UNIVERSIDADE REGIONAL DE BLUMENAU

CENTRO DE CIÊNCIAS EXATAS E NATURAIS

CURSO DE CIÊNCIAS DA COMPUTAÇÃO

(Bacharelado)

AMBIENTE PARA CRIAÇÃO DE MATERIAL DIDÁTICO MULTIMÍDIA

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

LÚCIA HELENA TRENTINI DOS SANTOS

BLUMENAU, JULHO/2001

2001/1-45

AMBIENTE PARA CRIAÇÃO DE MATERIAL DIDÁTICO MULTIMÍDIA

LÚCIA HELENA TRENTINI DOS SANTOS

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

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

_____

Prof. Carlos Eduardo Negrão Bizzotto — Orientador na FURB

_____

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

_____

Prof. Carlos Eduardo Negrão Bizzotto

_____

Prof. Dalton Solano dos Reis

_____

Prof. Oscar Dalfovo

AGRADECIMENTOS

Agradecimento especial às pessoas que compartilharam comigo mais esta etapa de minha vida. Em especial, a minha família, principalmente meus pais, Júlio César e Arlete, que me deram todo o incentivo necessário para sempre continuar e jamais desistir.

Agradeço também a todos os professores que no decorrer destes anos foram de total importância, com especial carinho ao professor Carlos Eduardo Negrão Bizzotto, meu orientador, que acreditou no meu trabalho.

Aos meus amigos, aos antigos amigos que ficaram meio esquecidos em virtude da faculdade, mas que em nenhum minuto deixaram de ser verdadeiros amigos, e aos amigos que surgiram na faculdade e passaram a fazer parte da minha vida, e já estão deixando saudades. Amigos que sempre incentivaram, e me deram força.

A todos dedico este trabalho de conclusão de curso.

SUMÁRIO

LISTA DE FIGURAS VI

RESUMO VIII

ABSTRACT IX

1 INTRODUÇÃO 1

1.1 OBJETIVOS DO TRABALHO 2

1.2 ESTRUTURA DO TRABALHO 3

2 TEORIAS DE APRENDIZAGEM 4

2.1 BEHAVIORISMO 4

2.1.1 FORMAÇÃO DOS HÁBITOS 4

2.1.2 EDUCAÇÃO BEHAVIORÍSTA 5

2.2 ABORDAGEM COGNITIVISTA 7

2.3 ABORDAGEM HUMANISTA 10

3 INFORMÁTICA NA EDUCAÇÃO 14

3.1 HISTÓRICO 14

3.2 GERAÇÕES 16

3.3 DIFERENTES USOS DO COMPUTADOR NA EDUCAÇÃO 17

3.4 TIPOS DE SOFTWARES 19

3.5 SOFTWARES EDUCACIONAIS 21

3.5.1 CARACTERÍSTICAS DE UM SOFTWARE EDUCACIONAL 22

3.6 PROFESSOR E A INFORMÁTICA 23

3.7 PONTOS POSITIVOS PROPORCIONADOS PELOS AMBIENTES DE INFORMÁTICA EDUCATIVA 24

4 METODOLOGIA 26

4.1 METODOLOGIA DE ORIENTAÇÃO A OBJETOS 26

4.2 HIPERMÍDIA 31
4.3 MÉTODO ORIENTADO A OBJETOS PARA DESIGN HIPERMÍDIA 33
4.4 DESIGN CONCEITUAL 35
4.5 DESIGN DE NAVEGAÇÃO 37
4.5.1 NODOS 38
4.5.2 LINK'S 39
4.5.3 ESTRUTURAS DE ACESSO 40
4.6 DESIGN DA INTERFACE ABSTRATA 42
4.6.1 ABSTRACT DATA VIEWS(ADV) 42
4.6.2 DIAGRAMAS DE CONFIGURAÇÃO 44
4.7 IMPLEMENTAÇÃO 44
4.8 AMBIENTE MACROMEDIA DIRECTOR 46
5 DESENVOLVIMENTO DO PROTÓTIPO 52
5.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO 52
5.2 ESPECIFICAÇÃO 52
5.3 IMPLEMENTAÇÃO 57
5.3.1 OPERACIONALIDADE DA IMPLEMENTAÇÃO 57
5.4 RESULTADOS E DISCUSSÃO 69
6 CONCLUSÕES 70
6.1 EXTENSÕES 70
REFERÊNCIAS BIBLIOGRÁFICAS 71

LISTA DE FIGURAS

FIGURA 2.1: RELAÇÃO ENTRE SUJEITO E OBJETO..... 9

FIGURA 3.1: ENSINO APRENDIZAGEM ATRAVÉS DO COMPUTADOR.....18

FIGURA 4.1: ESTRUTURA DE UMA CLASSE.....27

FIGURA 4.2: ANATOMIA DE UM OBJETO.....28

FIGURA 4.3: OBJETOS SE COMUNICANDO COM SOLICITAÇÕES..... 29

FIGURA 4.4: EXEMPLO DE HERANÇA – FIGURAS GRÁFICAS.....30

TABELA 4.5 :TERMOS UTILIZADOS EM ORIENTAÇÃO A OBJETOS E EQUIVALENTE EM LINGO.....31

FIGURA 4.6 : ATIVIDADES DA OOHMD.....34

FIGURA 4.7 : PARTE DO ESQUEMA CONCEITUAL.....36

FIGURA 4.8 :CARTÕES DE IDENTIFICAÇÃO.....37

FIGURA 4.9 : NÓS COM VISUAL SEMELHANTE AO DAS CLASSES CONCEITUAIS.....38

FIGURA 4.10 : CARTÕES DE NÓS..... 39

FIGURA 4.11 : DEMONSTRAÇÃO DOS COMPORTAMENTOS DO CONTEXTO DE NAVEGAÇÃO..... 41

FIGURA 4.12 : DEMONSTRAÇÃO DOS RELACIONAMENTOS ENTRE ADV'S E ADO'S.....43

FIGURA 4.13 :EXEMPLO DE HERANÇA DE ADV'S.....44

FIGURA 4.14: ELEMENTOS DO DIRECTOR.....47

FIGURA 4.15: HANDLER ON NEW ME……...50

FIGURA 4.16: CRIANDO OBJETO.....51

FIGURA 4.17: – ENVIANDO MENSAGEM PARA UM OBJETO QUE ESTÁ NA ACTORLIST.....51

FIGURA 5.1:DIAGRAMA DE CASO DE USO.....53

FIGURA 5.2 :DIAGRAMA DE CLASSES.....54

FIGURA 5.3: ESQUEMA CONCEITUAL..... 55

FIGURA 5.4 : ESQUEMA NAVEGACIONAL.....56

FIGURA 5.5 : TELA APRESENTAÇÃO..... 57

FIGURA 5.6 : CAIXA DE DIÁLOGO PARA DIGITAR NOME DA APRESENTAÇÃO…….....58

FIGURA 5.7: CAIXA DE SOLICITAÇÃO DO NOME DA PRIMEIRA TELA 58

FIGURA 5.8: TELA PRINCIPAL DO PROTÓTIPO..... 58

FIGURA 5.9 : BARRA DE FERRAMENTAS.....59

FIGURA 5.10 : INSERIR IMAGEM.....59

FIGURA 5.11 : CÓDIGO CORRESPONDENTE À FUNÇÃO INSERIR IMAGEM..... 60

FIGURA 5.12 : INSERIR TEXTO..... 60

FIGURA 5.13 : PRIMEIRA TELA.....61

FIGURA 5.14 : PROPRIEDADES NOVA TELA..... 61

FIGURA 5.15 : PARTE DO CÓDIGO UTILIZADO PARA INSERIR UMA NOVA TELA.....62

FIGURA 5.16 : AMBIENTE PARA CRIAÇÃO DE IMAGENS.....62

FIGURA 5.17 : TELA 02.....63

FIGURA 5.18 : TELA 03.....63

FIGURA 5.19 : PROPRIEDADES DO BOTÃO.....64

FIGURA 5.20 : FUNÇÕES DO BOTÃO.....64

FIGURA 5.21 : PARTE DO CÓDIGO PARA EXECUTAR AS FUNÇÕES DO BOTÃO....65

FIGURA 5.22 : TELA 02 COM BOTÕES, NO MODO VISUALIZAR.....66

FIGURA 5.23 : ENVIO DE E-MAIL.....66

FIGURA 5.24 : PARTE DO CÓDIGO PARA ABRIR APRESENTAÇÃO.....67

FIGURA 5.25 : IMAGEM MOSTRANDO REDIMENSIONAR.....68

FIGURA 5.26 : PARTE DO CÓDIGO PARA FUNÇÃO REDIMENSIONAR.....68

RESUMO

O presente trabalho de conclusão de curso está propondo a criação de um ambiente hipermídia, onde o professor possa criar o material didático de sua disciplina. O software proposto se baseia em 3 princípios básicos: construção, interação e cooperação. A partir do princípio da construção, o professor é o autor de seu próprio material. O professor pode interagir ativamente com o ambiente proposto, deixando de atuar como selecionador de softwares educacionais. Para o desenvolvimento do ambiente proposto será utilizado o software de autoria Macromedia Director 8.0, seguindo o paradigma da orientação a objetos. Como resultado ter-se-á um ambiente que possibilitará ao professor / aluno inserir diferentes tipos de mídias (texto, vídeo e imagem), inserir botões e enviar E-mail's.

ABSTRACT

The following project proposes a creation of a hypermedia element where the teacher can create a kit for his subject. The software bases in three basic principles: construction, interaction and co-operation. From the principle of construction, the teacher will be the author of his own kit. The teacher could interact actively with the element, without using the educational softwares anymore. To develop this element will be used a software made by macromedia director 8,0, who follows the paragon of the objects orientation. As result it will have an element that will permit the teacher/student insert many kinds of midia (text, video, image), insert buttons and send e-mails.

1 INTRODUÇÃO

Segundo Valente (1993), a introdução do computador na educação tem provocado uma verdadeira revolução na concepção do processo de ensino e aprendizagem. Primeiro, os computadores podem ser usados para ensinar. A quantidade de programas educacionais e as diferentes modalidades de uso do computador mostram que esta tecnologia pode ser bastante útil no processo de ensino-aprendizagem. Segundo, a análise desses programas mostra que, num primeiro momento, eles podem ser caracterizados como simplesmente uma versão computadorizada dos atuais métodos de ensino.

Ainda não existem avaliações definitivas quanto ao uso do computador como "máquina de ensino", o que existe são análises parciais que, de uma forma geral divulgam questões como: a necessidade de formação e atualização dos educadores, a tecnologia prende mais a atenção dos alunos, o computador torna mais fácil o aprendizado de disciplinas consideradas difíceis, como a física e a química, e aumenta o desempenho escolar. Um dos fatores primordiais para a obtenção do sucesso perante essa nova realidade educacional é a capacitação do professor perante essa nova realidade educacional. O professor deverá estar capacitado de tal forma que perceba como deve efetuar a integração da tecnologia com a sua proposta de ensino. Cabe a cada professor descobrir a sua própria forma de utilizar esta nova tecnologia, conforme o seu interesse educacional, pois não existe uma forma universal para a utilização dos computadores na sala de aula (Tajra, 1998).

De acordo com Salgado (1992), os sistemas multimídia são ferramentas que permitem a representação da realidade. Na verdade estas mudanças começaram com o surgimento dos microcomputadores e sua rápida aceitação pela sociedade, e continuaram com o aparecimento de novas tecnologias. Um sistema multimídia caracteriza-se pelo tipo de informação que é especificada, manipulada, editada e armazenada.

Segundo Ramos (1996), existem pessoas que não tem coragem de começar a usar os computadores porque pensam que precisam fazer alguns anos de cursos antes de conseguir algum resultado efetivo. Nesse caso não tem coragem de começar.

Atualmente, poucos são os ambientes que permitem ao professor a criação de seu próprio material didático, uma vez que o conteúdo já vem pronto. Assim, qualquer inclusão do conteúdo ou mesmo adaptação para a realidade dos alunos é dificultada.

Portanto, tem-se um problema sério: grande desenvolvimento da tecnologia com a disponibilização de centenas de títulos para os alunos, mas poucos trabalhos são desenvolvidos no sentido de colocar esta tecnologia à disposição do professor, de forma que ele mesmo possa criar seu material didático, utilizando as novas tecnologias.

Entre os softwares de autoria disponíveis no mercado nacional e em português pode-se citar: Visual Class, o Everest e o Hyperstudio (Tajra, 1998). Os softwares citados ainda não estão difundidos entre os professores, em função, principalmente, da dificuldade de compreensão das interfaces e das poucas possibilidades oferecidas. São softwares de difícil manuseio e entendimento.

