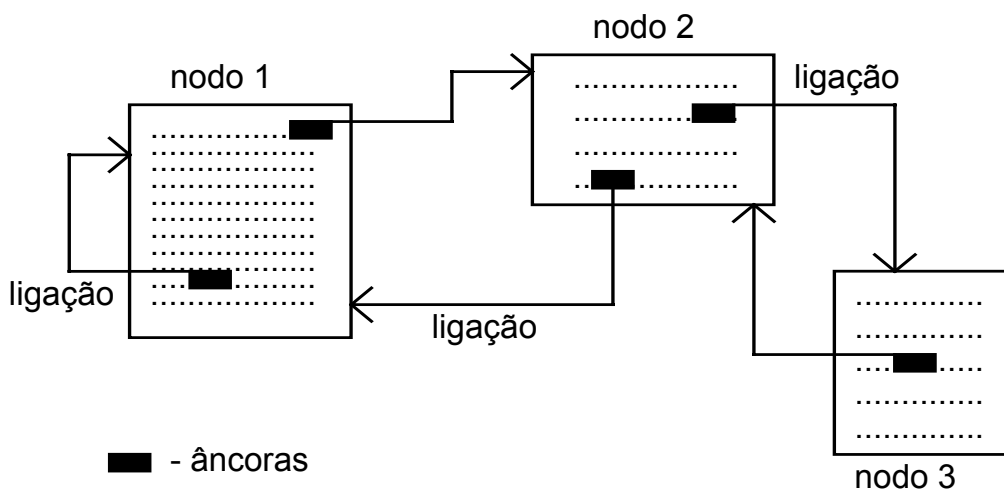
Um hipertexto é um sistema que possibilita a organização de documentos de modo não linear. Ou seja, ao contrário de livros, artigos e textos, que são divididos em capítulos, seções, páginas e parágrafos, dispostos em uma sequência linear pré-definida, um sistema de hipertexto permite que o acesso seja feito conforme o interesse do leitor.[Mar92a]

Por possuir uma estrutura em formato de grafo ou rede é possível "navegar" através de textos com assuntos que possuam uma relação sintática e/ou semântica. Os sistemas com tais características são denominados Sistemas de Hipertexto.

Um hiperdocumento é um documento que possui ligações entre palavras-chave e telas de informação relacionados. Em outras palavras, hiperdocumento é um documento em forma de hipertexto.

Assim, identificamos na figura 2.1 as unidades que compõem um hiperdocumento: nodos, ligações e âncoras, descritos a seguir.

FIGURA 2.1: NODOS, LIGAÇÕES E ÂNCORAS DE UM HIPERDOCUMENTO



2.1 NODOS

Os nodos são as unidades de informações de um hiperdocumento. Esses nodos representam idéias, conceitos, comentários, explicações ou outros detalhes.

Um nodo deve conter uma idéia ou um conceito da forma mais concisa possível, levando em conta as limitações de tamanho da tela. Detalhamentos, observações, notas de rodapé ou até informações entre parênteses devem ser colocados como ligações, possibilitando a escolha entre um aprofundamento maior no tema ou não.

Os nodos são os elementos básicos de estruturação do hiperdocumento, sendo, portanto, muito importante o seu dimensionamento. O tamanho dos nodos deve ser tal que não ultrapasse muito os limites da tela do computador, pois isto dificulta a navegação. Mas nodos muito pequenos podem acarretar uma grande fragmentação da informação, dificultando a compreensão das idéias de uma maneira lógica.

Podem existir nodos de vários tipos, como: textuais, gráficos, de imagens, sons, seqüências de vídeo, e também nodos que executem programas quando são selecionados. Esta capacidade de utilização de vários recursos simultâneos no mesmo documento aumenta a capacidade de representação dos documentos, podendo aumentar a facilidade de compreensão por parte do leitor.

2.2 LIGAÇÃO OU ELO

Os elos ou ligações são essenciais num sistema de hipertexto, pois é através deles que buscamos as informações pertinentes ao assunto. Os elos não são necessariamente lineares ou hierárquicos. Eles estabelecem a ligação entre os nodos. Cada elo possui um nodo origem e um nodo destino, que é ativado por comando do usuário.

As funções de um elo ou ligação são:

- conectar uma referência de um documento ao próprio documento;
- colocar uma observação ou comentário em anexo;
- interligar textos;
- associar gráficos, figuras, tabelas, imagens ou sons;
- conectar documentos independentes e fornecer mais detalhes sobre as informações, por exemplo, uma nota de rodapé.[Mar92a]

Os elos podem também, assim como os nodos, serem de diversos tipos, cada um com um método de exibição. Um exemplo seria um nodo indicando uma nota de rodapé, que pode simplesmente ser apresentada na tela sobre o texto comum, enquanto o leitor mantiver uma tecla pressionada. Outro exemplo seria o de uma ligação associada a um trecho de programa, onde sua ativação executasse o trecho, provocando uma animação, a apresentação de uma mensagem, etc...

Um tipo interessante de elo é o que pode levar a vários outros nodos relacionados. Este elo/ligação pode apresentar um menu onde o leitor escolhe qual nodo deseja visitar, ou simplesmente levar ao primeiro de uma série, apresentando a opção do leitor "folhear" todos os nodos de destino, ou retornar ao nodo origem.

Os elos, também chamados *links* são direcionados, partindo do nodo origem e chegando ao nodo destino. Entretanto, quase todos os sistemas de hipertexto possibilitam o caminhar inverso, retornando ao nodo origem. Isto proporciona o retorno sempre ao último nodo visitado, até o nodo inicial do hipertexto.

2.3 ÂNCORA

Uma âncora é uma região em destaque na tela que é o ponto de partida de uma ligação, isto é, uma palavra-chave que relaciona o texto deste nodo com as informações contidas no nodo destino deste elo. Podem existir também âncoras gráficas, que se posicionam sobre um desenho ou gráfico. Uma âncora deve ser bem visível, de modo que seja de simples localização e facilite a navegação pelo hiperdocumento.

As âncoras devem ser bem escolhidas, de modo a intuir ao usuário que tipo e informação ele encontrará ao seguir o *link* correspondente. Isto evitará que o leitor do hiperdocumento siga ligações que não lhe interessem. Alguns sistemas chegam a fornecer uma breve descrição do nodo destino quando o leitor posiciona o apontador (*mouse*) sobre uma âncora, possibilitando uma melhor previsão da importância ou não daquele elo. A utilização de elos tipados (apresentação diferente para elos de comentário, referência bibliográfica, e de aprofundamento no assunto) também facilita a utilização mais precisa do sistema de hipertexto, e, conseqüentemente, a facilidade de leitura dos hiperdocumentos.

2.4 CLASSIFICAÇÃO DE HIPERTEXTOS

Uma classificação bastante aceita para hipertextos é a proposta por Jeff Conklin [Con87], devido a sua globalidade e generalização. Segundo ele, os sistemas de hipertexto podem ser divididos, segundo a área de aplicação, em: Sistemas Macro-Literários, Sistemas para Exploração de Problemas, Sistemas Folheadores (*Browsing Systems*) e Sistemas de Uso Geral.

2.4.1 Sistemas Macro-Literários

Estes sistemas têm por objetivo integrar imensas bases de conhecimento, de modo a facilitar o acesso através de uma interface simples. Estes sistemas devem suportar a leitura, colaborações, críticas e edição de partes do texto. Uma utilização é automatizar consultas a dicionários, enciclopédias e bibliotecas, mas sua implementação é difícil devido ao tamanho

das bases de dados, e a conversão de texto linear em hipertexto de forma automática. Exemplos: Memex e Xanadu [Mar92a][Con87].

2.4.2 Sistemas para Exploração de Problemas

Servem para auxiliar a estruturação e manipulação de idéias para a solução de problemas. Estes sistemas possibilitam a organização de idéias sobre um problema através de uma rede de nodos e elos tipificados, com a contribuição de várias pessoas compartilhadamente. Podem ser usados para análise de sistemas, processamento de idéias e outros campos onde seja necessária colaboração de um grupo de pessoas para solucionar um problema. Exemplos: NLS/Augment [EE88], gIBIS [BC88].

2.4.3 Sistemas Folheadores (*Browsing Systems*)

Estes sistemas são utilizados para promover acesso rápido à informação de um modo simples e de fácil aprendizado. Servem principalmente para treinamento, referência à informações (manuais) e sistemas públicos de informação, não editáveis. São muito semelhantes aos sistemas Macro-literários, porém armazenam quantidades menores de informação. Exemplos: KMS [AMY88] (Universidade de Carnegie-Mellon, EUA) e Hyperties [SM86].

2.4.4 Sistemas de Uso Geral

São sistemas desenvolvidos para uso genérico, geralmente suportando mais de uma aplicação. Utilizam uma ampla faixa de ferramentas, e seu objetivo principal é permitir experimentação com a tecnologia de hipertexto. Exemplos: Notecards [Hal88], Neptune [DS86], HyperCard [Nie90] e Guide [Her87].

As gerações dos sistemas de Hipertexto são divididas, segundo Halasz [Hal88], em:

Primeira Geração: representada pelos sistemas NLS/Augment [EE68] e ZOG [AMY88], eram baseadas em "*mainframes*", utilizando modo texto, com bases de dados compartilhadas por grupos de usuários, geralmente visando trabalho cooperativo.

Segunda Geração: representada por sistemas como Notecards [Hal88], Intermedia [YHMD88] e Neptune [DS86], possui como características o uso de gráficos e interfaces com o usuário mais avançadas. Estes sistemas apresentam facilidades para localização dentro dos hiperdocumentos, mapas gráficos de localização, e são geralmente projetados para um só usuário ou pequenos grupos de desenvolvedores. Pertencem a esta geração também os sistemas desenvolvidos para o sistema PC, como o Guide [Her87], ou para o Macintosh, como o HyperCard [Nie90].

Terceira Geração: as características para os sistemas de última geração são apresentadas na seção 2.8.

2.5 UTILIZAÇÃO DE HIPERTEXTOS.

Uma utilização bastante promissora dos hipertextos é na área de educação auxiliada por computador, devido às facilidades oferecidas por esta tecnologia para auxiliar na captação do conhecimento por parte do "aluno", além de possibilitar que cada um possa escolher o caminho mais conveniente para o aprendizado. Esta área está começando a ser pesquisada e pode apresentar resultados promissores, principalmente com o uso das características de multimídia que começaram a se tornar comuns.[GM88][Woo91].

Outra utilização promissora é a de auxílio ao desenvolvimento e documentação em projetos de sistemas de software. Esta área conjugaria as potencialidades dos sistemas de hipertexto para exploração de problemas, dos sistemas folheadores e de propriedades avançadas das tecnologias experimentais de hipertextos [Mar92b][Woo91].

Uma área de pesquisa que tem despertado interesse é a dos "livros eletrônicos". Atualmente, já existem várias versões eletrônicas de livros da literatura, manuais de referência e bancos de dados de texto, sobre diversos assuntos. Porém, a grande capacidade

do uso de hipertextos nesta área é o desenvolvimento de sistemas que sirvam como um verdadeiro banco de informações pessoais, onde cada pessoa possa estruturar sua documentação e seus conhecimentos eletronicamente, utilizando técnicas de hipertexto para facilitar o acesso a estas informações [Woo91].

Outras utilizações, esta já bem difundida, são os manuais on-line. Este tipo de documentação possui certas características que o tornam ideal para a organização em forma de hipertexto:

- são rigidamente estruturados em seções/capítulos;
- não são usados de forma linear;
- geralmente são grandes e difíceis de manejar;
- crescem com o passar do tempo: novas versões, avanços tecnológicos,...
- Muitos sistemas já estão comercialmente disponíveis para este tipo de utilização [Dam88].

Outras áreas que vem despertando bastante interesse para a utilização de hipertextos que podem ser citadas são:

Área jurídica e médica, para fazer referência a casos já ocorridos (jurisprudência) e auxiliar diagnósticos médicos;

Sistemas de co-autoria ou ambientes de trabalho cooperativo, onde uma equipe pode desenvolver um determinado assunto, dentro de um ambiente de hipermídia [AMY88].

2.6 CARACTERÍSTICAS COMUNS ENTRE SISTEMAS DE HIPERTEXTO.

A grande maioria dos sistemas de hipertexto atuais apresentam, cada qual, suas peculiaridades e particularidades. Entretanto, algumas características se apresentam comuns a todos os sistemas:

Velocidade de atualização: para ser agradável ao leitor, e poder ser aceito e utilizado, os sistemas de hipertexto devem apresentar uma característica simples, porém essencial e por vezes problemática: apresentar as atualizações dos nodos na tela em tempo

hábil. Este tempo pode ser de, no máximo, dois a três segundos [Mar92b], que é o limite de tempo pelo qual um usuário pode esperar sem começar a pensar em desistir de utilizar o sistema. Tempos maiores de atualização só são permitidos em casos especiais, e mesmo assim raramente, sob pena do sistema ser abandonado por falta de paciência dos usuários, independente de quão bom sejam suas características;

A questão da facilidade de localização: esta é uma questão ainda em aberto, devido, talvez, ao costume dos usuários à ordem linear das coisas. Ao ser apresentado a um sistema de hipertexto, um usuário poderá perder o controle sobre sua localização incrivelmente rápido, caso não sejam fornecidas ferramentas simples e poderosas de localização, tais como capacidades de retroceder, localizadores gráficos e históricos indicando por onde já se passou;

Facilidade de uso: é fundamental para sistemas de hipertexto uma interface o mais amigável (*user-friendly*) possível. Devido ao hipertexto ser uma forma relativamente nova de acesso a informação, os usuários devem precisar se preocupar apenas em como conseguir as informações que precisam, sem a necessidade de preocupações adicionais sobre opções variadas de como seguir ligações, e formas muito complicadas de acesso;

Capacidade gráfica/hipermídia: atualmente, estas duas capacidades estão se tornando obrigatórias a todos os tipos de sistemas. Com o avanço dos computadores, tais capacidades se tornarão logo indispensáveis, e por isto todos os hipertextos devem apresentar facilidades de utilização de gráficos coloridos, sons, animações, etc...

2.7 VANTAGENS E DESVANTAGENS DO HIPERTEXTO.

A tecnologia de hipertexto apresenta vantagens significativas em relação ao meio impresso de manutenção de informações em forma de texto, e até mesmo em forma gráfica. A organização da informação em forma de hipertexto possibilita, entre outras coisas:

- facilidade muito maior para investigação de referências, o que auxilia no processo de pesquisa e aprendizado;

- facilidade de manipulação de grandes bases de conhecimento, simultaneamente, de acordo com as necessidades que se apresentem, sem os inconvenientes do meio impresso (peso, volume, necessidade de deslocamento);
- capacidade de organização de formas dinâmicas e flexíveis, através de maneiras diversas de estruturação (hierárquica, não hierárquica, com diversos níveis ou visões).
- possibilidade do leitor realizar anotações e referências, sem a necessidade de utilizar margens ou folhas separadas;
- facilidades adicionais de pesquisa e recuperação de informação, através de palavras-chave, assuntos, referência a sessões anteriores de pesquisa, histórico de visita a hiperdocumentos, etc.
- possibilidade de busca de informações através de sistemas gráficos (*browsers* ou folheadores) que permitem um acesso mais direto e intuitivo;
- facilidades para criação/manutenção de hiperdocumentos, apresentando um meio mais simplificado para alterações/modificações que o meio impresso;
- por estar em um ambiente computacional, o hipertexto possibilita a utilização dos recursos dos equipamentos para criar audiovisuaizações, combinar texto com imagens estáticas (gráficos) e também com imagens animadas (animações, sequências de vídeo), além de sons e programas executáveis. "As vantagens não se resumem à integração de toda esta variedade de tipos de informação, mas também na possibilidade de se ter controle interativo sobre seus elementos" [Mar92b].

Entretanto, apesar de apresentar todas estas vantagens, os sistemas de hipertexto são criticados por suas limitações. A maior e mais importante limitação ocorre devido à estrutura fragmentada na qual as informações são organizadas num hipertexto. Esta estrutura, quando mal organizada ou muito complexa, pode facilmente levar ao leitor a sensação de

"desorientação". Esta dificuldade pode ser minimizada através do uso de *browsers* e outras ferramentas de auxílio à localização [Nie90]. Outra desvantagem é a dificuldade encontrada, muitas vezes, em estruturar o conhecimento e as informações em nodos discretos, visto que muitas vezes as informações precisam ser pesquisadas de forma linear, ou são tão fortemente interligadas que uma separação em nodos distintos, apenas prejudicaria a compreensão [Mar92b].

2.8 CARACTERÍSTICAS AVANÇADAS PARA SISTEMAS DE HIPERTEXTO.

A terceira geração de sistemas deve apresentar (e alguns sistemas já possuem) características avançadas, citadas a seguir:

2.8.1 Métodos de Pesquisa em redes de Informação.

O principal meio de busca de informações em um hiperdocumento é através dos elos, e este é justamente seu ponto forte. Este esquema facilita a tarefa de encontrar informações por parte do usuário. Entretanto, com o aumento das "redes de informação" [Hal88] e da complexidade dos hiperdocumentos, começa a se tornar trabalhosa e cansativa a busca pela informação requerida, e por isto começam a se tornar importantes as capacidades de pesquisa através de busca por assunto ou por palavra-chave. Estas capacidades facilitam o encontro de informações relevantes, onde quer que estas se encontrem. Isto também pode servir como um "primeiro passo" em um processo de busca, restringindo a área a ser pesquisada, e aumentando deste modo a velocidade de acesso às informações. Alguns sistemas já implementam capacidades simples de pesquisa por assunto ou palavra-chave, mas ainda há muito a ser pesquisado nesta área, até que possam ser respondidas questões como: "Quais nodos possuem ligações que levam a nodos que contenham a palavra hipertexto?".

2.8.2 Composições aumentando o modelo de nodos e elos.

O modelo de hipertexto é formado apenas por nodos e elos. Esta concepção simples e poderosa é a base de toda a tecnologia. Pode-se, entretanto, melhorar este modelo adicionando-se nodos que são composições de outros nodos fortemente relacionados, como se fossem "capítulos" do hiperdocumento. Através desta idéia, é possível estruturar melhor as redes de informação complexas, organizando-as de uma forma mais ordenada e consistente do que o atual agrupamento de nodos individuais formando imensos hiperdocumentos. Estas composições deveriam possuir nodos relacionados com o assunto em questão, um browser de localização interna, e um cartão inicial que representaria um "índice" para este grupo de nodos. Algumas experiências já foram feitas neste sentido, mas ainda há muito espaço para pesquisa e desenvolvimento de estruturas mais complexas e ao mesmo tempo mantendo a simplicidade e funcionalidade do modelo de hipertexto.

2.8.3 Estruturas que possibilitem a mudança da estrutura da informação.

Grande parte dos sistemas de hipertexto possibilitam aos leitores acrescentar informações ou notas aos hiperdocumentos. O próprio processo de autoria destes documentos é um processo contínuo de criação, onde muitas vezes no início da criação o autor não tem uma idéia precisa da melhor estrutura para seu documento. Deste modo, é importante que os sistemas possuam capacidade de re-estruturação dos nodos e *links*, possibilitando que o hiperdocumento, mesmo com acréscimos e modificações, possa ser reconfigurado de modo a continuar compreensível e estruturado, e não se torne um "caos" de informações desconexas e nem sempre relevantes ao assunto pesquisado.