Neste sentido, o presente trabalho de conclusão de curso propõe o desenvolvimento de um ambiente de autoria para que o professor possa desenvolver seus materiais didáticos. O software proposto é um “ambiente integrado”, no qual o participante tem à sua disposição ferramentas que vão desde a importação de textos, edição de imagens, até o envio de *E-mail’s*. Esta característica pode contribuir para diminuir as resistências dos professores, uma vez que eles não precisam utilizar várias ferramentas diferentes e complementares.

1.1 OBJETIVOS DO TRABALHO

O objetivo principal do trabalho proposto é o desenvolvimento e implementação de um ambiente baseado em computador que permite ao professor criar seu próprio material didático hipermídia.

Para que se possa atingir o objetivo geral, alguns objetivos específicos devem ser atendidos:

a) permitir que o professor insira textos, vídeos e imagens;
b) permitir a criação de botões que facilitam a navegação entre as telas;
c) tornar possível a comunicação entre professor e alunos através de *E-mail*.

1.2 ESTRUTURA DO TRABALHO

No capítulo 1 serão apresentados a introdução do trabalho, com os objetivos e a organização do texto.

O capítulo 2 apresentará as teorias da aprendizagem e sua importância no presente trabalho.

No capítulo 3 existe uma explanação sobre a informática na educação e os tipos de softwares atualmente utilizados.

O capítulo 4 apresentará a metodologia utilizada neste trabalho a OOHDM, e alguns conceitos sobre o software Macromedia Director e sua linguagem Lingo.

O capítulo 5 apresenta o desenvolvimento e implementação do protótipo do software.

O sexto capítulo faz análises conclusivas sobre o trabalho, suas limitações e possíveis extensões deste, que poderão ser realizadas.

2 TEORIAS DE APRENDIZAGEM

Muitas teorias surgiram para explicar de que forma ocorre o aprendizado. No presente capítulo, cada uma destas teorias será apresentada detalhadamente, como forma de fundamentar teoricamente o protótipo a ser desenvolvido.

2.1 BEHAVIORISMO

Segundo Rich (1975), os primeiros behavioristas acreditavam que a psicologia, para se tornar ciência, teria de focalizar sua atenção no comportamento aparente, em vez de se dedicar ao misterioso funcionamento interior da mente e da consciência. John B. Watson é considerado o pai do behaviorismo e é nos seus textos que se pode encontrar a nova abordagem que prometia fazer da psicologia uma ciência plenamente desenvolvida. O sistema de Watson, com base num modelo de estímulo-reação, era determinista. Watson acreditava que, conhecendo os estímulos, a psicologia podia prever a reação; por outro lado, se a reação fosse conhecida, a natureza do estímulo real podia ser especificada.

Watson afirma que o valor da psicologia behaviorista se baseia toda nos reflexos condicionados. O condicionamento, pois, aumentará a extensão do ambiente no qual o indivíduo vive. Cada estímulo pode condicionar várias respostas. Assim a criança à vista de certas pessoas (estímulo) sorri, bate palmas e estende os braços para uma carícia (resposta). Dias após, a mesma visão (estímulo presente) a faz chorar e lhe dá movimentos de fuga (novas respostas). As respostas podem ser condicionadas, e isso é que dará a riqueza de adaptação do homem (Modesti, 1959).

2.1.1 FORMAÇÃO DOS HÁBITOS

Todo o comportamento humano para Watson se desenvolverá por meio de condicionamento, único meio para formação dos hábitos, cujo produto final será a personalidade, escopo de todo o behaviorismo. Os seres humanos distinguiram-se naquilo que comumente se chama formação de hábitos, os hábitos propriamente ditos e os movimentos habituais. A diferença entre hábitos e movimentos habituais é que os movimentos habituais dependem do mecanismo de ação e da sua repetição caindo num automatismo mais ou menos consciente; ao passo que o hábito quer na formação quer na existência, é completamente independente de qualquer mecanismo nervoso ou automatismo (Modesti, 1959).

Seguindo a mesma linha de raciocínio Rich (1975), também afirma que a personalidade é considerada o produto acabado de sistemas de hábitos.

2.1.2 EDUCAÇÃO BEHAVIORÍSTA

Burrhus Frederic Skinner é um behaviorista contemporâneo que evita teorias dedutivas e emprega uma abordagem indutiva na sua experimentação. Suas investigações começam com dados empíricos (baseados na experiência) e prosseguem por tentativas para limitar generalizações em cadeia. Skinner rejeita qualquer método de investigação que não dependa de observação sensorial. Seu método desenvolve-se identificando claramente, em primeiro lugar, os dados observados. Em seguida, os dados são separados em classes e, então, são estabelecidas leis para indicar relações mútuas. Skinner tem sido uma figura influente na psicologia e no seu papel de liderança no desenvolvimento de materiais programados para escolas (Rich, 1975).

A utilização dos princípios do behaviorismo na educação gerou os “estudos programados” ou “instruções programadas”. Possui algumas características básicas:

a) apresentam a informação em seções breves. Os objetivos do curso, em termos de comportamento esperado do aluno, são claramente explicitados;
b) os conteúdos são divididos em unidades pequenas e bem definidas, e testam o estudante após cada seção;
c) as avaliações são constantes, e apresentam feedback imediato para as respostas dos estudantes;
d) o conteúdo é planejado de forma que o aprendizado ocorra “do simples para o complexo”.

De acordo com esta teoria, acreditava-se que confirmando as respostas corretas e enfraquecendo as respostas que não deveriam ser adquiridas, uma máquina de auto-instrução poderia ensinar.

Toda a revolução que Watson quis introduzir no campo psicológico, não foi pelo fato de julgar a psicologia apoiada em falsos fundamentos da introspecção e querer corrigi-la; mas o fim proposto à sua revolução foi o de corrigir a velha psicologia para unicamente preparar as bases de uma verdadeira educação das crianças, para transformar este mundo em um lugar

digno da existência humana. Para ele a educação apoiada em sua psicologia *"haverá de formar uma geração feliz que ocupará o nosso lugar, vivendo uma vida mais digna que a nossa, tão cheia de erros e infelicidades por causa da educação não behaviorista"*. A educação dará ao indivíduo, por meio dos estímulos bem dosados, aquele domínio sobre as suas atividades viscerais, manuais e a atividade verbal poderá regularizar todo o seu comportamento. A esse domínio de si é que Watson chama de personalidade. Afirma ainda, que "*os homens nascem iguais, e os adultos se diferenciam por causa da educação, que por meio do ambiente e do condicionamento construiu a curva de atividade de cada um"*. Para o behaviorista tudo o que o homem fará será produto da aprendizagem (Modesti, 1959).

O behaviorismo tem grande influência na educação, pois o aluno é tratado como um agente passivo que recebe informações (conteúdo), de uma fonte (professor). O behaviorismo teve e tem até hoje, uma importância capital na educação de um modo geral. De acordo com Ramos (1996), "*... das múltiplas escolas psicológicas existentes, pode-se dizer que esta foi a que conformou, de forma mais definitiva, a prática pedagógica do mundo ocidental...*".

Com base no exposto, pode-se concluir que o behaviorismo possui os seguintes princípios básicos (Bizzotto, 1999):

a) heteronomia: o indivíduo é subordinado a um poder externo, ou seja, não existe autonomia;
b) auto-didatismo: o material didático é elaborado de forma a privilegiar o estudo individual. Portanto, não existe interação entre os participantes (aluno-aluno ou aluno-professor);
c) reprodução: o aluno não constrói nada de novo. Ele apenas reproduz o conteúdo repassado pelo professor;
d) massificação: o conteúdo não é adaptado às diferenças individuais, sendo o mesmo para todos os alunos. Neste sentido, o material didático é elaborado antes mesmo de se conhecer os alunos.

2.2 ABORDAGEM COGNITIVISTA

Ao contrário do Behaviorismo, o cognitivismo não se baseia na transmissão de conteúdo do professor ao aluno. A ênfase é sobre a pesquisa, a investigação e a solução de problemas por parte do próprio aluno.

A abordagem cognitivista se preocupa com a capacidade do indivíduo em integrar e processar informações. Os cognitivistas investigam os "processos centrais" do indivíduo, tais como: aquisição de conceitos, processamento de informações, tomadas de decisão, resolução de problemas.

Conforme ressalta Ramos (1996), *"...se não é o ambiente o único responsável, se o sujeito como corpo, mente e consciência também tem parte ativa no processo de desenvolvimento, então é na interação do sujeito com o ambiente que o desenvolvimento se dá"*. Este é o entendimento da corrente interacionista que surgiu no início deste século.

De acordo com Dupuy (1995), se o que a mente faz é simular o "ambiente", todo conhecimento é simulação, seja ele científico ou não. E simular a mente é criar modelos que simulam esse "ambiente" e interagem com ele.Uma forma de testar essa capacidade de simulação de um modelo da mente humana é o teste de Turing. Imagine uma sala onde está um observador que será o juiz. Este observador recebe mensagens de uma mulher. Um homem tenta de todas as formas simular as respostas de uma mulher e o juiz que tem acesso somente às mensagens, deve julgar se elas provêm do homem ou da mulher. Em seguida este homem é substituído por uma máquina, que tentará fazer o mesmo. Se o juiz não for capaz de distinguir entre as respostas da máquina e as da mulher então esta máquina passou no teste, e é um simulador perfeito da mente humana. Observando que neste teste o que a máquina simula não é simplesmente o comportamento da mulher, mas a capacidade de simulação do homem. Ou seja, o melhor modelo da mente seria aquele que é capaz de simular a sua capacidade de simulação.

O construtivismo de Jean Piaget é considerado por muitos autores como de uma abordagem Cognitivista. É assim que a professora Mizukami (1986) a classifica. Embora Piaget, não tenha desenvolvido uma teoria de aprendizagem, sua teoria epistemológica de como, quando e porque o conhecimento se constrói obteve grande repercussão na área educacional. Conforme ressalta Fialho (1998), *"...Piaget pressupunha o conhecimento*

humano como sendo adquirido através do processo de regulação e de equilibrío, como condições básicas para se viabilizar a adaptação e a inteligência, no momento que considera os fatores biológicos, psicológicos e sociológicos como vitais na construção do conhecimento, sobre a ótica de um sistema aberto, que propicia um inter-relacionamento contínuo do organismo com o ambiente em que se encontra inserido, através de um processo de trocas de influências equivalentes".

De acordo com Matui (1996), a epistemologia genética é o estudo da gênese e desenvolvimento das estruturas lógicas do sujeito em interação com o objeto de aprendizagem, ou seja, o estudo do processo de construção dos conhecimentos.

Conforme ressalta Varela (1997), Jean Piaget procurou explicar o desenvolvimento da criança, desde um organismo biologicamente imaturo no nascimento para um ser com argumentação abstrata quando adulto. A criança começa apenas com seu sistema sensório-motor e Piaget desejava entender como a inteligência sensório-motor evolui para uma concepção da criança com relação ao mundo externo, com objetos permanentes localizados no espaço e no tempo e na concepção da criança de si mesma tanto como um objeto dentre outros objetos como uma mente interna.

Um aspecto importante na obra de Piaget é que a construção orgânica das estruturas mentais se dá somente através das trocas com o meio. Conforme afirma Piaget, "*o esquema é a condição primeira da ação, ou seja, a troca do organismo com o meio. Ele é encontrado pelo funcionamento geral de todo o ser vivo: a adaptação do organismo com sua bagagem hereditária, em contato com o meio, perturba-se, desequilibra-se e, para superar este desequilíbrio, ou seja, para se adaptar, constrói os esquemas*" (Taylller, 1993).

A obra de Piaget é no mínimo consagrada a dois propósitos:

1. descrever estruturas psicológica que nos possibilitam o conhecimento(no sentido científico).
2. descrever processos que nos direcionam para estruturas cada vez mais elaboradas quanto a essa tarefa de "conhecer".

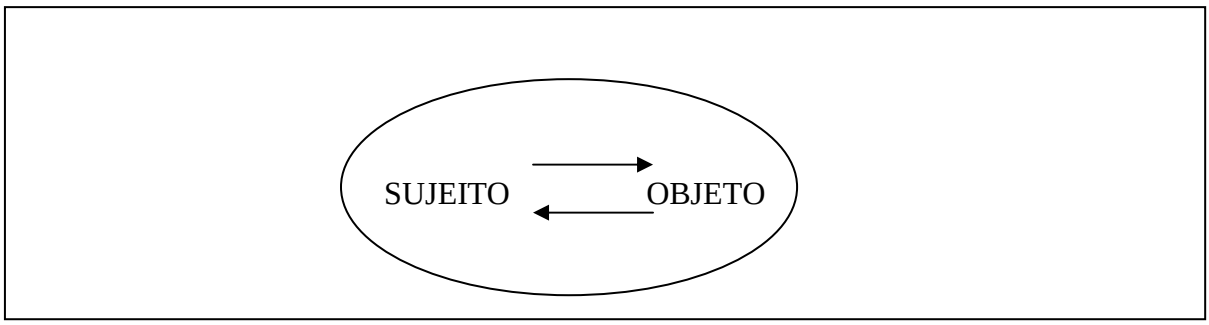
Estas estruturas e processos não são herdadas, ou seja, não estão pré-formadas, nem são recebidas de fora do sujeito, ou seja, de uma fonte exógena, mas são construídas por ele,

interagindo com objetos e pessoas, tornando-se possíveis e necessárias em função desta troca. Este conhecimento, produto de nossa ação sobre objetos e pessoas, pode ser subdividido em, pelo menos, três aspectos: conteúdos, procedimentos e operações(estruturas) (Alencar, 1995):

a) construir conteúdos: o que são os objetos, de que partes se compõem, suas propriedades, seu nome, etc. Constitui-se naquilo que Piaget designa por “esquemas presentativos”. Refere-se a modos organizados de assimilar objetos e pessoas quanto àquilo que eles são, na perspectiva do sujeito. Conceitos, noções, imagens, características dos objetos e pessoas, independente do contexto em que estão, “pertencem” a estes objetos e pessoas, e constituem os esquemas presentativos.
b) conhecer procedimentos: como produzir um resultado sobre objetos e pessoas, constitui-se naquilo que Piaget designa por “esquemas procedimentais”. Refere-se a modos organizados de produzir resultados sobre objetos e pessoas. O procedimento é algo de ordem espaço-temporal e depende de um conjunto de circunstâncias específicas, de uma dada situação.
c) contruir operações: por que agir ou pensar deste modo e não de outro sobre objetos e pessoas específicas. Constitui-se naquilo que Piaget designa por “esquemas operatórios”. Refere-se a modos organizados de estabelecer relações, correspondências, morfismos, entre ações ou objetos tais que definam um sentido ou lei de composição que os estruturam como algo inteiramente necessário. Por isso os esquemas operatórios sintetizam os esquemas de procedimentos (enquanto estruturas) e os presentativos (enquanto conceitos ou noções).

No construtivismo de Jean Piaget o sujeito e o objeto não são estruturas separadas, mas constituem uma só estrutura pela interação recíproca. O sujeito não existe sem o objeto nem o objeto (meio), sem o sujeito, conforme figura 2.1.

Figura 2.1: Relação entre sujeito e objeto



Fonte:adaptado de Matui (1996).

Segundo Matui (1996), o construtivismo é a notícia mais auspiciosa que a educação está recebendo. O ensino nas escolas fugindo do verbalismo tradicional caiu na robotização dos livros didáticos. No atual ambiente de socialização do saber, é grande a cobrança por abertura e novas alternativas. O construtivismo vem atender a essas aspirações. No construtivismo pode-se dizer que a interação constrói o próprio ser humano. Portanto construir significa promover a interação do sujeito com o meio. O processo desta construção é a aprendizagem, o mecanismo de "como" se realiza a interação construtiva. O conhecimento, que, na escola, é conteúdo das matérias, assume importância especial no construtivismo, em razão do compromisso de garantir a construção do saber dos alunos, notadamente das classes populares, visando uma apreensão global e não parcelada. É um grande mal-entendido julgar que o construtivismo é pelo empobrecimento e mal-entendimento dos conteúdos, pois se o aluno constrói algo na escola, isso só pode ser conhecimento.

Assim, com base no que foi exposto, pode-se relacionar como princípios utilizados pela abordagem cognitivista (Bizzotto, 1999):

a) autonomia: fundamentada na igualdade e na reciprocidade dos parceiros, libertando-se tanto na anomia ao egocentrismo quanto da heteronomia própria da coação;
b) interação: a comunicação é multidirecionada, ocorrendo tanto entre professor e aluno quanto entre alunos;
c) construção: a ênfase não é sobre a reprodução, mas sim na construção de artefatos que sejam significativos para o aluno;
d) personalização: o processo de aprendizagem leva em consideração as características específicas de cada aluno. Assim, ao invés de um material didático massificado, existe a construção do conteúdo ao longo do processo;
e) previsibilidade: esta abordagem se baseia na existência de um observador independente em um mundo pré-dado.

2.3 ABORDAGEM HUMANISTA

De acordo com Rich (1975), a psicologia humanística ou humanista difere do behaviorismo pelo fato de empregar uma abordagem organísmica e holística no estudo do homem. Tais estudos têm por intenção ver o homem como uma pessoa inteira que formula

objetivos e propósitos e que procura alcança-los. A força da psicologia humanística reside na sua disposição para ver o homem como um todo, suas escolhas, sua busca de finalidade, criatividade e tentativas para utilizar todos os seus potenciais.

Dentre os autores associados a esta abordagem, podemos destacar o trabalho de Paulo Freire. De acordo com sua proposta, toda pedagogia é subjacente a um conceito de homem e de mundo.

Conforme ressalta Fialho (1998), *"...é o homem um ser de adaptação ao mundo? Ou, é o homem um ser de transformação do mundo? Para Paulo Freire o homem é um ser no mundo e com o mundo. Um ser capaz de admirar o mundo objetivando-o, transcedendo-o através da sua consciência"*.

A proposta de Freire parte do estudo da realidade(fala do educando), e a organização dos dados(fala do educador). Nesse processo surgem os Temas Geradores, extraídos da problematização da prática de vida dos educandos. Os conteúdos de ensino são resultados de uma metodologia dialógica. Cada pessoa, cada grupo envolvido na ação pedagógica dispõe em si próprio, ainda que de forma rudimentar, os conteúdos necessários dos quais se parte. O importante não é transmitir conteúdos específicos, mas despertar uma nova forma de relação com a experiência vivida. A transmissão de conteúdos estruturados fora do contexto social do educando é considerada "invasão cultural" ou "depósito de informações" porque não emerge do saber popular. Portanto, antes de qualquer coisa, é preciso conhecer o aluno. Conhecê-lo enquanto indivíduo inserido num contexto social de onde deverá sair o "conteúdo" a ser trabalhado.

Assim sendo, não se admite uma prática metodológica com um programa previamente estruturado assim como qualquer tipo de exercícios mecânicos para verificação da aprendizagem, formas essas próprias da "educação bancária", onde o saber do professor é depositado no aluno, práticas essas domesticadoras. O relacionamento educador-educando nessa perspectiva se estabelece na horizontalidade onde juntos se posicionam como sujeitos do ato do conhecimento. Eliminam-se, portanto toda relação de autoridade uma vez que essa prática inviabiliza o trabalho de criticidade e conscientização (Feitosa, 1999).

Conforme ressalta Fialho (1998), Paulo Freire afirma que outra questão importante diz respeito aos "*conteúdos programáticos da educação*". A primeira constatação que faz é a de que toda prática educativa implica sempre a existência de sujeitos, aquele ou aquela que ensina e aprende e aquele ou aquela que, em situação de aprendiz, ensina também, a existência do objeto a ser ensinado e aprendido, a ser reconhecido e conhecido, o conteúdo final. Não centra a prática educativa, por exemplo, nem no educando, nem no educador, nem no conteúdo, nem nos métodos, mas a compreende nas relações de seus vários componentes, no uso coerente por parte do educador ou da educadora dos materiais, dos métodos, das técnicas.

A concepção de educação de Paulo Freire percebe o homem como um ser autônomo**.** Esta autonomia está presente na definição de vocação ontológica de *'ser mais'* que está associada com a capacidade de transformar o mundo. É exatamente aí que o homem se diferencia do animal. Por viver num presente indiferenciado e por não se perceber como um ser unitário distinto do mundo, o animal não tem história.

Segundo Bello (2000), para Paulo Freire o diálogo é o elemento chave onde o professor e aluno sejam sujeitos atuantes. Sendo estabelecido o diálogo processar-se-á a conscientização porque:

a) é horizontalidade, igualdade em que todos procuram pensar e agir criticamente;
b) parte da linguagem comum que exprime o pensamento que é sempre um pensar a partir de uma realidade concreta. A linguagem comum é captada no próprio meio onde vai ser executada a sua ação pedagógica;
c) funde-se no amor que busca a síntese das reflexões e das ações de elite versus povo e não a conquista, a dominação de um pelo outro;
d) exige humildade, colocando-se a elite em igualdade com o povo para aprender e ensinar, porque percebe que todos os sujeitos do diálogo sabem e ignoram sempre, sem nunca chegar ao ponto do saber absoluto, como jamais se encontram na absoluta ignorância;
e) traduz a fé na historicidade de todos os homens como construtores do mundo;
f) implica na esperança de que nesse encontro pedagógico sejam vislumbrados meios de tornar o amanhã melhor para todos e

g) supõe paciência de amadurecer o povo, de modo que a reflexão e a ação sejam realmente sínteses elaboradas com o povo.

Desta forma pode-se concluir que a proposta humanista de Paulo Freire se baseia em alguns princípios, dentre os quais pode-se destacar (Bizzotto, 1999):

a) autonomia: fundamentada na igualdade e na reciprocidade dos parceiros, libertando-se tanto na anomia própria ao egocentrismo quanto a heteronomia própria da coação;
b) interação: a comunicação é multidirecional, ocorrendo tanto entre professor e aluno quanto entre alunos;
c) construção: a ênfase não é sobre a reprodução, mas sim na construção de artefatos que sejam significativos para o aluno;
d) personalização: o processo de aprendizagem leva em consideração as características de cada aluno. Assim, ao invés de um material didático massificado, existe a construção do conteúdo ao longo do processo;
e) cooperação: o ser humano isolado, não chegaria jamais a conhecer, sendo essencial a comunhão entre as pessoas.

Este trabalho de conclusão de curso está baseado em três princípios: construção, interação e cooperação, que se adequaram na abordagem humanista de Paulo Freire.

O uso do ambiente permitirá a personalização do mesmo por parte do professor, que terá a capacidade de poder construir sua apresentação da forma que desejar, utilizando textos, imagens, vídeos, o professor será o autor do seu próprio material. O princípio da cooperação está ligado mais diretamente com a troca de informações entre professor e aluno, que pode ser feita tanto pela apresentação desenvolvida pelo professor, através de discussões e explanações sobre o assunto, como através de E-mail’s, que permitirá uma troca mais direta de informações. O princípio da interação diz respeito tanto a interação entre o professor e o programa como a interação através do programa entre professores e alunos. Permitirá também a autonomia que será obtida pelo aluno, através da utilização do ambiente.

3 INFORMÁTICA NA EDUCAÇÃO

A utilização da informática na educação vem crescendo e ampliando seu raio de atuação, embora algumas questões não só educacionais, mas de ordem social tenham sido levantadas. A produção e o uso do conhecimento em países de terceiro mundo apontam que há necessidade de melhorar a capacitação tecnológica, sem, no entanto, investir nas formas de aquisição do novo conhecimento. Recomendações de especialistas, como King, Market, Caillods, Salm, evidenciam que a tecnologia deve ser inserida no currículo regular das escolas do sistema de educação formal, a fim de que seja iniciada a capacitação tecnológica, simultaneamente à formação geral do aluno. Estudos nessa área vem apontando para a rápida mudança tecnológica e mostrando ser necessário que o indivíduo possua uma base sólida de conhecimentos (Campos,1994).

3.1 HISTÓRICO

De acordo com Lucena (2000), parte das dificuldades para a utilização de recursos de informática na educação e de uma certa resistência dos seus profissionais em incorporar as novas tecnologias está ligada à própria história da tentativa de implantação de informática nas atividades das escolas em nosso país. Em primeiro lugar, porque para o educador, qualquer atividade que envolva computadores aparece como herdeira e continuadora da política de “Tecnologia Educacional” impulsionada a partir do final da década de 1960, quando se procurou levar a escola a um “funcionamento racional de formação de mão de obra”. Naquela época era flagrante a crise que se manifestava na estrutura, no conteúdo e nos métodos arcaicos de uma prática educativa que não respondia às necessidades de desenvolvimento do país. O que marca as iniciativas nesta área é a supervalorização da Tecnologia Educacional, como solução apresentada para os problemas da educação brasileira.