As estruturas que possibilitam este tratamento são as "Estruturas Virtuais". As estruturas atuais de nodos e ligações são representadas estaticamente, isto é, as ligações são feitas entre palavras-chave e nodos de informação fixos. Ligações virtuais não ligariam palavras, e sim idéias-chaves a nodos, que também não seriam fixos, contendo apenas texto, mas representariam conceitos completos a respeito de um assunto. Algumas experiências já

foram realizadas neste sentido, mas um problema ainda enfrentado é que o tempo gasto para gerenciar estas estruturas inviabiliza a velocidade de acesso, tão importante para o sucesso de um sistema de hipertexto.

2.8.4 Redes hipermídia com poder de processamento.

Sistema de hipertexto ou hipermídia geralmente são apenas armazenadores de informações, com possibilidades de leitura/apresentação de forma não linear. Estes sistemas geralmente não possuem capacidade de processamento de informações, além daquelas necessárias para a procura dos nodos e sua apresentação na tela. Características poderiam ser adicionadas, proporcionando um aumento significativo nas capacidades dos hiperdocumentos, tais como:

- nodos que, quando selecionados, apresentariam sequências de animações ou explicação a respeito do assunto, de uma forma interativa;
- ligações que, ao serem selecionadas, apresentariam animações, de acordo com o tipo da ligação;(uma página sendo folheada, uma animação representando busca de informações, etc.);
- inclusão da utilização de ferramentas de inteligência artificial, como pesquisas em linguagem natural, e adaptação às formas de utilização de cada usuário (suas características pessoais, facilidades de aprendizado, etc...);

2.8.5 Utilização de Versões

"Versões" ou "visões" são ferramentas dos hipertextos que possibilitam que leitores diferentes "vejam" o hiperdocumento de maneira diferente. Esta característica faz com que um usuário iniciante veja apenas as ligações mais simples e diretas, correndo menos risco de se confundir durante a navegação. Já leitores mais experientes ou autores de hiperdocumentos teriam acesso a todas as ligações do sistema, pois devido à sua maior familiaridade de uso, correm menos riscos de se confundir, além de precisarem acessar toda a rede o mais rápido

possível. Vários sistemas já utilizam estas técnicas, que tem como único inconveniente a necessidade de se manter grandes bases de informação para conter as várias versões de um mesmo documento. Uma utilização que tem se mostrado promissora para a pesquisa é a utilizada em sistemas de desenvolvimento de software, onde se pode acompanhar o projeto através das versões que vão surgindo durante o desenvolvimento [AMM88].

2.8.6 Suporte para ambientes de trabalho colaborativo.

Sistemas de hipertexto e hiperídia são formas naturais de suportar trabalho cooperativo. Capacidade de fazer anotações, de manter múltiplas organizações do mesmo material (Visões) e transferir mensagens¹ entre várias pessoas que atuam sobre um determinado documento são formas básicas de trabalho cooperativo. Muitos dos sistemas já desenvolvidos possibilitam o acesso simultâneo de vários usuários leitores ao mesmo hiperdocumento (KMS, Intermedia). Isto provém de sua origem em sistemas centralizados (*Mainframe based*). Entretanto, nenhum sistema ainda contém os mecanismos necessários a um verdadeiro trabalho cooperativo: capacidade de alteração, por vários usuários, de um mesmo hiperdocumento, e comunicação mais direta entre colaboradores. Os novos sistemas devem possuir uma preocupação maior com os bancos de dados associados, de modo que seja possibilitada uma facilidade de alteração maior que a atual, onde apenas um usuário pode modificar um documento a cada instante. Além disto, é necessário que, com a experiência no uso mais direto de sistemas de hipertexto para trabalho cooperativo, sejam criadas formas de interação entre colaboradores que facilitem o entendimento por parte de todos do objetivo global do projeto, de modo que este possa ser alcançado. Uma maior pesquisa nestas áreas (bancos de dados compartilhados e comunicação entre colaboradores) é essencial para que o hipertexto alcance sua capacidade potencial na área de trabalho cooperativo.

¹ Esta capacidade de transferir mensagens é inerente às anotações, quando estas são possíveis. Entretanto, vários sistemas incrementam esta característica realmente possibilitando o envio de mensagens eletrônicas entre vários usuários de um mesmo hiperdocumento, no estilo do correio eletrônico das grandes redes de computadores.

2.8.7 Extensibilidade e Configurabilidade

O modelo de hipertexto, por sua simplicidade e flexibilidade, tem capacidade de ser utilizado nas mais diferentes áreas: medicina, direito, literatura, pesquisa científica, entre outras. Entretanto, justamente pela simplicidade do modelo, é muito difícil adaptá-lo as necessidades de cada um destes tipos de aplicação. As capacidades dos sistemas de hipertexto e hipermídia são de manipulação e representação do conhecimento, nas mais variadas formas. Deste modo, é difícil encontrar um sistema que se adeque perfeitamente a um tipo de aplicação, pois todos os sistemas se adequam mais ou menos bem a todos os tipos de aplicações. A solução para este problema é incluir nos sistemas meios para que o próprio usuário possa configurar/estender as capacidades do sistema de modo a atender suas necessidades específicas. Isto pode ser conseguido através da inclusão de uma linguagem de programação para o próprio sistema, ou um ambiente altamente configurável ao gosto do usuário. Exemplos destas técnicas são encontrados nos sistemas HyperCard (com a linguagem de programação de eventos HyperTalk) [Nie90] e NoteCards [HAL88].

Um modelo adequado deveria apresentar formas simples de configuração, de modo que usuários sem experiência em programação pudessem adequar o sistema a sua própria forma de utilização, e possuindo também uma linguagem de programação que controlasse todos os eventos de interação com o usuário, estendendo ao máximo as possibilidades de melhoramentos no sistema.

3. SLIDE - UM SISTEMA DE HIPERTEXTO.

O sistema, denominado "Sistema não Linear para organização de Documentos" (SLiDe), é um gerenciador de hipertexto. Este sistema tem por objetivo possibilitar aos usuários acessar (ler) hiperdocumentos, além de criar (escrever e formatar) os seus próprios documentos.

Uma das principais preocupações do sistema SLiDe é com a facilidade de interação com o usuário, possibilitando a qualquer pessoa utilizar eficientemente o sistema com um mínimo de treinamento. Deste modo, procuramos atingir um dos principais objetivos deste sistema, que é facilitar o acesso às informações colocando-as em um formato que permite maior facilidade de pesquisa e acesso rápido.

Este capítulo descreve o sistema, o projeto e a implementação do núcleo do sistema. A interface será tratada no capítulo 4.

3.1. ESPECIFICAÇÃO DO SISTEMA.

O sistema é um sistema de hipertexto de uso geral, isto é, não está voltado para nenhuma aplicação em especial. O sistema será apresentado em duas versões: a versão de leitura que permite somente a leitura do hiperdocumento e a versão de edição que permite, além da leitura, a edição do hiperdocumento. O sistema funciona em computadores do tipo IBM-PC, sob ambiente WINDOWS 3.1 [NY92].

A versão de leitura tem ferramentas para minimizar algumas dificuldades que ocorrem na leitura de um hiperdocumento. Devido a sua estrutura, quase sempre em rede, a leitura de um hiperdocumento às vezes é difícil até para usuários experimentados. Uma das dificuldades é o usuário ficar "perdido no hiperespaço" [Mar92b], ou seja, ele entrar num circuito do grafo do hiperdocumento e não saber por onde sair. Outra dificuldade ocorre quando o usuário segue pelo caminho errado e não consegue chegar ao tópico procurado. Estas dificuldades podem ocorrer até em hiperdocumentos pequenos, e nos grandes torna quase impossível a leitura. Estas ferramentas são as seguintes:

- Lista de tópicos: é uma lista de todos os nodos do hiperdocumento, onde cada item leva diretamente ao nodo procurado.
- Histórico: é uma lista de todos os nodos já visitados do hiperdocumento, na ordem de visitaç o, onde cada item leva diretamente ao nodo procurado.

- Pesquisa: uma opção que permite ir diretamente a um nodo através do assunto sobre o qual ele trata.
- Anterior: permite retornar ao último nodo visitado anteriormente.
- Avançar e Voltar: permitem visitar os nodos na ordem da lista de tópicos descrita acima.
- Início: Vai para o nodo inicial do hiperdocumento.
- *Browser*.

Um *browser* é um mapa gráfico que serve para que o leitor do hiperdocumento possa se "localizar" internamente, visualizando graficamente quais foram os últimos nodos visitados, quais nodos ele pode acessar a partir do nodo atual, e também o caminho a seguir caso não esteja mais interessado nos assuntos relativos a este nodo específico. Através do *browser*, o leitor pode facilmente localizar seu posicionamento dentro do hiperdocumento, e saltar diretamente para um nodo que seja de maior interesse, sem precisar seguir até lá através das ligações.

A versão de edição, além das facilidades da versão de leitura, provê ferramentas para a transformação de arquivos de texto normal em hiperdocumentos e para a edição de nodos e âncoras. O sistema não possui um editor de textos, mas aceita textos nos formatos dos principais editores do mercado.

3.2 PROJETO DO SISTEMA.

Para a implementação do sistema foi usado o paradigma de orientação a objetos. Este paradigma foi escolhido porque permite uma melhor modelagem do mundo real para o computador. Ele utiliza conceitos mais naturais e que têm correspondência com o mundo real, como objetos e mensagens, o que torna mais fácil a passagem do mundo real para o computacional.

A orientação a objetos favorece também alguns itens de qualidade de software como: legibilidade, facilidade de manutenção e reaproveitamento de código.

3.2.1 Metodologia Orientada a Objetos.

Para desenvolver o projeto foi utilizada a metodologia descrita no artigo "A Laboratory For Teaching Object-Oriented Thinking" [BC89], por ser simples e fácil de ser aplicada. Esta é uma adaptação da metodologia apresentada por Wirfs-Brock, Wilkerson e Wiener no livro "Designing Object-Oriented Software" [WWW90].

A metodologia é baseada nos três atributos principais de um objeto na fase de projeto: nome da classe, suas responsabilidades e seus colaboradores.