Na década de 1980, o uso das tecnologias educacionais voltou a ser re-valorizado, agora tendo o computador como um dos seus instrumentos centrais. Como no final da década de 60, essa nova experiência não partiu da decisão dos educadores, mas da iniciativa de altos escalões do governo brasileiro.

Nos anos de 1981 e 1982, foram realizados o I e II Seminário Nacional de Informática na Educação envolvendo, pela primeira vez, pessoas ligadas diretamente ao processo educacional.

Estes seminários mostraram uma comunidade educativa preocupada com as experiências anteriores de tecnologia educacional.

Entre as recomendações do seminário, destaca-se.(Lucena, 2000):

a) a preocupação em afirmar que investir em tecnologia educacional não era resposta aos problemas que vivia a educação;
b) a preocupação quanto ao uso indiscriminado de programas estrangeiros que pudessem influenciar os conceitos e padrões culturais nacionais;
c) a preocupação quanto ao investimento em máquinas, apenas para "satisfazer os interesses do mercado, principalmente ao considerar que, no Brasil, o início do uso dos computadores nas escolas públicas ocorreu no momento em que se investia no crescimento e no favorecimento das industrias brasileiras de informática".

Para Lucena (2000), "*as experiências nos anos seguintes foram frustrantes*". A descontinuidade dos recursos, a ausência de suporte, a falta de preparação das equipes constituídas, a ausência de discussão e de participação mais ampla de professores e estudantes e a falta de infra-estrutura, terminaram por paralisar e isolar os laboratórios no interior da escola.

Acrescente-se às dificuldades o fato de professores, alunos e funcionários que até aqui haviam sido marginalizados do processo de elaboração e decisão sobre estas políticas verem, agora, seu papel reduzido a mero aplicadores de decisões preestabelecidas, cobrados por uma política e iniciativas das quais não tinham sido constituintes.

Os computadores que chegavam às escolas provocavam reações contraditórias: por um lado, eram vistos como algo que poderia melhorar e facilitar o trabalho, por outro, vistos como improdutivos como um desperdício do dinheiro que faltava em tantas coisas "mais importantes".

Segundo Angioletti (1995), a partir de 1990, foram criados os Centros de Informática Educativa para atender às escolas, garantindo tecnologia da informática para a população.

Estes centros tinham o objetivo de promover a formação dos professores para que estes soubessem utilizar as novas tecnologias disponíveis.

Um dos primeiros programas educacionais desenvolvidos no Brasil que visam a introdução de novas tecnologias de informação e comunicação na escola pública como ferramenta de apoio no processo ensino-aprendizagem é o ProInfo (Programa Nacional de Informática na Educação). O ProInfo é uma iniciativa do Ministério da Educação, por meio da Secretaria de Educação a distância – SEED criado pela Portaria nº 522, de 09 de abril de 1997, sendo desenvolvido em parceria com os governos estaduais e alguns municipais. O ProInfo tem na preparação de recursos humanos os professores, como sua principal condição de sucesso. Os professores são capacitados em dois níveis: multiplicadores e de escolas.

O professor - multiplicador é um especialista em capacitação de professores para o uso da telemática em sala de aula: adota-se no programa, portanto, o princípio professor capacitando professor. Os multiplicadores capacitam os professores das escolas nas bases tecnológicas do ProInfo nos estados, os Núcleos de Tecnologia Educacional (NTE), que são estruturas descentralizadas de apoio ao processo de informatização das escolas, auxiliando tanto no processo de planejamento e incorporação das novas tecnologias, quanto no suporte técnico e capacitação dos professores e das equipes administrativas das escolas (Proinfo, 1999).

De acordo com Universal (1999), telemática é o conjunto das técnicas e dos serviços de comunicação à distância que associam meios informáticos aos sistemas de telecomunicações.

3.2 GERAÇÕES

Os softwares educativos passaram por quatro gerações, cada uma com suas características (Winn, 1993):

a) primeira geração: os softwares educativos desta geração tomam como referência que quase todas as ações dos estudantes são previsíveis, portanto as instruções desenvolvidas levam em consideração um aluno médio, ou seja, não apresentam diferenciação quanto ao grau de conhecimento ou de interesse do aluno;

b) segunda geração: ocorre somente uma troca de enfoque de papéis, onde o que se torna importante é como é apresentada a informação ao estudante. Ocorre um impacto de como os estudantes vão processar estas informações;

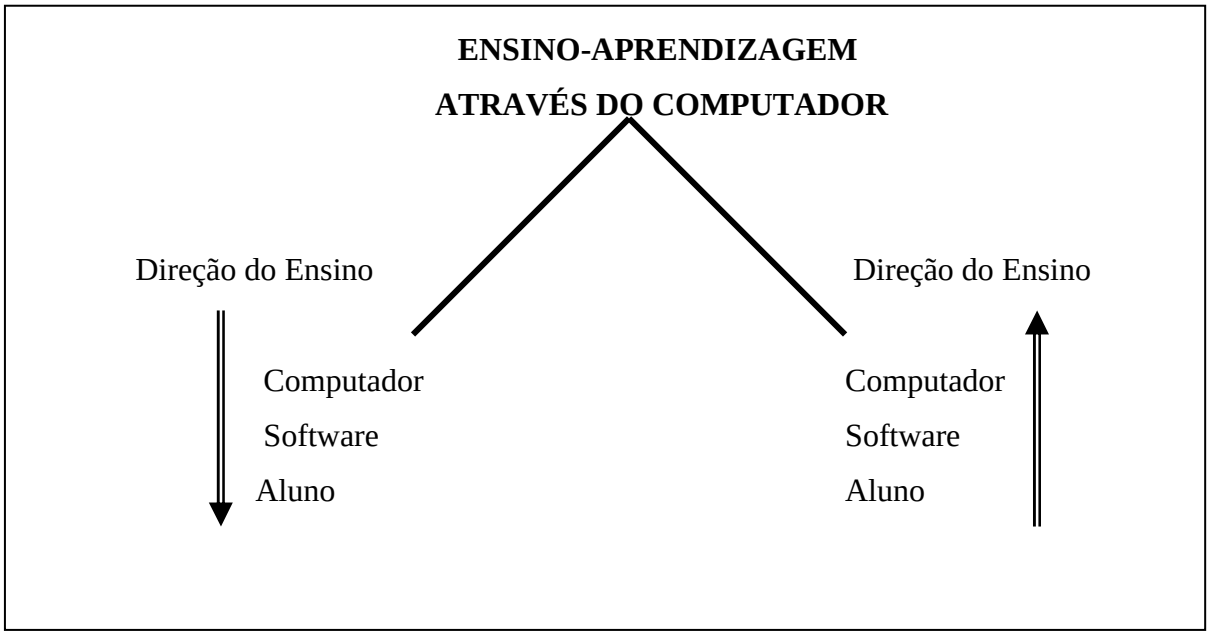
c) terceira geração: admite-se que é através da interação entre o estudante e a informação é que ocorre o aprendizado, ou seja, a forma como a informação é apresentada é da maior importância. Os softwares desta geração também seguem a teoria cognitiva, onde se firma que toda a aprendizagem é resultado de uma interação entre estudante e programa;

d) quarta geração: parte do pressuposto que a aprendizagem se dá através da interação direta entre o aluno e o ambiente de aprendizado. Os softwares característicos desta geração, que está surgindo são os que permitem a imersão em realidade virtual.

3.3 DIFERENTES USOS DO COMPUTADOR NA EDUCAÇÃO

Segundo Valente (1993), na educação o computador tem sido utilizado tanto para ensinar sobre computação, “ensino de computação”, como para ensinar praticamente qualquer assunto, com o “ensino através do computador”. No “ensino de computação’ o computador é usado como objeto de estudo, ou seja, o aluno usa o computador para adquirir conceitos computacionais, como princípios de funcionamento do computador, noções de programação e implicações sociais do computador. O “ensino através do computador” implica que o aluno, através da máquina, possa adquirir conceitos sobre praticamente qualquer domínio. Entretanto, a abordagem pedagógica de como isto acontece é bastante variada, oscilando entre dois grandes pólos, como mostra a figura 3.1.

Figura 3.1 – Ensino aprendizagem através do computador.



Fonte: adaptado de Valente (1993).

Estes pólos são caracterizados pelos mesmos ingredientes: computador (hardware), o software (programa de computador que permite a interação homem-computador) e o aluno.

Porem, o que estabelece a polaridade é a maneira como esses ingredientes são usados. Em um lado o computador, através do software ensina o aluno. Enquanto no outro, o aluno, através do software, “ensina” o computador. Quando o computador ensina o aluno o computador assume o papel de máquina de ensinar e a abordagem educacional é a instrução auxiliada por computador. Exemplos de softwares que implementam esta abordagem são os tutoriais, jogos educacionais e simulação.

No outro pólo, para o aprendiz “ensinar” o computador o software é uma linguagem computacional, tipo, Pascal, Delphi, ou, uma linguagem para criação de banco de dados, ou mesmo um processador de textos, que permite ao aprendiz representar suas idéias segundo esses softwares. Neste caso o computador pode ser visto como uma ferramenta que permite ao aprendiz resolver problemas ou realizar tarefas como, desenhar, escrever, comunicar-se.

3.4 TIPOS DE SOFTWARES

A partir do momento em que o professor disponibilizar os softwares para auxilia-lo nas suas aulas, é importante que ele efetue uma avaliação do software para que possa utiliza-lo da forma mais adequada, conforme as suas necessidades, bem como, em função dos recursos oferecidos pelo próprio programa.

Em um estudo recente Tajra (1998), classifica os softwares existentes em grandes grupos, com as seguintes características:

a) tutoriais: apresentam conceitos, entretanto, possuem uma interatividade muito restrita, os conceitos se limitam ao que a equipe de desenvolvimento previu, o que muitas vezes não coincide com a necessidade do professor nem com o enfoque que é orientado por ele;

b) exercitação: são os softwares que possibilitam uma interatividade por meio de respostas às questões apresentadas. Com esses softwares, os professores podem inicialmente apresentar conceitos a serem trabalhados no ambiente de aula, de acordo com a disciplina ministrada e, por fim, efetuar exercitações sobre tais conceitos. Uma vez os conceitos tendo sido analisados com outros recursos, principalmente, materiais concretos, facilmente os alunos poderão deliciar-se com as aventuras oferecidas pelos softwares de exercitação;

c) investigação: neste grupo, encontra-se as enciclopédias, em que pode-se localizar várias informações a respeito de assuntos diversos. Com o advento da *Internet*, muitos questionam sobre a real necessidade de obtermos os programas de investigação, visto que, por meio da internet, é possível pesquisar a qualquer momento e sobre qualquer assunto. Entretanto, o conteúdo da internet depara-se com uma série de informações desnecessárias, incorretas e, muitas vezes, mal elaboradas, cabendo ao professor efetuar as devidas análises com seus alunos;

d) simulação: nada melhor do que poder visualizar “virtualmente” grandes fenômenos da natureza e, ainda, fazer experimentos em situações bastante adversas ou simulações que de fato poderiam ocorrer na realidade. Esses softwares exigem uma maior habilidade por parte dos professores para que eles possam analisar os

possíveis acontecimentos havidos. Os softwares simuladores são recursos significativos para o aprendizado e atrativos para os alunos e professores;

e) jogos: são os softwares de entretenimento, a sua maior indicação são o lazer e a diversão. Com certeza, os jogos apresentam grande interatividade e recursos de programação muito sofisticados. Os jogos sofrem grande preconceito na área educacional, pois é comum ouvirmos professores informando aos pais que os alunos usam o ambiente de informática para aprender, com propósitos apenas educacionais, mas os jogos também são grandes ferramentas de que os professores dispõem para ministrar aulas mais divertidas e animadas;

f) abertos: são os de livre produções, o que será elaborado dependerá muito da criatividade do usuário. Oferecem várias ferramentas, as quais podem ser relacionadas conforme o objetivo a ser atingido. Dentre eles pode-se citar: os editores de textos, os bancos de dados, as planilhas eletrônicas, os programas gráficos, softwares de autoria, softwares de apresentações e os de programações:
   - editores de textos: são softwares que apresentam vários recursos de elaboração de textos, tornando mais fácil e rica a produção de trabalhos, visto que por meio deles é possível incluir diversos tipos de fontes, estilos, bordas, figuras, margens, parágrafos. Com ele pode-se elaborar atividades de criações de relatórios, cartas, poesias, músicas, entrevistas e, inclusive, produzir jornais;
   - banco de dados: possibilitam o arquivamento de informações que poderão posteriormente ser relacionadas para diversos tipos de análises e ordenações conforme o interesse do usuário. Com o banco de dados, o professor pode, juntamente com os alunos, efetuar uma coletânea de informações sobre países, tais como: nome do país, extensão territorial, população, etnias, religiões, etc, e em seguida, efetuar uma série de comparações entre os países para uma posterior análise;
   - planilhas eletrônicas: possibilitam a realização de cálculos, de uma forma rápida, a partir de dados informados e, posteriormente, a elaboração de gráficos que facilitam a visualização das informações;
   - softwares gráficos: são aqueles voltados para a elaboração de desenhos e produções artísticas. Por meios destes programas é possível criar desenhos sobre