O nome da classe deve descrever o objeto no contexto geral do ambiente. Deve-se tomar um certo cuidado e gastar um pouco de tempo para escolher o conjunto certo de palavras para descrever o objeto.

As responsabilidades devem identificar os problemas a serem resolvidos e não as soluções. Identificando-se os problemas fica fácil escolher entre as soluções possíveis. Novamente deve-se tomar cuidado com o conjunto de palavras usadas. As frases devem ser curtas e concisas.

Os colaboradores de um objeto são os objetos para os quais este irá enviar mensagens a fim de satisfazer as suas responsabilidades.

Para facilitar o projeto, Cunningham inventou os cartões Classe-Responsabilidades-Colaboradores (CRC). Esses cartões servem, também, como meio de documentação do projeto. Os cartões CRC são pedaços de papel grosso medindo 10x15cm e contém o nome da classe, as responsabilidades e os colaboradores, como na figura 3.1.

A disposição espacial dos cartões é importante. Quando duas classes são mútuas colaboradoras, seus cartões devem ser colocados levemente sobrepostos. Quando uma classe é colaboradora de outras classes (a classe é usada pelas outras), seu cartão deve ficar abaixo dos cartões das outras classes. Se há uma relação de herança, deverá haver uma sobreposição quase completa dos cartões. A disposição dos cartões é importante, pois em projetos ainda incompletos é possível notar a localização de uma classe antes da mesma ter sido projetada.

Esta metodologia tem as seguintes etapas:

- Definição das classes e das estruturas de dados de cada classe;
- Definição das responsabilidades de cada classe;
- Definição dos colaboradores de cada classe;
- Definição da hierarquia de classes;
- Definição do protocolo de cada classe;
- Implementação dos métodos ;

As etapas de mapeamento para a linguagem e a implementação serão descritas no capítulo que trata da implementação. As demais etapas são descritas a seguir.

3.2.2 Definição das Classes.

As classes do sistema foram definidas segundo a metodologia descrita acima. As classes do sistema estão também descritas nos cartões CRC da figura 3.1. As classes do sistema são:

- classe Hipertexto: subclasse da classe *object*; classe que inicia e executa o programa;
- classe Nodo: subclasse da classe *object*; classe que cria e gerencia as operações sobre um nodo;
- classe Link: subclasse da classe *object*; classe que cria e executa operações sobre *links*;
- classe Lista: subclasse da classe *Queue*; classe que cria uma lista de objetos e opera sobre ela;
- classe Gerador_ID: classe que gera o identificador de um nodo;
- classe Pilha: subclasse da classe *Stack*; classe que cria uma pilha de objetos e opera sobre ela;
- classe Gerenciador_De_Tela: classe que gerencia a exibição de objetos na tela;
- classe Tela: classe que exibe objetos na tela;

- classe Tela_De_Edição: classe que faz a edição de nodos e âncoras;
- classe Ident: classe que contém o identificador de um nodo;
- classe Arquivo_Em_Disco: classe que gerencia a gravação e recuperação de nodos do disco.

FIGURA 3.1: CARTÕES CRC DAS CLASSES DO SISTEMA.

HIPERTEXTO cria hipertexto executa finaliza hipertexto	TELA_EDIÇÃO GERENCIA_TELA TELA
LINK cria um link obtém nodo apontado retira o link	ARQUIVO_DISCO NODO
GERENCIADOR DE TELA gerencia a exibição do nodo gerencia a exibição da âncora gerencia ícones de edição gerencia exibição do cursor gerencia histórico de navegação obtem nodo visitado anteriormente	NODO TELA TELA_EDIÇÃO
GERADOR_ID gera um novo ID de nodo	
TELA exibe nodo exibe ícone exibe cursor exibe âncora	
NODO ter um ID cria um nodo insere uma âncora obtém link rel. com a âncora retira âncora retira nodo	LINK GERADOR_ID LISTA IDENT
LISTA coloca elemento na lista retira elemento da lista verifica se está vazia	LINK
PILHA coloca elemento na pilha retira elemento da pilha verifica se está vazia	IDENT
IDENT contém o ID de um nodo	
TELA DE EDIÇÃO separa texto em nodos escolhe âncoras escolhe nodos destino nomeia os nodos	NODO TELA GERENCIADOR DE TELA
ARQUIVO EM DISCO grava nodo no disco recupera nodo do disco	

3.2.3 Definição das Responsabilidades

As responsabilidades de cada classe são:

- classe Hipertexto: criar hipertexto, executar, finalizar hipertexto;
- classe Nodo: ter um ID, criar um nodo, inserir uma âncora, obter *link* relacionado com a âncora, retirar âncora, retirar nodo;
- classe Link: criar um *link*, obter o nodo apontado, retirar o *link*;
- classe Lista: colocar um elemento na lista, retirar um elemento da lista, verificar se a lista está vazia;
- classe Gerenciador de Tela: gerenciar a exibição do nodo, gerenciar a exibição das âncoras, gerenciar os ícones de edição, gerenciar a exibição do cursor, gerenciar o histórico de navegação, obter nodo visitado anteriormente;
- classe Pilha: colocar um elemento na pilha, retirar um elemento da pilha, verificar se a pilha está vazia;
- classe Gerador ID: gerar um novo ID para um nodo;
- classe Tela: exibir nodo, exibir âncoras, exibir cursor, exibir ícones;
- classe Tela de Edição: separar o texto em nodos, escolher as âncoras, escolher os nodos destino, nomear os nodos;
- classe Arquivo em Disco: gravar nodo no disco, recuperar nodo no disco.
- classe Ident: conter o identificador de um nodo.

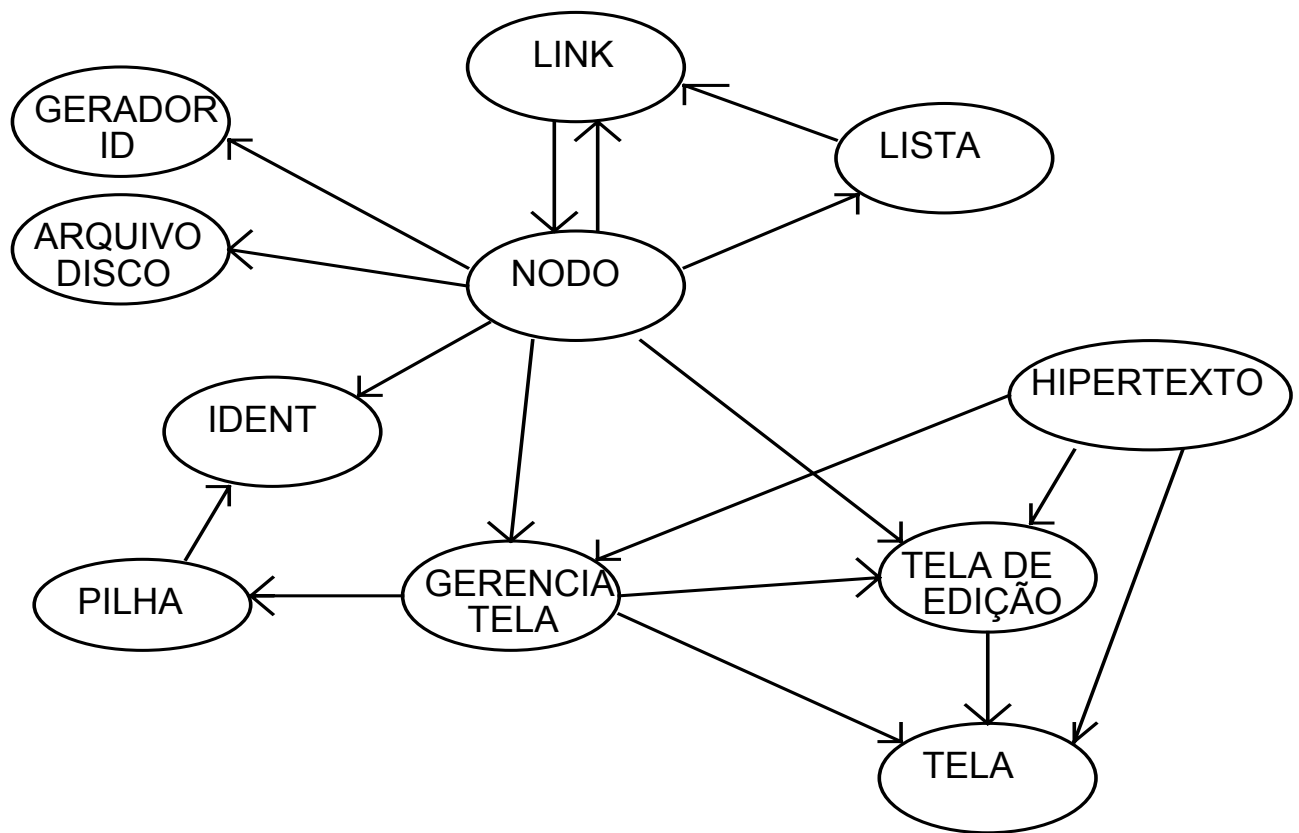
3.2.4 Definição dos Colaboradores

Os colaboradores das classes são as classes que recebem mensagens das classes do sistema. O diagrama da figura 3.2 mostra, através do sentido das setas, quais as classes que mandam mensagens e para quais classes são mandadas. Os colaboradores são:

- classe Hipertexto: classe Tela de Edição, classe Gerencia Tela de Edição, classe Tela;

- classe Nodo: classe Link, classe Gerador ID, classe Ident, classe Lista;
- classe Link: classe Arquivo em Disco, classe Nodo;
- classe Lista: classe Link;
- classe Tela de Edição: classe Nodo, classe Tela, classe Gerenciador de Tela;
- classe Pilha: classe Ident.

FIGURA 3.2: DIAGRAMA DE CLASSES DO SISTEMA SLIDE

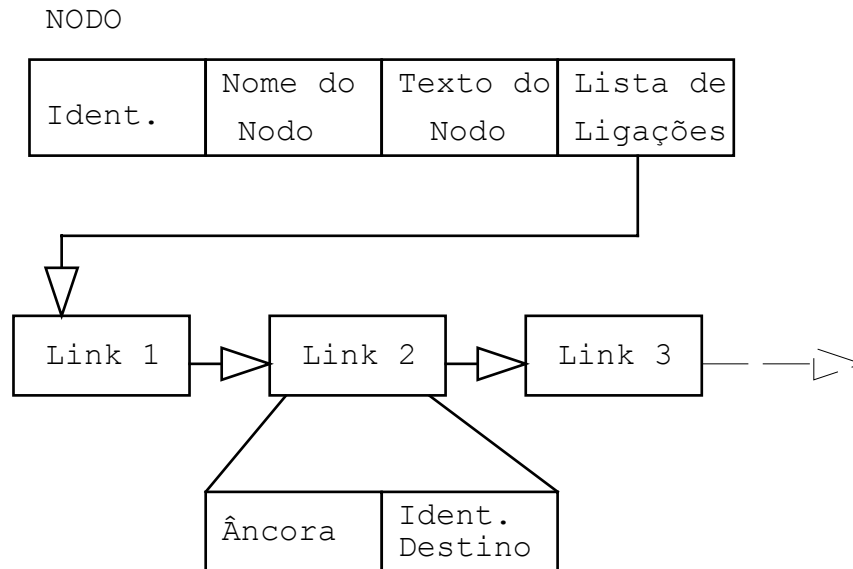


3.2.5. Estruturas de Dados.

A estrutura de dados usada pelo objeto *Nodo* é a seguinte: um *nodo* tem um objeto *Ident*, o texto do *nodo* e uma lista de objetos *Link*. Um objeto *Ident* tem um número identificador que é único para cada *Nodo*. Um objeto *Link* tem um objeto Âncora e o número identificador do *nodo* apontado por este objeto (figura 3.3). Um objeto *Ident* tem um número

identificador do nodo apontado por este objeto (figura 3.3). Um objeto Âncora tem a posição inicial da âncora na tela e a palavra-chave da âncora.

FIGURA 3.3: DIAGRAMA DA ESTRUTURA DE DADOS UTILIZADA



O objeto Pilha contém uma pilha de objetos Ident. Os objetos Ident vão sendo colocados na pilha à medida que os respectivos nodos vão sendo visitados para se saber o caminho percorrido durante a navegação pelo hiperdocumento.

3.3. IMPLEMENTAÇÃO DO SISTEMA.

A implementação do sistema, numa primeira versão, foi feita para micro-computadores IBM-PC compatíveis, utilizando somente ambiente operacional DOS e linguagem C++, mais especificamente o compilador Borland C++ 2.0 da Borland International. A versão atual foi desenvolvida tendo em vista o ambiente gráfico Windows 3.1, da Microsoft, e para isto foi utilizada a versão 3.1 do compilador C++ da Borland.

Para implementar o sistema foi escolhida a tecnologia orientada a objetos, como citado anteriormente. Deste modo houve grande preocupação com a utilização das características desta tecnologia, isto é, reutilização de código, herança de atributos,

polimorfismo, abstração e encapsulamento. Assim, para o desenvolvimento do sistema foi escolhida a biblioteca de classes CLASSLIB, fornecida junto com o compilador.

3.3.1. A biblioteca de Classes CLASSLIB

As bibliotecas de classes são fornecidas com o intuito de poupar ao programador o tempo que ele gastaria para criar independentemente ferramentas que são comuns a grande maioria dos projetos de desenvolvimento de software. Isto acontece devido a uma das grandes características da programação orientada a objetos, a reutilização de código.

A biblioteca CLASSLIB é uma coleção de classes desenvolvida pela Borland International e distribuída em conjunto com seu compilador C++, com o intuito de suprir necessidades comuns a grande parte dos sistemas de software.

O sistema SLiDe foi desenvolvido desde o início tendo em vista a utilização das facilidades fornecidas por esta biblioteca. Procuramos tirar o maior proveito possível das características e da estrutura das classes CLASSLIB.

As classes Lista (uma lista encadeada de objetos *Link*, para conter as ligações entre um nodo e outros) e Pilha (uma pilha de objetos *Ident*, visando a implementação do Histórico) foram desenvolvidas como sub-classes das classes pré-definidas Queue e Stack. Estas classes pré-definidas possuem todas as funções necessárias já implementadas, sendo necessário apenas usá-las corretamente. Para que isto pudesse ocorrer, foi preciso declarar as classes *Nodo* e *Link* como sub-classes da classe abstrata² Object.

3.3.2. A Implementação dos Objetos do Sistema SLiDe.

O sistema foi implementado através de objetos que o constituem. O principal objeto é o Hipertexto, que controla toda a inicialização, a carga e o início da navegação pelo hiperdocumento. Este objeto inicializa todos os outros objetos necessários (Objetos

²(Uma classe abstrata é uma classe que não possui Métodos ou membros, e serve apenas como base comum para toda uma hierarquia de classes que serão usadas juntas, como é o caso da classe Lista e da Classe Link.)

Arquivo_em_Disco, Gerenciador_de_Tela ou Tela_de_Edição, conforme o caso, e o Nodo que conterá as informações a serem apresentadas na tela).

Além disto, são também criados os objetos independentes, Gerador_de_ID, que gera identificações únicas para os nodos do hiperdocumento, e Pilha, que guarda uma pilha dos nodos visitados desde o início da navegação. É através deste objeto que pode-se retornar ao último nodo visitado, através da opção "Anterior", ou escolher qualquer dos nodos anteriormente visitados, através da opção "Histórico".

3.3.3. A Implementação das Ferramentas do Sistema SLiDe.

As ferramentas para facilitar a leitura, descritas na seção 3.1 foram implementadas como métodos das classes Hipertexto e Pilha utilizando algumas das estruturas de dados descritas na seção 3.2.1.

A Lista de Tópicos é um método da classe Hipertexto que obtém todos os identificadores de cada nodo do hiperdocumento e gera uma lista com todos os nodos, onde cada item leva diretamente ao nodo procurado.

Pesquisa é uma opção que permite ir diretamente a um nodo através do assunto sobre o qual ele trata.

Os comandos Anterior e Histórico são métodos da classe Pilha. Todos os nodos visitados são colocados na pilha. O comando Anterior permite retornar ao último nodo visitado e o Histórico mostra todos os nodos visitados, por ordem de visitação, e pode acessar qualquer nodo visitado.

Avançar e Voltar são métodos da classe Hipertexto que, usando a mesma lista da Lista de Tópicos, permitem visitar os nodos na ordem descrita acima.

Início é um método da classe Hipertexto que retorna o nodo inicial do hiperdocumento.

Browser é um método da classe hipertexto que, baseado em uma lista com todos os objetos *Link* do hiperdocumento, desenha um mapa com os nodos e em cada desenho de nodo coloca uma âncora para o mesmo.

4. - PROJETO DA INTERFACE.

Com os recentes avanços e a multiplicação da utilização do computador nas mais diferentes áreas de conhecimento, está se tornando cada vez mais importante o modo de interação homem-computador. O aumento constante da complexidade dos programas torna fundamental que a interface apresentada pelos computadores seja simples e de fácil compreensão. Os esforços na área estão convergindo para aumentar a ergonomia dos programas, ou seja, tornar sua utilização mais fácil e agradável. Para que a utilização do programa seja mais eficiente é importante que seja levada em consideração a população a quem se dirige o programa, ou seja, os usuários.

4.1. DEFINIÇÃO DA POPULAÇÃO USUÁRIA.

A definição da população usuária pode ser dividida em duas fases: a definição do modelo mental do usuário e a classificação da população usuária [EE89] [Nor87]. A classificação da população usuária neste trabalho segue a proposição de Vassiliou & Jarke [JV84]. O sistema tem basicamente dois tipos de usuários, o autor e o leitor.

O autor é o usuário que irá criar o hiperdocumento. Pode-se supor que o autor terá alguns conhecimentos de informática, já que ele está disposto a utilizar um sistema baseado em computador para escrever. O autor está classificado da seguinte maneira:

- Dimensão sintática - facilidade de uso do computador. O autor está classificado como habilitado por que ele tem pouca familiaridade com conceitos de programação e usa frequentemente o sistema (supõe-se que um

pesquisador está sempre acrescentando novas partes aos seus hiperdocumentos).

- Dimensão semântica - conhecimento a respeito do programa. O autor está classificado como gerencial por que ele utiliza uma ampla faixa de operações, pois para criar o hiperdocumento; o autor também tem um conhecimento geral da aplicação.

O leitor é o usuário que só irá ler o hiperdocumento. Ele não tem acesso ao módulo de edição do sistema. O leitor precisa ter algum conhecimento mínimo sobre o padrão de interface "Windows". O leitor está classificado da seguinte maneira:

- Dimensão sintática. O leitor é um usuário novato, pois tem baixa frequência de utilização e pouca familiaridade com os conceitos de programação. Supõe-se que o leitor irá usar o sistema esporadicamente.
- Dimensão semântica. O leitor é um usuário novato, pois usa uma faixa estreita das operações do sistema (apenas o suficiente para percorrer o hiperdocumento) e tem apenas um conhecimento geral sobre o sistema.

4.2 DESCRIÇÃO DA INTERAÇÃO COM O USUÁRIO.

O leitor terá uma interação bastante simples com o programa. As entradas que o usuário deve fornecer se resumem a:

- O nome do hiperdocumento a ser pesquisado;
- após a inicialização do sistema, a entrada do usuário consistirá apenas de movimentação do cursor (via teclado ou *mouse*) e subsequente seleção das ligações a serem seguidas dentro do hipertexto.