os mais variados temas, o aluno pode usar *clip-arts* a partir do próprio programa ou adquiri-los isoladamente em revistas ou lojas de distribuição e revenda de softwares e ainda criar os seus próprios desenhos de acordo com sua imaginação e criatividade. Pode-se ainda incluir neste grupo os softwares de captura de imagem, onde através do *scanner* extrai imagens de fotos, revistas, jornais, podendo melhorar a imagem capturada;

g) softwares de autoria: com certeza este tipo de software é um dos mais gratificantes para professores e alunos. Pode-se através destes softwares desenvolver aulas utilizando recursos multimídia e sem grandes complicações. Estes softwares funcionam como um aglutinador de produções elaboradas em outros softwares. A grande vantagem dos softwares de autorias, além da facilidade de manuseio que eles possuem, o professor poderá montar rapidamente uma aula dentro do roteiro e enfoque em que ele aborda a disciplina a qual leciona;

h) softwares de apresentação: estes programas são muito utilizados para elaborar apresentações de palestras e aulas. Possuem recursos de visualização de telas, bem como, permitem produções de slides e transparências. Estes softwares, também, são muito bem aceitos pelos alunos, pois estes poderão elaborar seus trabalhos em forma de apresentação diferentemente de textos impressos;

i) softwares de programação: são os que permitem a criação de outros programas, ou seja, programas executáveis. Estes programas são ótimos para estimular raciocínio lógico, entretanto, as suas produções geralmente são mais demoradas que as dos softwares anteriormente citados e requerem um bom preparo do professor quanto ao domínio dos comandos dos softwares de programação, bem como uma visão bastante sistemática das rotinas de programação.

3.5 SOFTWARES EDUCACIONAIS

De acordo com Campos (1991), o software educacional vem entrando no mercado mundial de forma muito acelerada. Inúmeros países como Inglaterra, França e EUA, entre outros, desenvolveram projetos de uso do microcomputador em educação e, conseqüentemente necessitaram desenvolver produtos de software específicos para suas

necessidades. O mesmo tem ocorrido no Brasil, aonde diversos projetos de pesquisa vêm sendo desenvolvidos não só relacionados ao uso do microcomputador em sala de aula como, também, ao desenvolvimento de software para os mais diversos conteúdos programáticos. Alguns grupos de pesquisa utilizam o termo software educacional, ou software educativo, outros o termo *courseware*, outros, ainda, o termo programas educativos por computador. Todos estes termos possuem um mesmo significado: material educacional para microcomputadores.

Didaticamente deve-ser ressaltar o estímulo pelo computador da relação entre a criatividade e a inteligência. O computador é um instrumento que obriga à criatividade e dela elaborar os *input* de forma lógica: fornece instrumentos lógicos que exigem um desempenho criativo por parte da criança. Dar aula não significa mais trabalhar sobre símbolos, mas construí-los a partir de experiências (Lollini, 1991).

3.5.1 CARACTERÍSTICAS DE UM SOFTWARE EDUCACIONAL

Um software multimídia apresenta determinadas características próprias. Estas são classificadas em três dimensões conforme descritas abaixo (Winn, 1997).

a) autonomia: neste caso o ambiente funciona por si só, ou seja, não necessita de nenhuma ação do usuário. Exemplos característicos de baixa autonomia são os tutoriais e programas de exercício e prática que ficam parados no aguardo de alguma ação do usuário. Como sistemas autônomos, pode ser incluídas as simulações em tempo real e alguns jogos. Nestes casos, o ambiente segue o seu desenvolvimento próprio sem esperar por uma atitude do usuário;

b) presença: esta característica está relacionada em fazer com que o usuário se sinta como se estivesse realmente no lugar representado. Com ambientes baseados em computador, o senso de presença aumenta à medida em que a interface se torna mais intuitiva, mais transparente, tornando-se mais próximo do mundo real;

c) interação: está intimamente relacionada com a forma como o usuário interage com o ambiente, ou seja, o aprendizado ocorre de acordo com as observações e ações que o usuário toma em relação a determinados assuntos. A interação depende única e exclusivamente das ações e reações do usuário.

3.6 PROFESSOR E A INFORMÁTICA

Com o advento do computador surge uma dúvida: qual o papel dos professores? O seu relacionamento com os alunos poderá sofrer mudanças radicais, e principalmente, para pior? É este medo que faz com que ocorra uma grande resistência por parte dos professores em utilizar a informática a serviço da educação.

De acordo com Lollini (1991), o computador ameaça o difícil equilíbrio existente nos relacionamentos interpessoais no interior da escola. Como no passado, também agora tais inquietações disfarçam um antigo problema do homem e seus sistemas educativos: o problema da transformação e a resistência a ela oferecida.

As mudanças geram ansiedade na instituição e nos seus membros, porque são tidas como causa de incerteza e põem em discussão rotinas já testadas pela experiência. A escola e seus profissionais opõem sempre algum tipo de resistência à mudança. Não se trata de assumir isso como um dado de fato inalterável nem como motivo para guerras santas. A resistência às mudanças deve ser enfrentada com inteligência:

a) fazendo com que o computador não seja resultado de uma escolha imposta do alto ou por alguém tecnicamente mais bem preparado, mais uma escolha ponderada, decidida em conjunto;
b) evitando considerar o computador como máquina diabólica que tudo faz, nunca erra. Ele deve ter um lugar na aula e seu uso deve ser estimulado nos limites aconselháveis ou imprescindíveis. Nunca além;
c) evitando considera-lo como substituto do professor, mas também evitando continuar a assumir tarefas que podem ser melhor desempenhadas por uma máquina.

Uma outra definição bastante parecida é de Tajra (1998), onde afirma que no início da introdução dos recursos tecnológicos de comunicação na área educacional, houve uma tendência a imaginar que os instrumentos iriam solucionar os problemas educacionais, podendo chegar, inclusive a substituir os próprios professores. Com o passar do tempo, não foi isso que se percebeu, mas a possibilidade de utilizar esses instrumentos para sistematizar os processos e a organização educacional.

Lucena (2000), destaca que quanto aos professores, eles devem ser preparados para trabalhar como facilitadores, tutores e até mesmo provocadores de participação. O professor não tem mais a missão de transmitir conhecimento e, sim, de orientar o aluno e ajuda-lo na busca do conhecimento. Essas são as novas tarefas do professor: estipular metas, planejar e estar atento para que os recursos estejam disponíveis. Com esse tipo de tecnologia pode-se ter muito mais interação e com maior qualidade. Pode-se propiciar mais fontes de ajuda para os alunos e um ambiente de discussões mais participativo. Esta participação está embutida na natureza das ferramentas de comunicação disponíveis, e em uma postura de motivar o aluno, uma vez que nesse ambiente ele precisa buscar a informação e tratar do conhecimento.

Para Mielke (1991), não se pode desconsiderar o papel do professor como agente motivador no ensino. Nesta tarefa, o computador não será capaz de substituir o professor, já que depende bastante de um apoio psicológico que ele exerce sobre seus alunos, motivando-os e adequando as diversas ferramentas disponíveis, para que se adequem ao perfil e necessidades de cada aluno. Dessa forma, cabe ao computador o papel de ferramenta auxiliar no processo de ensino-aprendizagem, conhecimentos desejados, e treinar repetitivamente o aluno até que ele apresente o desempenho pré-estabelecido, deixando para o professor definir “quando” e “como” usar a informática.

3.7 PONTOS POSITIVOS PROPORCIONADOS PELOS AMBIENTES DE INFORMÁTICA EDUCATIVA

Segundo Tajra (1998), o que os ambientes de informática educativa proporcionam positivamente são:

a) os alunos ganham autonomia nos trabalhos, podendo desenvolver boa parte das atividades sozinhos, dentro de características pessoais. Atendendo de forma mais nítida ao aprendizado individualizado, existe de fato um respeito pelo desenvolvimento individual;
b) em função da gama de ferramentas disponíveis nos softwares, os alunos, além de ficarem mais motivados, também se tornam mais criativos. A curiosidade é outro elemento que é bastante aguçado com a informática, visto que é ilimitado o que se pode aprender e pesquisar com os softwares e *sites* disponíveis;

c) é muito comum os alunos se auto-ajudando. Os ambientes tornam-se mais dinâmicos e ativos, os alunos que sobressaem pelo uso da tecnologia costumam ajudar aqueles que estão com dificuldades. Alunos com dificuldades de concentração tornam-se mais concentrados. Portanto estes ambientes favorecem inclusive, uma socialização que, às vezes, não se consegue nos ambientes tradicionais;
d) as aulas corporativas perdem espaços para os trabalhos corporativos;
e) estímulo a uma forma de comunicação voltada para a realidade atual de globalização;
f) além de a escola poder direcionar as fontes de pesquisas para os recursos já existentes, tais como livros, enciclopédias, revistas, jornais e vídeos, ela pode optar por mais uma fonte de aprendizagem : o computador;
g) com certeza, com certeza, uma grande contribuição da informática é o auxílio para o desenvolvimento das habilidades de comunicação e de estrutura lógica de pensamento.

O ambiente proposto será um ambiente do tipo autoria, onde os usuários (professor e aluno), poderão criar os seus próprios materiais didáticos. Fundamenta-se, pois, nas bases humanistas de Paulo Freire, que afirma que o importante não é apenas transmitir conteúdos específicos, mas despertar uma nova forma de relação com a experiência vivida. O humanismo de Paulo Freire acredita que o saber do professor não deve apenas ser depositado no aluno, mas que deve existir o relacionamento educador-educando, onde juntos se posicionam como sujeitos do ato do conhecimento.E o ambiente de autoria possui esta capacidade de interação entre professor e aluno, de forma que o próprio professor irá criar o seu material, de acordo com o seu conteúdo, e com as necessidades de seus alunos. A partir deste material, os alunos poderão fazer alterações, acrescentando e/ou alterando o conteúdo repassado pelo professor.

A união dos fundamentos humanistas de Paulo Freire e dos fundamentos de criação de um ambiente de autoria são a base para o desenvolvimento do “*Ambiente para Criação de Material Didático Multimídia*” proposto neste trabalho de conclusão de curso.

4 METODOLOGIA

Para este trabalho foram usadas as seguintes tecnologias: metodologia orientada a objetos, ferramenta de desenvolvimento CASE e ambiente de desenvolvimento Macromedia Director, através da linguagem de programação Lingo.

4.1 METODOLOGIA DE ORIENTAÇÃO A OBJETOS

Segundo Winn (1993), a orientação a objeto representa um avanço a partir dos métodos tradicionais de construção de softwares. Os métodos tradicionais aplicam procedimentos ativos a dados passivos. Os métodos objeto-orientado encapsulam procedimentos e dados. A orientação a objeto abrange o maior número dos componentes de softwares, incluindo linguagens, banco de dados e interfaces. Os resultados são softwares mais fáceis de manter e expandir, e aplicações flexíveis e mais fáceis de usar. A análise orientada é importante atualmente devido ao aumento da complexidade dos softwares e a necessidade de melhores processos de construção.

Os benefícios primários oferecidos aos criadores e usuários com a implementação do paradigma orientado ao objeto referem-se à habilidade de resolver dois dos principais problemas de engenharia de software: gerenciamento de complexidade e aumento de produtividade no processo de desenvolvimento de software. A programação orientada ao objeto trata estes temas direcionando as seguintes estratégias de desenvolvimento de software(Winblad, 1993):

a) escrevendo código de programação reutilizáveis;
b) escrevendo códigos de fácil manutenção;
c) refinando módulos de códigos existentes;
d) compartilhando códigos de programação.

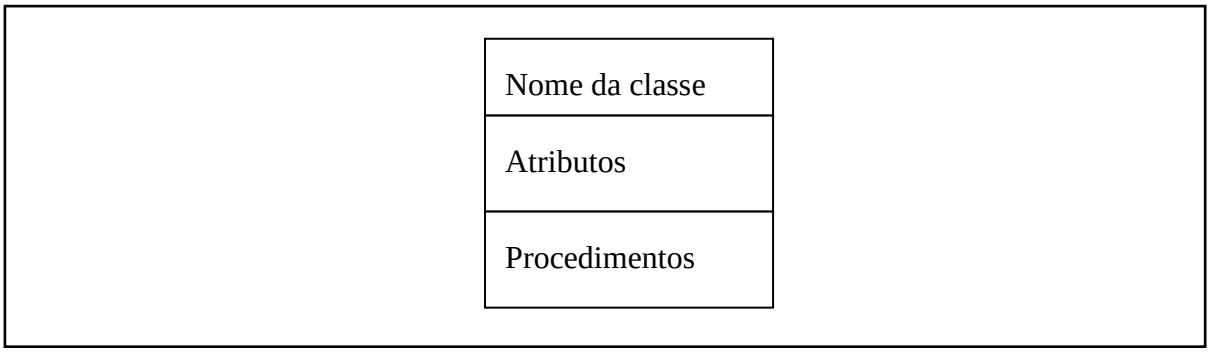
A complexidade diminui e a produtividade aumenta quando a codificação de alta qualidade está disponível para a reutilização. Os mecanismos orientados ao objeto, principalmente a hereditariedade, estimulam a reutilização de maneira positiva. Em vez de copiar e modificar os módulos, os programadores podem utilizar bibliotecas de classe que contêm códigos refinados e testados. Uma codificação de programação orientada ao objeto geralmente pode ser suficiente para reutilização sem codificação. Como cada objeto sabe

responder às solicitações com o seu próprio meio adequado, as mesmas instruções podem operar em diferentes objetos.

Para atender a Orientação a Objetos é necessário antes entender alguns conceitos básicos dessa metodologia, descritos sucintamente a seguir:

a) objeto: Martin (1996), define objeto como, "*qualquer coisa, real ou abstrata, a respeito da qual armazena-se dados e os métodos que os manipulam*". Para Winblad (1993), um programa tradicional consiste em *procedures* e dados. Um programa orientado ao objeto consiste somente em objetos que contêm *procedures* e dados. Em outras palavras, objetos são módulos que contem dados e instruções para operar sobre estes dados. Dentro dos objetos, residem os dados das linguagens tradicionais, como números, matrizes, *strings*, registros, bem como funções, instruções ou sub-rotinas que os operam. Os objetos então são entidades que tem atributos específicos (dados) e maneiras de comportamento (procedures);
b) classe: De acordo com Rumbaugh (1994), uma classe de objetos descreve um grupo de objetos com propriedades semelhantes (atributos), o mesmo comportamento (operações), os mesmos relacionamentos com outros objetos e a mesma semântica. Generalização é o relacionamento entre uma classe e uma ou mais versões refinadas dela. A classe que estiver em processo de refinamento é chamada de superclasse e cada versão refinada é denominada subclasse. Martin (1996), define classe como: "*... uma classe é uma implementação de um tipo de objeto. Ela especifica uma estrutura de dados e os métodos operacionais permissíveis que se aplicam a cada um dos seus objetos*", como mostra a figura 4.1;

Figura 4.1:Estrutura de uma classe

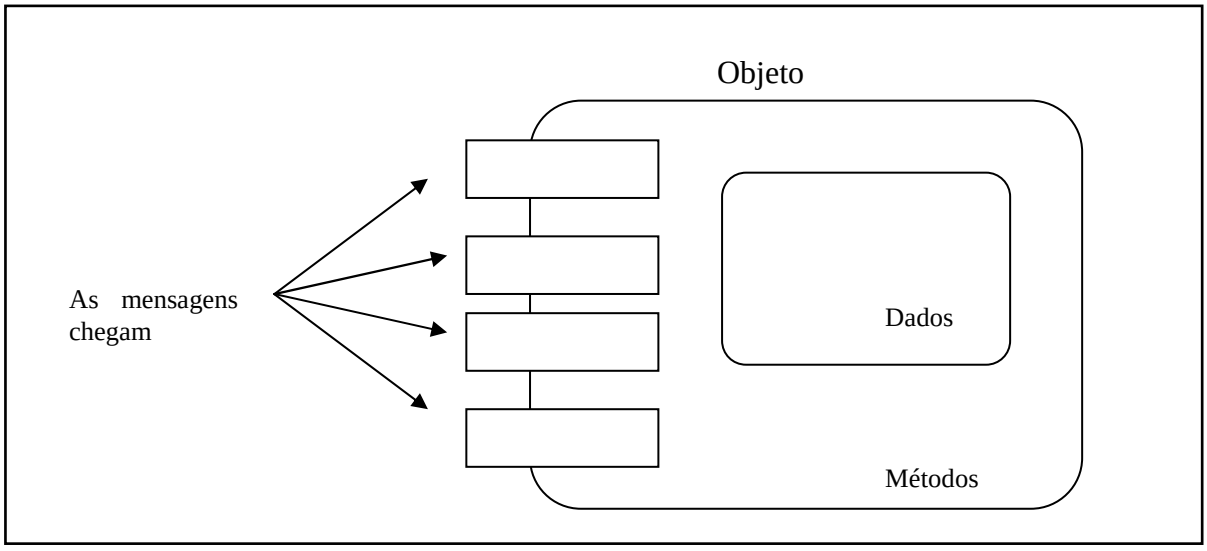


Fonte: adaptado de Martin (1996).

c) métodos: especificam a maneira pela qual os dados de um objeto são manipulados (Martin, 1996). Para Winblad (1993), as procedures chamadas “métodos” residem em objetos e determinam como o objeto atuará quando receber uma mensagem;

d) mensagem: para que um objeto realize alguma coisa, envia-se uma solicitação, que faz com que uma operação seja invocada. A operação executa o método apropriado e opcionalmente, retorna uma resposta. A mensagem que constitui a solicitação contém o nome do objeto, o nome da operação e, às vezes um grupo de parâmetros (Martin, 1996). A figura 4.2 ilustra a anatomia de um objeto, e a figura 4.3 ilustra objetos se comunicando através de solicitações, onde cada solicitação é uma mensagem especificando que uma operação indicada seja executada;

Figura 4.2 : Anatomia de um objeto

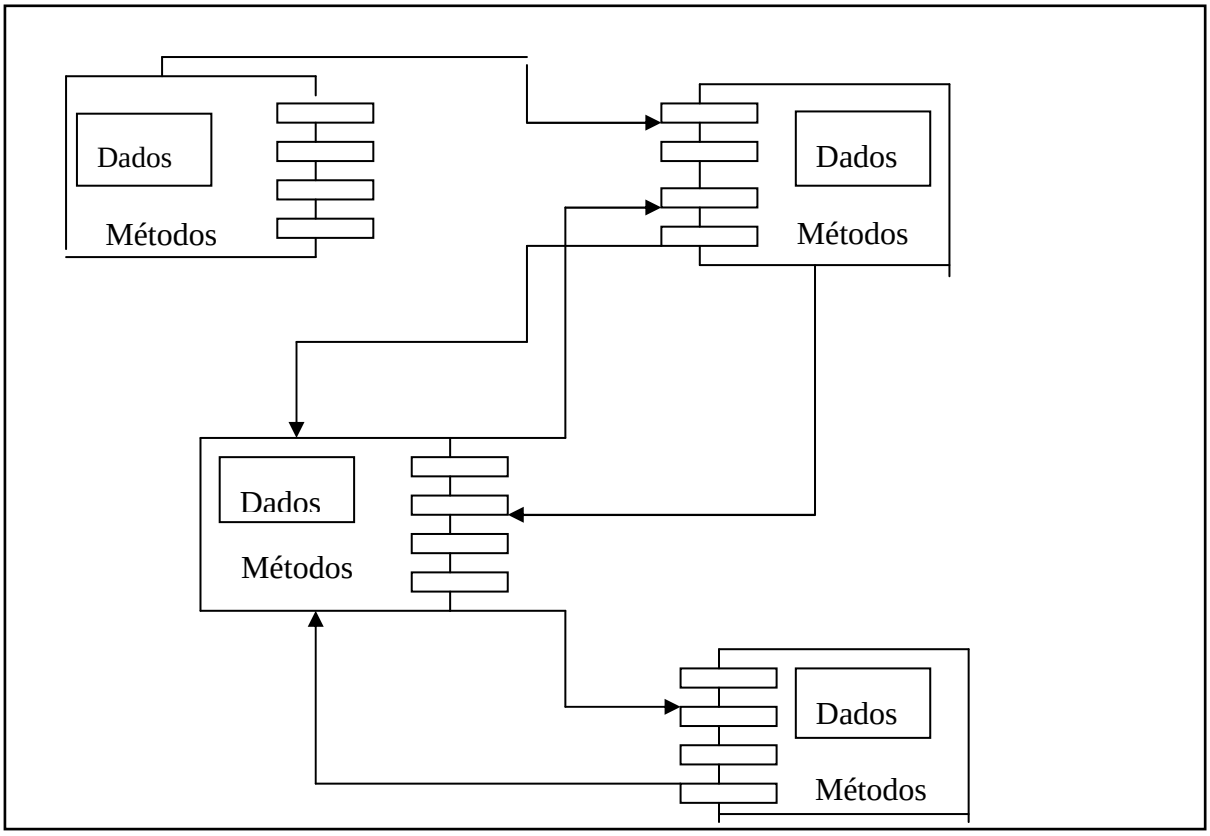


Fonte: Adaptado de Winblad (1993).

e) encapsulamento: O ato de empacotar ao mesmo tempo dados e métodos é denominado encapsulamento. O objeto esconde seus dados de outros objetos e permite que os dados sejam acessados por intermédio de seus próprios métodos. O encapsulamento protege os dados de um objeto contra a adulteração (Martin, 1996). De acordo com Winblad (1993), encapsulação é um termo formal que descreve a junção de métodos e dados dentro de um objeto de maneira que o acesso aos dados seja permitido somente por meio dos próprios métodos do objeto. Nenhuma outra parte do programa orientado ao objeto pode operar diretamente em um dado do objeto;

f) herança: é o compartilhamento de atributos e comportamentos entre objetos com base em um relacionamento hierárquico. Cada subclasse (classe que herda propriedades) incorpora todas as propriedades de sua superclasse (classe que define propriedades) e acrescenta suas próprias e exclusivas características (Martin, 1996), como mostra a figura 4.4;

Figura 4.3: Objetos se comunicando com solicitações



Fonte: adaptado de Winblad (1993).

g) relacionamentos: os relacionamentos entre os objetos são demonstrados através dos seguintes símbolos (Martin, 1996):

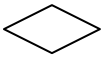
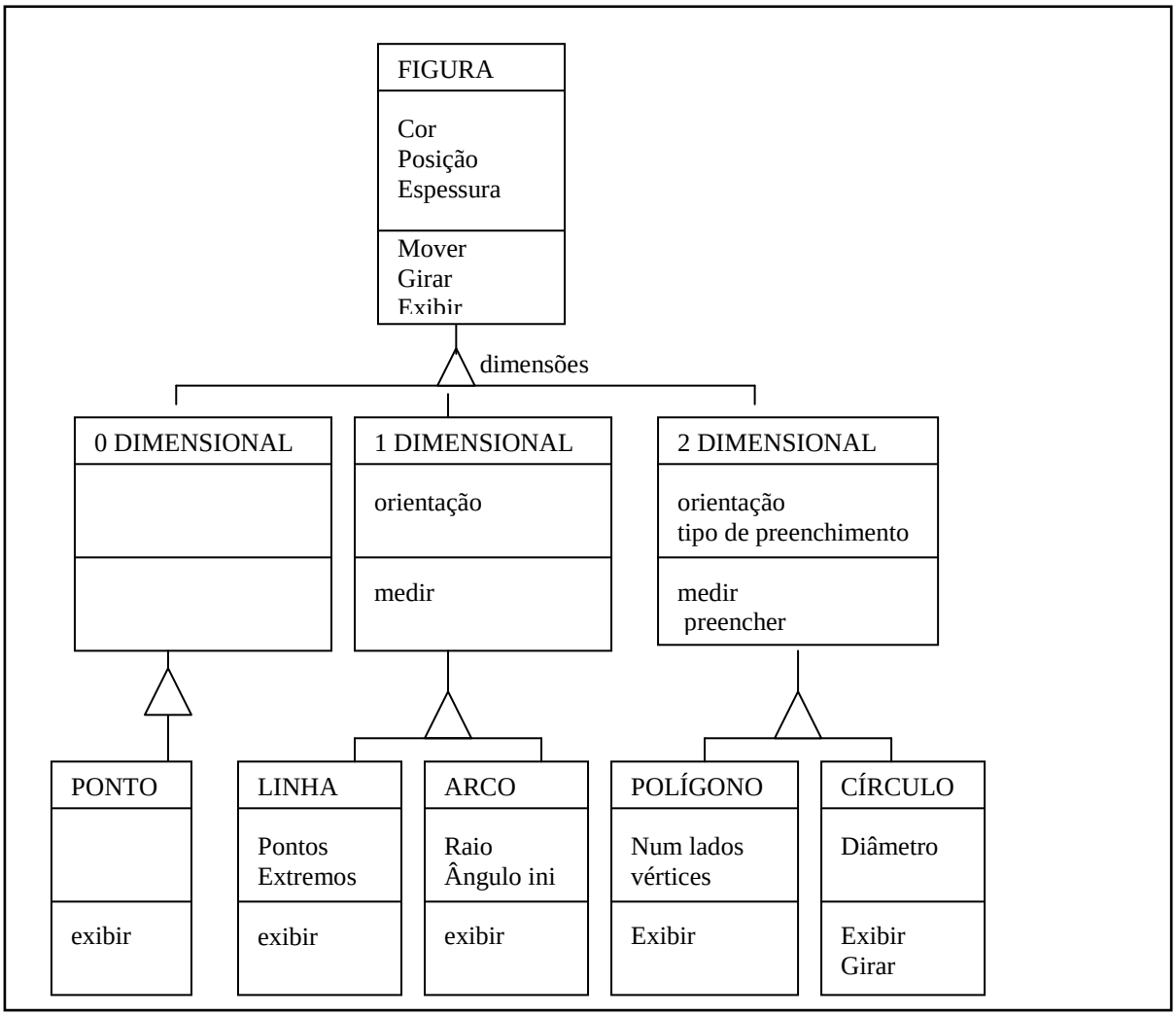
	Dependência – o objeto relacionado a ele depende dele para existir.
	Herança – os objetos relacionados ao objeto acima, herdam as características dele.
	Cardinalidade – o objeto se relaciona uma ou várias vezes (1, N).
	Cardinalidade – o objeto pode se relacionar várias vezes (0, N).
	Cardinalidade – o objeto se relaciona uma vez (1,1).