Algumas das ferramentas mencionadas no capítulo 3 (retornar ao último nodo visitado, seguir o documento linearmente, obter um mapa gráfico ou browser) serão colocadas apenas como ligações especiais que podem ser seguidas pelo usuário. As opções "pesquisa" e "retorno" a qualquer nodo anteriormente visitado serão apresentadas na forma de "menus",

para que o usuário possa escolher o assunto a ser pesquisado ou o nodo para onde se deve retornar. A opção destes "menus" será através das setas cursoras, através do *mouse*, digitando-se o assunto procurado ou o nome (identificação) do nodo para onde se deseja retornar.

O autor do hiperdocumento terá, além das capacidades de interação já citadas, algumas outras específicas para a criação de hiperdocumentos, a saber:

- A entrada do nome do arquivo contendo o texto a ser formatado em hipertexto (separado em nodos de informação e conectado através de ligações);
- O nome do arquivo que conterá o hipertexto criado;
- Capacidade de, através de um cursor texto ou o *mouse*, marcar as delimitações de um texto linear que formará um novo nodo do hipertexto;
- Capacidade de marcar uma palavra ou seqüência de palavras para que esta seja uma saída de ligação para um nodo que seja explicativo sobre esta palavra-chave;
- Capacidade de conectar uma palavra-chave previamente marcada com um nodo (fazer a ligação). Esta facilidade será fornecida através de uma escolha em um menu de opções, que conterá também opções para percorrer o hiperdocumento, percorrer os nodos e desmarcar palavras-chave;

4.3. DESCRIÇÃO DA INTERFACE.

A interface do sistema será implementada seguindo o padrão "Windows" (Microsoft Corporation). A escolha se deve ao fato deste padrão de interface ser aceito quase como um padrão internacional para os usuários de computadores tipo IBM-PC, devido à sua facilidade de aprendizado e utilização e devido à existência de bibliotecas, em diversas linguagens, o que facilita sua implementação.

O comando mais usado é o "Segue o *Link*". É o comando para seguir a ligação entre dois nodos e exibir o nodo destino.

Os outros comandos da interface seguem o padrão "Windows" e estão agrupados em quatro "menus":

- Menu arquivo: os comandos são: "Novo" para abrir um novo arquivo texto e transformá-lo em hiperdocumento, "Abrir" para abrir um hiperdocumento para leitura, "Imprimir" para imprimir a tela corrente e "Sair" para sair do sistema.
- Menu edição: os comandos são: "Cria Nodo" para acrescentar um nodo ao hiperdocumento, "Remove Nodo" para remover o nodo corrente, "Cria Âncora" para criar uma nova ligação entre dois nodos e "Remove Âncora" para remover uma âncora.
- Menu navegação: os comandos são: "Anterior" para colocar na tela o último nodo visitado, "Avançar >>" para mostrar o próximo nodo, seguindo a seqüência linear do hiperdocumento, "Voltar <<" para mostrar o nodo anterior da seqüência linear, "Início" para voltar ao nodo inicial, "Browser" para mostrar um "mapa" do hiperdocumento, "Tópicos" para aparecer um menu com todos os nodos do hiperdocumento e "Pesquisa" para procurar um determinado nodo.
- Menu ajuda: os comandos são: "Índice", que apresenta uma relação dos tópicos explicados no Help sobre este programa, "Teclado", que é uma tela padrão do Windows para auxílio no uso do teclado, "Comandos" apresenta uma tela inicial a partir da qual pode-se encontrar auxílio sobre todos os comandos do programa, "Procedimentos" explica quais são as seqüências de passos necessárias à determinadas atividades, tais como os passos para a criação de hiperdocumentos, e etc., "Usando o Help" que é uma tela padrão do Windows, e "Sobre", que apresenta informações sobre versão e título do programa.

O menu navegação, para facilitar o acesso, aparece na tela se o botão direito do *mouse* for pressionado com o cursor em qualquer lugar da janela, além da posição normal do menu.

A ajuda do sistema será no padrão Windows e em forma de hiperdocumento.

4.4 CENÁRIOS.

Cenários são as telas do programa, acompanhadas das regras para mudança entre estas telas. As regras para mudança de telas, neste padrão, são muito simples. Para mostrar um novo nodo seguindo uma ligação a partir de um nodo mostrado numa tela basta posicionar o cursor sobre a âncora e pressionar o botão esquerdo do *mouse*.

Para selecionar uma opção de qualquer menu aberto como nas figuras 4.3 a 4.8 basta posicionar o cursor sobre a opção desejada e pressionar o botão esquerdo do *mouse*. Se esse botão for pressionado em uma posição que não seja âncora ou menu não acontece nada.