Figura 4.4 :Exemplo de Herança – Figuras Gráficas



Fonte: adaptado de Rumbaugh (1994)

O ambiente utilizado para o desenvolvimento do presente trabalho é o Macromedia Director 8.0, este ambiente permite a programação orientada a objetos através da linguagem Lingo. Apesar de permitir a orientação a objetos, a terminologia utilizada pelo Director é um pouco diferente daquela encontrada na literatura especializada. Na Figura 4.5, são apresentados os termos utilizados nos livros sobre orientação a objetos e o termo correspondente utilizado pelo Director.

Figura 4.5 - Termos utilizados em Orientação a Objetos e equivalente em Lingo.

Termos OOP	**Lingo**
Super classe	Ancestor
Classe	Parent Script
Variável de instância	Propriedade
Objeto	Child Object
Método	Método

Fonte: adaptado de Rosenzweig (2000).

4.2 HIPERMÍDIA

Uma das tecnologias de interesse pesquisada com finalidades educacionais é a dos sistemas *hipermídia.* Estes sistemas têm surgido como uma nova classe para o gerenciamento de informações, pois permitem criar, anotar, unir, e compartilhar informações a partir de uma variedade de meios (como texto, gráfico, som, vídeo e animação), proporcionando o acesso às informações de uma forma não seqüencial e utilizando métodos inteiramente novos, ao contrário dos sistemas de informações tradicionais que são seqüenciais por natureza.

O campo da hipermídia se encontra em rápido crescimento. Novas áreas de aplicação, que necessitam da flexibilidade da hipermídia unida a uma rica variedade de tipos de dados multimídia, estão emergindo. Estas aplicações abrangem desde sofisticados sistemas de informação suportados por bases de dados orientadas a objetos e relacionais, até sistemas de informações geográficas passando por ambientes de suporte de decisões e de engenharia de software (Bieber, 1995).

De acordo com Parkes (1994), a hipermídia que foi inicialmente concebida como uma ferramenta para a recuperação de informações, atualmente tem sido considerada como uma ferramenta para a aprendizagem. Esta tecnologia é considerada bastante adequada para aplicações educacionais, principalmente por causa de sua flexibilidade e grande capacidade de exploração de informações relevantes.

Para Smith (1988), hipermídia é *"uma abordagem para o gerenciamento de informação no qual os dados são armazenados em uma rede de nodos conectados por ligações. Os nodos podem conter texto, gráficos, som, vídeo, assim como código fonte ou outras formas de dados"*.

Segundo Reynolds (1996), hipermídia é um banco de dados que espera fornecer um fim amigável para o usuário, para mover as informações através de um caminho relacional, com base em palavras, objetos ou relacionamentos. Em um sistema hipermídia, ocorre a manipulação de um conjunto de informações que pertence a vários tipos de mídia (texto, som, imagem), sendo que essas informações podem ser lidas de uma forma não-linear.

O ambiente hipermídia oferece novas possibilidades de acesso às grandes e complexas fontes de informações. Um documento linear pode ser lido somente na ordem em que foi composto. A vantagem essencial do documento não linear é a capacidade de organizar objetos de diversas maneiras, dependendo das diferentes visões e demandas. O modelo hipermídia incentiva o autor a criar referências e modularizar suas idéias, embora ele seja obrigado a tomar decisões difíceis sobre qual a melhor maneira de particionar adequadamente as informações (Chaiben, 2000).

Conklin (1987), aponta uma lista de vantagens em um modelo hipermídia:

a) facilidade de seguir as ligações. A interface com o usuário é uma das características marcantes em sistemas hipermídia. O ambiente gráfico proporciona facilidades de navegação em grandes espaços de informação;
b) facilidade de criar novas referências. Os usuários podem simplesmente fazer comentários ou anotações em um documento, enquanto as demais referências continuam inalteradas;
c) estruturação da informação. Tanto as organizações hierárquicas como não hierárquicas, podem ser aplicadas sobre informações não estruturadas. Até mesmo hierarquias múltiplas podem organizar o mesmo material;
d) customização de documentos. Os segmentos podem ser estruturados de várias maneiras permitindo que o mesmo documento sirva para múltiplas funções;
e) modularidade Uma vez que partes do mesmo documento podem ser referenciadas de vários lugares, as idéias podem ser expressas com pouca sobreposição ou duplicação;

f) consistência da informação. As referências estão embutidas no documento e mesmo que este seja movido pelo autor, as informações das ligações continuam dando acesso direto àquelas referências;
g) colaboração. Vários autores podem cooperar na criação de um mesmo documento ou simplesmente adicionar e compartilhar comentários.

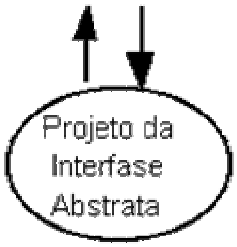
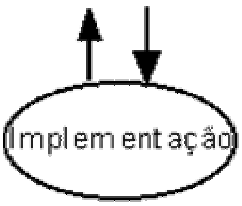
4.3 MÉTODO ORIENTADO A OBJETOS PARA DESIGN HIPERMÍDIA

Com o crescimento das aplicações que utilizam multimídia como, *web sites* e Cd - Rom's interativos, a necessidade de especificação de uma forma ou método para melhor planejar, modelar e construir estas aplicações era grande. Foi especificada então a OOHDM (*Object Oriented Hipermedia Design Method*), que traduzindo significa "Método orientado a objetos para design hipermídia", que procurou facilitar as tarefas de criação, manutenção e melhorias em projetos que possuam algum tipo de recurso multimídia, como vídeos ou fotos.

Os principais pontos a serem seguidos por esta metodologia compreendem: *design* conceitual, *design* da navegação, *design* da interface abstrata e a implementação.

Segundo Rossi (1996), OOHDM considera o processo de desenvolvimento da aplicação hipermídia como um processo de quatro atividades, desempenhadas em uma mistura de estilos iterativos e incrementais de desenvolvimento; em cada etapa um modelo é construído ou enriquecido, conforme figura 4.6.

Figura 4.6 – Atividades da OOHMD

Atividades	Produtos	Mecanismos	Interesses do Projeto
	Classes, sub-sistemas, relacionamentos, perspectivas de atributos	Classificação, composição,generalização e especialização	Modelagem da semântica do domínio de aplicação
	Nós, elos, estruturas de acesso, contextos de navegação, transformações navegacionais	Mapeamento entre objetos conceituais e de navegação.Padrões de navegação para a descrição da estrutura geral da aplicação	Leva em conta o perfil do usuário e a tarefa; ênfase em aspetos cognitivos e arquiteturais
	Objetos de interface abstrata, reações a eventos externos, transformações de interface	Mapeamento entre objetos de navegação e objetos de interface	Modelagem de objetos perceptíveis,implementa metáforas escolhidas. Descrição de interface para objetos navegacionais
	Aplicação em execução	Aqueles fornecidos pelo ambiente alvo	Desempenho, completitude

Fonte: adaptado de Rossi (1996).

Na figura 4.6 resume-se as quatro atividades: modelagem de domínio ou conceitual projeto navegacional, projeto da interface abstrata e implementação, assim como os produtos gerados em cada uma e os mecanismos de abstração utilizados ao longo do processo. As setas de avanço demonstram a evolução natural do empreendimento de projeto, enquanto as setas

de retorno expressam não apenas a possibilidade de *feedback loops* na forma mencionada em (Izakowitz, 1995), mas também a existência de um modelo simples, porém potente, de rastreamento que permite mapear as modificações em um modelo de projeto para partes de um outro modelo (rastreamento para trás).

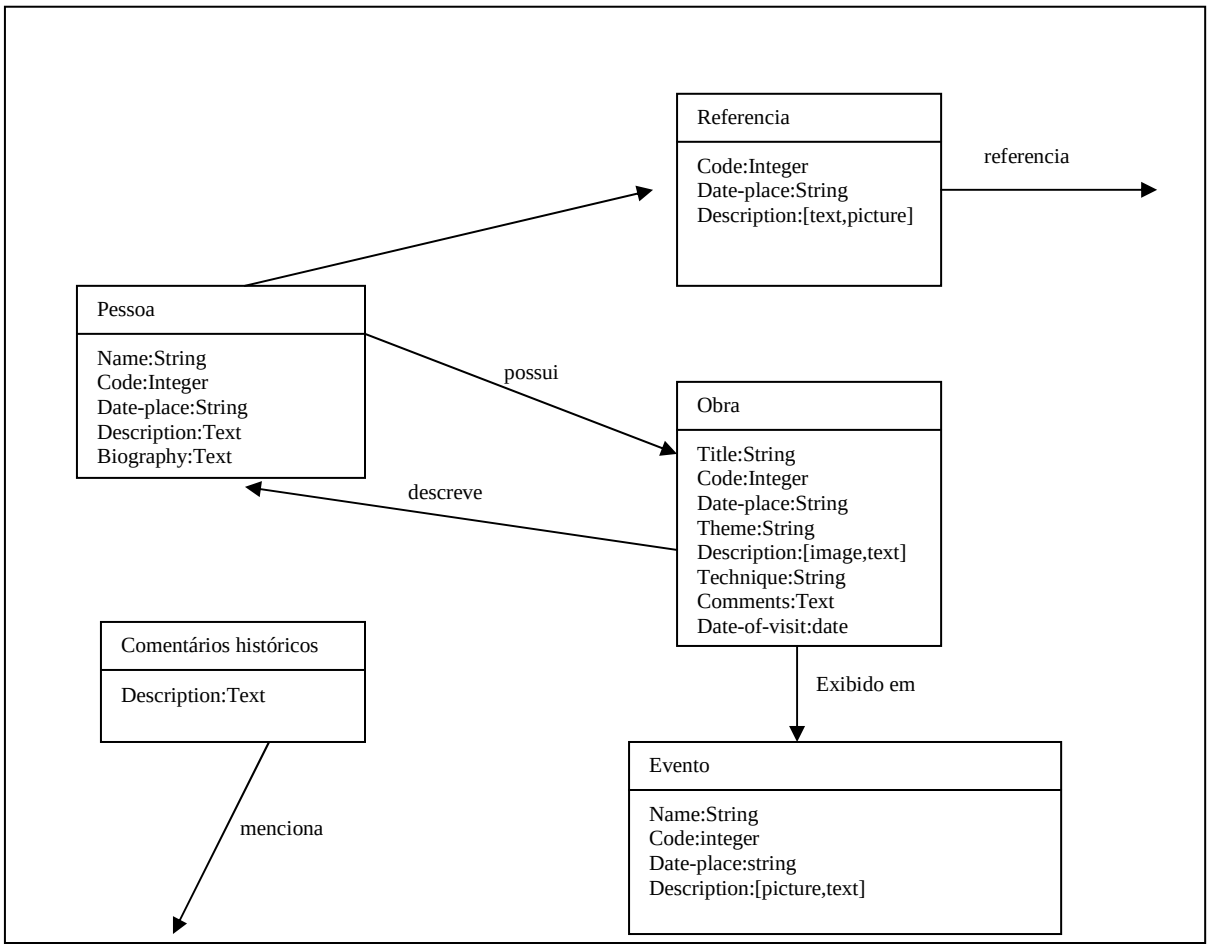
4.4 DESIGN CONCEITUAL

O design conceitual se constitui basicamente por classes, relações e subsistemas. O modelo conceitual é concebido utilizando as técnicas de modelagem já utilizadas para construção de aplicações orientadas a objetos, com a adição de outros elementos, como perspectivas de atributos. As classes conceituais podem ser montadas utilizando as hierarquias de agregação, generalização/especificação; cada classe pode se relacionar com um subsistema (abstração de um sistema conceitual completo), desde que o mesmo possua um ou mais pontos de entrada.A principal preocupação neste ponto é de observar e representar a semântica do domínio da aplicação, sem muita preocupação com usuários ou tarefas (Schwabe, 1996).