FIG 4.1. TELA MOSTRANDO UM NODO

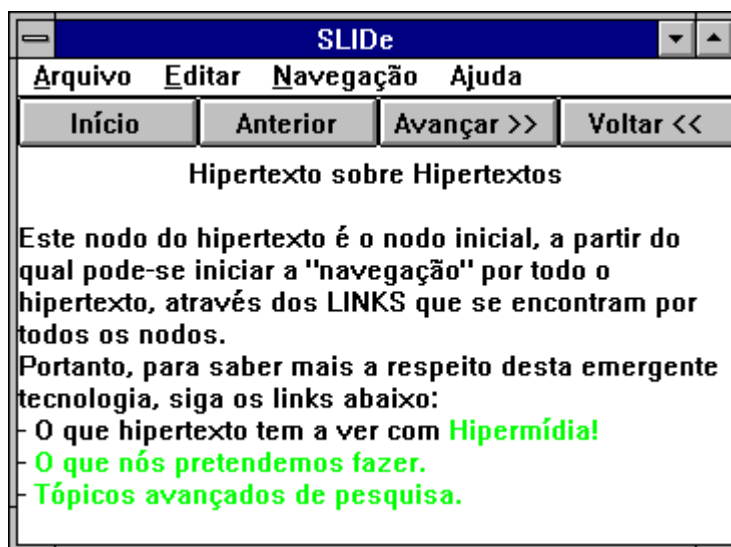


FIG 4.2. TELA MOSTRANDO UM NODO

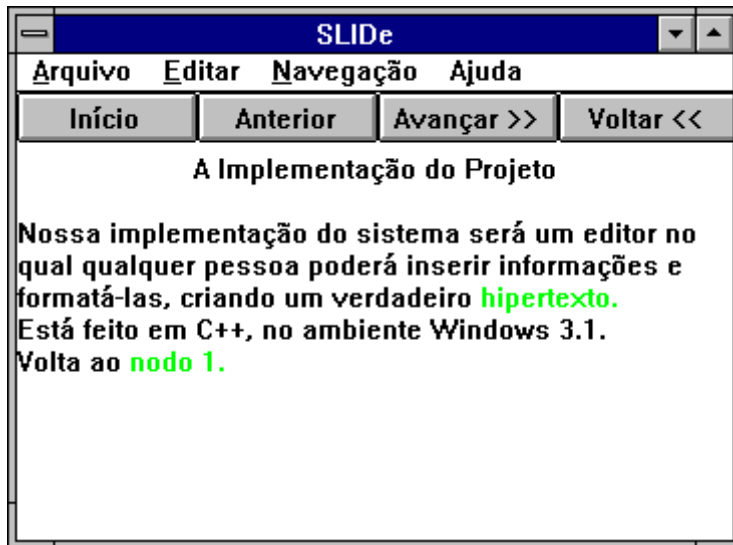


FIG 4.3. TELA MOSTRANDO O MENU "ARQUIVO" ABERTO

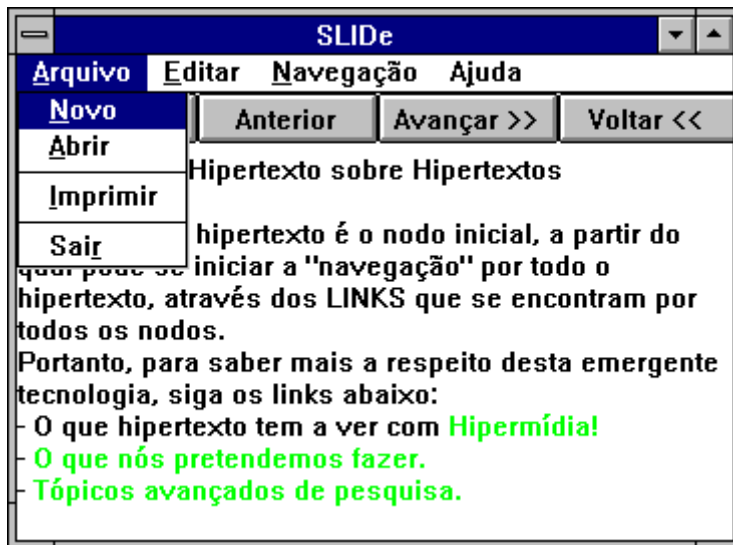


FIG 4.4. TELA MOSTRANDO O MENU "EDITAR" ABERTO

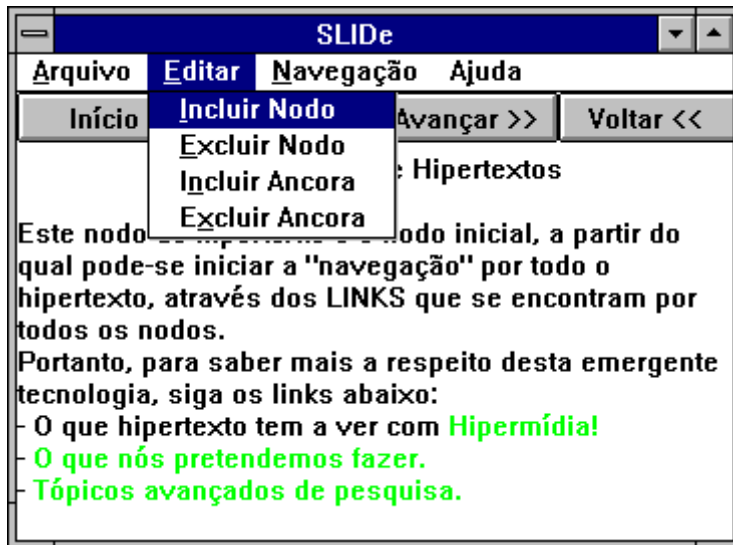


FIG 4.5. TELA MOSTRANDO O MENU "NAVEGAÇÃO" ABERTO NA POSIÇÃO NORMAL

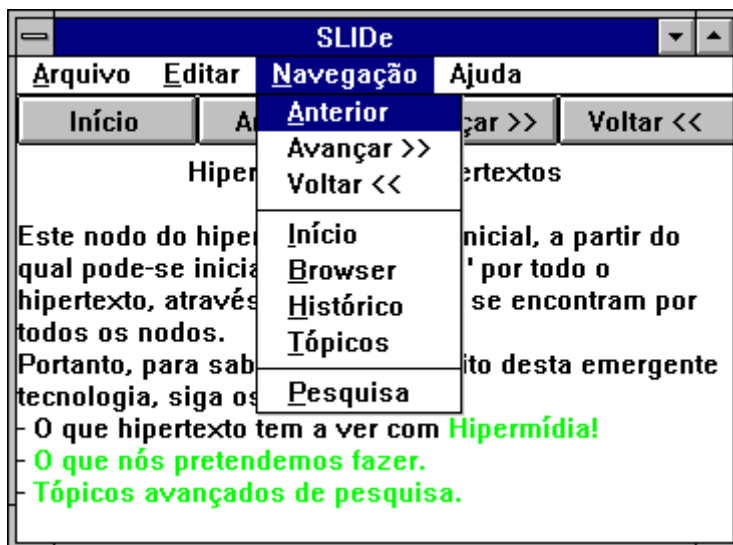


FIG 4.6. TELA MOSTRANDO A TELA DE DIÁLOGO REFERENTE AO COMANDO "TÓPICOS"

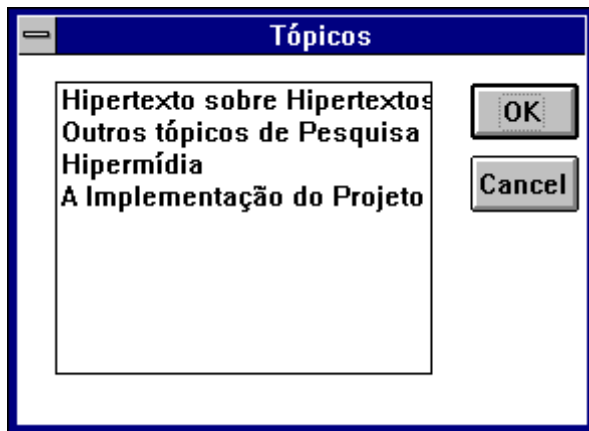


FIG 4.7. TELA MOSTRANDO O MENU "AJUDA" ABERTO

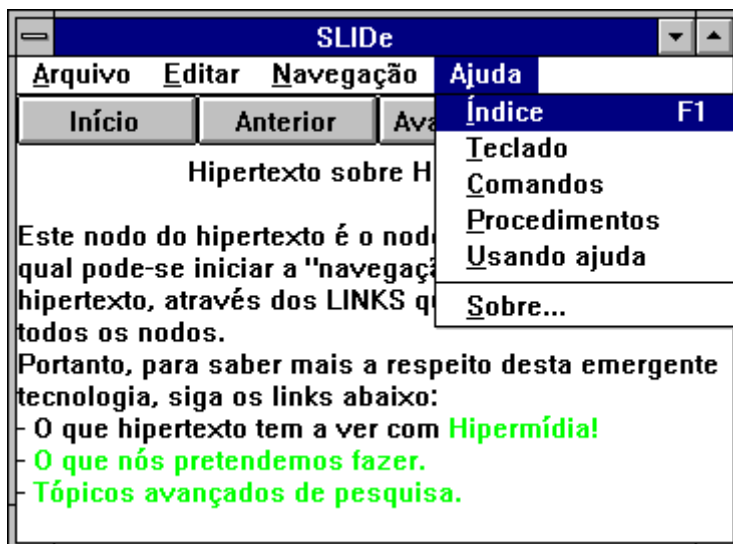
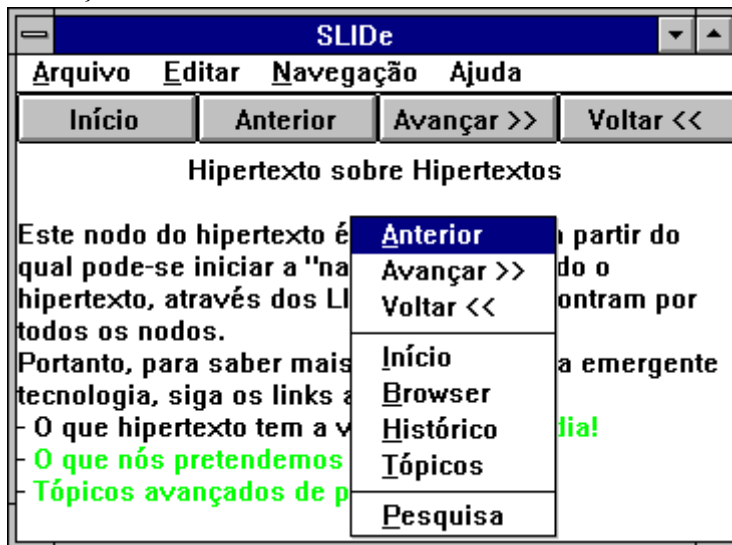


FIG 4.8. TELA MOSTRANDO O MENU NAVEGAÇÃO ABERTO EM QUALQUER POSIÇÃO DA TELA



4.5 IMPLEMENTAÇÃO DA INTERFACE

A implementação da interface foi realizado em linguagem C++, com a utilização do compilador "Borland C++ and Application Frameworks" da Borland International. Embora hoje a maioria do desenvolvimento seja realizado em C, em breve o C++ e a tecnologia de orientação a objetos deverão dominar o panorama de desenvolvimento. Seleccionamos as bibliotecas *ObjectWindows* (OWL) da Borland o qual nos fornece uma boa estrutura para a construção de aplicações para o ambiente gráfico Windows.

Uma característica adotada na programação Windows é a Nomenclatura Húngara [NY92] muito utilizada pelo programadores como uma convenção para a criação de nomes de variáveis e funções. A chave principal desta nomenclatura são os seus prefixos, como por exemplo: *cb* (contador de bytes), *ch* (caracter), *lp* (ponteiro long ou far) e *hwnd* (*handle* para uma janela).

Um *handle* é um inteiro sem sinal de 6 bits usado como identificador. Da mesma forma que num ambiente MS-DOS o *handle* identifica, por exemplo, um arquivo aberto, o Windows usa para identificar objetos como janelas, dispositivos de contexto, menus, etc.

As bibliotecas OWL seguem essa nomenclatura, iniciando as definições de classes com letras maiúsculas, notadamente a letra "T" de "tipo". As classes pré-definidas mais importantes são: TApplication, TControl, TDialog, TWindow e TWindowsObject.

4.5.1 Mecanismo de mensagens

O Windows é um sistema multitarefa não preemptivo, ou seja, os programas não são interrompidos pelo sistema operacional, mas se auto interrompem permitindo a execução dos outros programas. A implementação desse comutador de tarefas está embutido no núcleo do sistema.

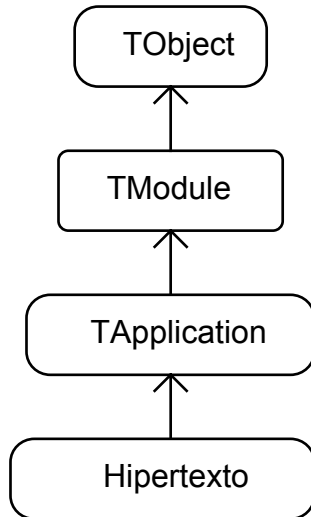
Os programas Windows recebem as entradas do usuário e dos objetos de interface com o usuário através das mensagens, por exemplo, o click do *mouse* ou uma tecla pressionada. As mensagens nesse ambiente são a unidade de processamento, cada programa possui o seu *loop* de controle de mensagens, permitindo que recebe e trate as mensagens.

Existem aproximadamente 250 mensagens pré-definidas no Windows. O conjunto de mensagens pode ser dividido em oito grupos: mensagens de Hardware, Manutenção de Janela, Interface com o Usuário, Privadas, Internas de Sistema, Notificação de Recurso de Sistema, Compartilhamento de Dados e de Término.

É através da definição de funções que tratem estas mensagens é que são confeccionados os objetos de interface necessários para o programa.

4.5.2. Objeto Aplicação

FIGURA 4.9: HIERARQUIA DO OBJETO HIPERTEXTO



Toda aplicação Windows possui uma função de entrada denominada WinMain, e através da biblioteca OWL devemos criar nosso próprio objeto aplicação, segundo a hierarquia acima (fig 4.9):

A OWL considera cada programa como um módulo, o Windows reconhece dois tipos de módulos: programas de aplicação e bibliotecas de ligação dinâmica (DLL). A classe TModule representa estes módulos, e contém 15 funções membros associadas.

A maioria das atividades de nível de aplicação é desenvolvida pela classe derivada de TModule, isto é, a classe TApplication. Esta classe é a classe parente imediata da classe do objeto da aplicação desenvolvida, TApplication trata a aplicação da mesma forma que uma aplicação típica do Windows, como a inicialização, criação da janela principal, e tratamento das mensagens no sistema.

A classe TApplication possui 16 funções membros, uma das mais importantes é MessageLoop() que instala o loop de mensagens para a nossa aplicação. Uma vez dentro do loop de mensagem, um programa Windows lê as mensagens e as envia para o local método correspondente, e o processamento da mensagem fica ao encargo das funções de cada janela. O objeto aplicação cria um objeto janela, apresenta a janela na tela e entra no loop de

mensagem. Após esta inicialização a função principal do objeto é manter as mensagens transitando entre o usuário e o objeto janela.

A classe Hipertexto é a própria instância do programa, sendo ela a responsável por gerenciar todo o controle das funções do programa.

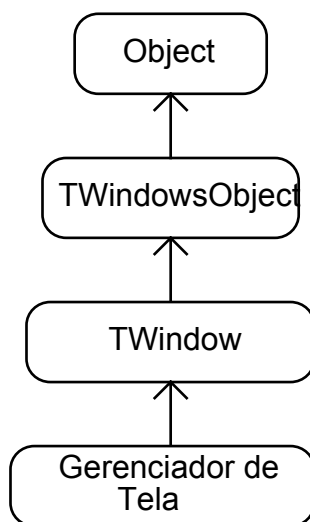
4.5.3. Objeto Janela

Além do objeto de aplicação, os aplicativos desenvolvidos com a OWL necessitam de objetos janela. Este objeto simplifica a interação com as bibliotecas do Windows através de suas classes extensíveis.

Este objeto é importante devido às características do ambiente de poder executar várias janelas ao mesmo tempo. Para cada instância deste objeto utilizada durante a execução, pode-se ter um hiperdocumento diferente, aumentando a capacidade elucidativa do sistema.

A hierarquia de classe do objeto janela segue abaixo:

FIGURA 4.10: HIERARQUIA DO OBJETO GERENCIADOR DE TELA



A OWL simplifica o tratamento das mensagens de janela, permitindo que você defina de um conjunto de funções membros de classe, ao invés de uma única função, para realizar todo o trabalho. Criando as funções de resposta de mensagens torna-se mais fácil a tarefa de escrever o código.

4.6 PORTABILIDADE DO SISTEMA

O sistema SLiDe foi desenvolvido inicialmente para ser utilizado em ambiente MS-DOS, aceitando somente informação através de texto. A versão atual foi desenvolvida utilizando o padrão de interface Windows. Durante o desenvolvimento do sistema, procurou-se fazer com que o núcleo e a interface fossem o mais independente possível, para que o sistema pudesse ser portado com facilidade para diversos ambientes como UNIX e MacIntosh, por exemplo. O fato da classe Tela ser a única responsável pela comunicação do núcleo com a interface torna muito mais fácil a aplicação deste conceito. É necessária apenas a reimplementação desta classe para passar de um ambiente para outro.

A continuação do trabalho deve incluir o porte do sistema para funcionamento sob o sistema UNIX, utilizando o sistema gráfico de janelas X-Window, segundo o padrão Motif.

5.CONCLUSÃO.

O trabalho apresentou os conceitos, os tipos, as aplicações e outras características de sistemas hipertexto e hipermídia. Foi descrita uma metodologia de projeto e implementação orientada a objetos, que foi utilizada para o desenvolvimento do sistema. Outro ponto importante foi a apresentação da especificação e da implementação da interface do sistema, elaborada de modo a tornar o sistema o mais amigável possível (*user friendly*).

Para fazer este trabalho, foram consultadas mais de 20 publicações sobre hipertexto, entre livros, teses de mestrado e artigos publicados. Toda esta bibliografia nos assegura que este trabalho está atualizado com as linhas de pesquisa sobre hipertexto. Uma das contribuições deste trabalho foi a reunião desta bibliografia, o que facilitará futuras pesquisas neste tema. Entretanto, a grande contribuição deste trabalho é o estudo de novas ferramentas que facilitam a interação do usuário com o sistema de hipertexto, visando eliminar ou minimizar o problema do usuário ficar "perdido no hiperespaço", descrito na seção 3.1.

Com as novas técnicas utilizadas no desenvolvimento, houve uma dificuldade inicial em adequar os meios utilizados para se chegar a solução dos problemas, já que todos estavam acostumados ao desenvolvimento estruturado e procedimental de programas. Além desta dificuldade de adaptação, e da escassez de informação e pessoas que entendessem mais a fundo do assunto de orientação à objetos, uma grande barreira encontrada foi durante a utilização das bibliotecas de classes. Com a realização deste trabalho conseguiu-se domina os conceitos de orientação a objetos, bem como técnicas de análise, projeto e programação orientadas a objetos. Este trabalho descreve a utilização destas técnicas, o que pode servir de base e orientação para trabalhos futuros. Outro mérito do trabalho é a utilização de técnicas de programação de interface utilizando o padrão MS-Windows.

Como sugestão para trabalhos futuros e avanços do sistema, temos a inclusão de técnicas de inteligência artificial para aumentar a capacidade de interação com o usuário. Outras sugestões seriam a inclusão de multimídia (sons e gráficos), e sua utilização como base para a construção de um sistema de ensino por computador.

REFERÊNCIAS BIBLIOGRÁFICAS

- [AMM88] Albuquerque, E. S.; Meira, S. L.; Melo, A. C.; Vasconcelos, A. *Conceitos e Características de Sistemas de Hipertexto*, Relatório técnico DI/UFPE 010/88, Universidade Federal de Pernambuco - Departamento de Informática, Recife, 1988.
- [AMY88] Aksyn, R., McCracken, D., and Yoder, E. *KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations*, Communications of the ACM, n. 7, vol. 31, pp 820-835, ACM, julho 1988
- [BC88] Begeman, M.; Conklin, J. *The Right Tool for the Job*. Byte, 13(10) pp 225-266, Outubro 1988
- [BC89] Beck, K.; Cunningham, W. *A Laboratory For Teaching Object-Oriented Thinking*, OOPSLA '89 Proceedings, pp 1-6, ACM, Outubro 1989.
- [Bus45] Bush, V. *As We May Think*, Atlantic Monthly, pp 101-108, julho 1945.
- [Con87] Conklin, J. *Hypertext: An Introduction and Survey*. Computer n. 20, pp17-41, Setembro 1987
- [Dam88] Van Dam, A. *Hypertext '87 Keynote Address*, Communications of the ACM, n. 7, vol. 31, pp 887-895, ACM, julho 1988

- [DS86] Delisle, N.; Schwartz, M. *Neptune: Hypertext System for CAD Applications*, Proceedings of ACM SIGMOD International Conference on Management of Data, ACM, Nova Iorque, pp 132-143, maio 86
- [EE68] Engelbart, D.C.; English, W.K. *Research Center For Augmenting Human Intellect*, AFIPS Conference Proceedings 33, pp 395-410, 1968.
- [EE89] Eberts, R. E.; Eberts, C.G. *Four Approaches to Human Computer Interaction. Intelligent Interfaces - Theory, Research and Design*. North-Holland, 1989.
- [FJP93] Fabro, J. A.; Jaime, P.; Pimentel, A. R. *SOLÃO: Sistema Não Linear Para Organização de Documentos - Um Sistema de Hipertexto e Sua Implementação*. Trabalho Submetido ao I Encontro Inter-Universitário de Informática do Paraná. Curitiba, 1993.
- [GM88] Marchionini, G. *Hypermedia and Learning: Freedom and Chaos* - Educational Technology Special Issue on Hypermedia - pp 7-12, Novembro 1988
- [Hal88] Halasz, F. *Reflections On Notecards: Seven Issues For The Next Generation Of Hypermedia Systems*, Communications of the ACM, n. 7, vol. 31, pp 836-852, ACM, julho 1988.
- [Her87] Hershey, W. *Guide*, Byte 12(11) pp244-246 Outubro 1987
- [Mar92a] Marques, E. V. *Sistema de Hipertexto para Acompanhamento da Produção Científica de um Departamento Universitário*, Dissertação de Mestrado, UFMG, 1992.
- [Mar92b] Martin, J. *Hiper Documentos E Como Criá-los*, Editora Campus, Rio de Janeiro, 1992.
- [Mur90] Murta, C. D. *Uma Interface Para Usuários Novatos do Sistema Patplus*. Tese de Mestrado. Departamento de Ciência da Computação da UFMG. Belo Horizonte, 1990.
- [Nel81] Nelson, T.H. *Literary Machines, Project Xanadu*, 1981, disponível com o autor.
- [Nie90] Nielsen, J. *Through Hypertext*, Communications of the ACM, n. 3, vol. 33, pp 298-310, ACM, março 1990.
- [Nor87] Norman, D. *Some Observations on Mental Models. in Readings in Human-Computer Interaction - A Multidisciplinary Approach*, Baecker, R.M. & Buxton, W.A.S, Eds., Morgan Kaufmann Publishers, Inc., 1987, pp. 241-244.
- [NY92] Norton, P.; Yao, P. *Programando em Borland C++ para Windows*, Editora Berkeley, Rio de Janeiro, 1992.
- [Pim89] Pimentel, M. G. *Sistemas Hipertexto: Discussões, Um Projeto e Sua Implementação*, Dissertação de Mestrado, USP - São Carlos, setembro 1989.
- [PN93] Puccetti, H.G.; Nasr, A. C. *Estrutura de Armazenamento e Recuperação de Informações de Dados*, Relatório Técnico, departamento de informática da Universidade Federal do Paraná, julho 1993.
- [SM86] Schneiderman, B.; Morariu, J. *The Interactive Encyclopedia System (TIES)*. Departamento de Ciência da Computação - Univ. de Maryland, College Park, Maryland, 1986
- [Str91] Stroustrup, B. *The C++ Programming Language*, Addison-Wesley Publishing Company, 1991.
- [Woo91] Woodhead, N. *Hypertext & Hypermedia - Theory and Applications*, Addison-Wesley Publishing Company, 1991.
- [WWW90] Wirfs-Brock, Wilkerson, Wiener, *Designing Object-Oriented Software*.
- [YHMD88] Yankelovich, N.; Haan, B.; Meyrowitz, N.; Drucker, S. *Intermedia: The Concept and the Construction of a Seamless Information Environment*. IEEE Computer, 21(1), pp 81-96, Janeiro 1988