Na figura 4.7 pode-se ver um exemplo do esquema conceitual que consiste de um conjunto de objetos e classes unidos entre si por relacionamentos, como os objetos são instâncias de classes, um relacionamento entre as classes faz com que o relacionamento entre os objetos seja abstraído.

Durante o projeto de navegação, as classes serão mapeadas para nós de hipermídia, enquanto os relacionamentos serão mapeados para elos hipermídia.

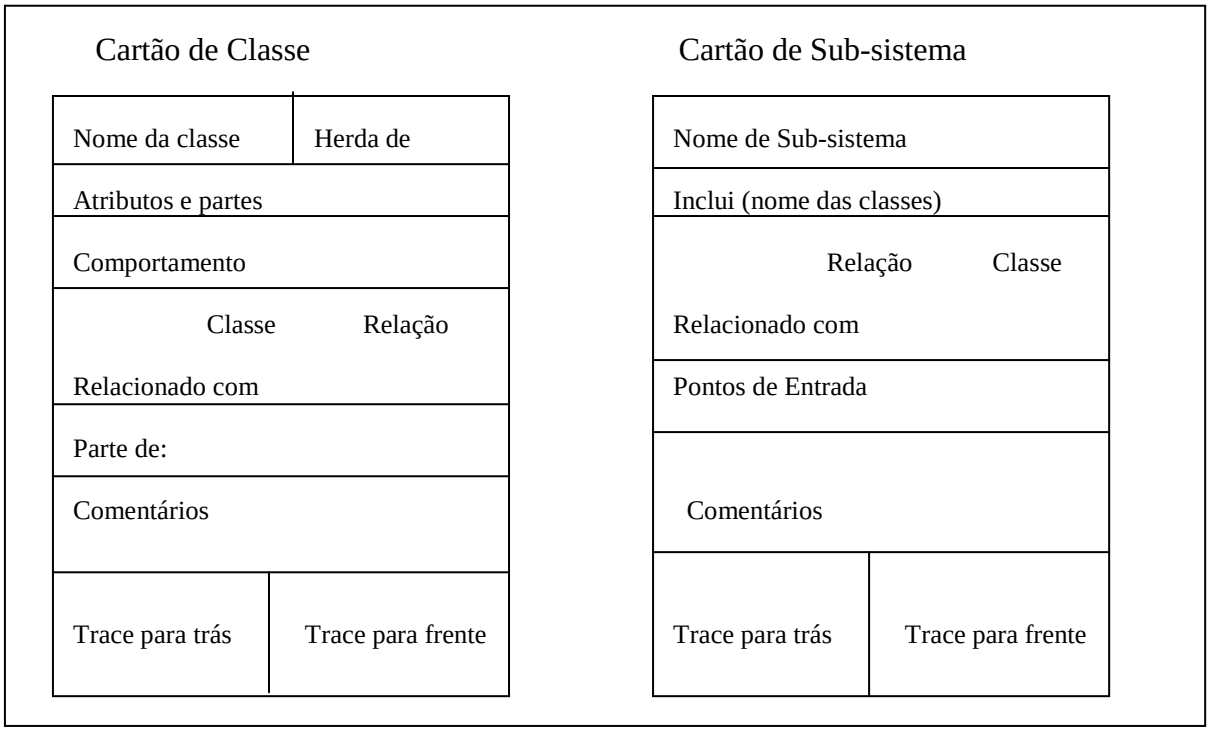
Figura 4.7 – Parte do esquema conceitual



Fonte: adaptado de Rossi (1996).

São utilizados cartões de identificação pra facilitar na documentação do sistema, conforme figura 4.8. Os cartões incluem informações sobre os artefatos documentados e rastreiam informações. O rastreamento "para trás" permite responder a perguntas como: de onde veio este artefato? (uma entidade do mundo real, ou um relacionamento, por exemplo), enquanto o rastreamento "para frente" é importante para a análise do impacto de mudanças no modelo durante a evolução do sistema (em que classes a mudança influi?). As informações de rastreamento contêm um conjunto de itens indicando o tipo de relacionamento de rastreamento e o seu alvo.

Figura 4.8 –Cartões de identificação



Fonte: adaptado de Rossi (1996).

4.5 DESIGN DE NAVEGAÇÃO

Os aplicativos de hipermídia são projetados para efetuar navegação através de um espaço de informações. Por isto, o projeto da estrutura de navegação de tais aplicativos é a etapa crucial no empreendimento de desenvolvimento.

Segundo Schwabe (1995), no método OOHDM a aplicação é vista como uma visão navegacional sobre o domínio conceitual. Isto demonstra que uma das principais características desta metodologia que difere das outras é a noção de navegação. Neste ponto o desenvolvedor deve levar em conta os tipos de usuários e as tarefas que os mesmos irão realizar no uso da aplicação.

Qualquer aplicação hipermídia bem projetada deve levar em conta o modo como o usuário explora o espaço hipermídia. Deve-se evitar a apresentação de informações redundantes e ajudar, de forma consistente e controlada, o usuário a escolher a maneira como navegará. Um contexto navegacional é constituído por um conjunto de nós, elos e outros contextos navegacionais. Inclui, também, um caminho pré-definido entre seus elementos.

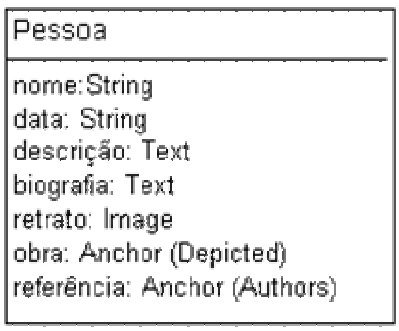
Para expressar diferentes visões (ou aplicações), modelos de navegação diferentes podem ser construídos sobre o mesmo esquema conceitual do domínio. A estrutura de navegação de uma aplicação hipermídia é definida por um esquema especificando classes de navegação que refletem a visão escolhida sobre o domínio da aplicação. Existe um conjunto de tipos pré-definidos de classes de navegação: os nodos, *link's* e estruturas de acesso, que são organizados em contextos de navegação, ou seja, um conjunto de nodos, *link's*, classes de contexto e outros contextos de navegação. A semântica de nodos e *link's* são comuns em aplicações hipermídia, enquanto as estruturas de acesso podem representar uma maneira alternativa de acessar nodos como índices e *tours* (Schwabe, 1995).

4.5.1 NODOS

Nodos são as estruturas básicas utilizadas nas aplicações hipermídia como armazenadores de informações. São visões orientadas-a-objetos de classes conceituais definidas durante o design conceitual, usando uma linguagem de consulta, permitindo assim que o nodo seja definido por uma combinação de atributos de diferentes classes relacionadas no esquema conceitual. Eles contem atributos unitários e *anchors* (âncoras), e podem ser atômicos ou compostos.

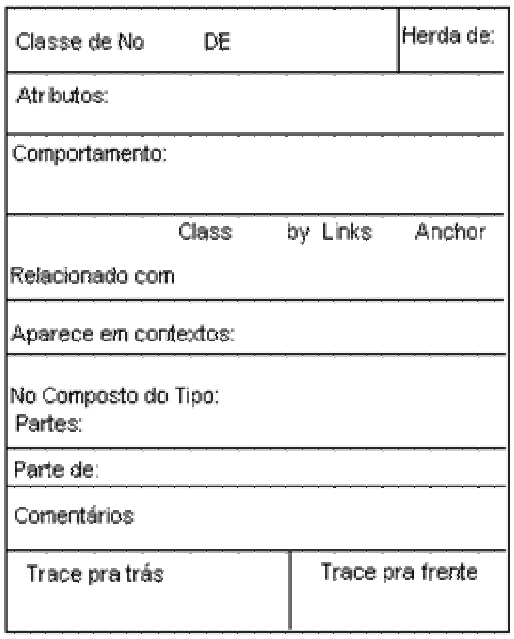
De acordo com Rossi (1996) os nós aparecem no esquema navegacional como pequenas caixas com estilo visual semelhante ao das classes conceituais, como demonstra a Figura 4.9. A especificação completa do nó é demonstrada nos Cartões de Nós definidos com o "*template*" da Figura 4.10.

Figura 4.9 – Nós com visual semelhante ao das classes conceituais.



Fonte: adaptado de Rossi (1996).

Figura 4.10 – Cartões de Nós.



Fonte: adaptado de Rossi (1996).

4.5.2 LINK'S

Link's ou ancoras refletem os relacionamentos que serão utilizados pelo usuário final são a realização navegacional dos relacionamentos. Classes de link's são definidas especificando-se atributos, comportamento do link, objeto fonte, alvo e cardinalidade. Atributos do link expressam propriedade maior que um. Em alguns casos o link pode se comportar como um nodo, agindo como um objeto intermediário (entre a fonte e o destino do link) durante a navegação (Schwabe, 1995).

Segundo Rossi (1996), as âncoras em OOHDM são tratadas como gatilhos navegacionais, isto é, quando uma âncora é selecionada, uma navegação ocorre. Esta interpretação sustenta a visão tradicional, em que as âncoras são apresentadas como objetos de interface física, sendo, além disto, a origem de elos ou estruturas de acesso. Em OOHDM a definição de âncora tem como parâmetro o tipo de elo ou estrutura de acesso originada naquela âncora; por exemplo :

- referência: Âncora (RefereA)

Indica que o atributo "referência" é uma âncora para um elo da classe de elo RefereA. No caso de a classe de Elo possuir subclasses, a âncora pode ser a origem dos elos em qualquer uma delas. O comportamento normal das âncoras quando recebem a mensagem "selecionada" é ativar a instância de elo correspondente. Este comportamento pode ser especializado em novas sub-classes, incluindo, por exemplo, algum teste de autorização. Tal tipo de comportamento especializado é comum em aplicativos de hipermídia educacionais, para levar o aluno a perseguir um elo somente após ter satisfeito algumas condições.

4.5.3 ESTRUTURAS DE ACESSO

De acordo com Schwabe (1995), estruturas de acesso agem como índices e são úteis para auxiliar o usuário final para encontrar a informação desejada. Menus, índices e tours são exemplos de estruturas de acesso. Estas estruturas de acesso são também modeladas como classes e mais tarde caracterizadas como um conjunto de selecionadores, um conjunto de objetos de alvo e um predicado em objetos-alvo. O predicado expressa quais objetos serão acessíveis em termos das suas propriedades. Selecionadores geralmente são usados para alguns atributos dos objetos-alvo e são organizados de acordo com a estrutura de dados pré-definida (uma lista ordenada, um conjunto de ícones). Em qualquer um dos casos, eles precisam ser explicitamente mencionados na definição da estrutura de acesso.

Segundo Rossi (1996), os diagramas de navegação visam refletir a dinâmica de um aplicativo de hipermídia, em termos de mudanças no espaço navegacional. Conseqüentemente, é necessário que permitam expressar facilmente quais nós são fechados e quais mantêm-se abertos, cada vez que um elo é atravessado. Deve-se oferecer um formalismo visual apropriado à compreensão de um determinado projeto, respondendo a perguntas como: que nós podem ser abertos simultaneamente e sob que condições? Ou, que nós são "mutuamente exclusivos?”, isto é, nunca estão acessíveis ao mesmo tempo. O espaço navegacional pode mudar de diversos modos, dependendo do contexto navegacional corrente.

Um diagrama de navegação de acordo com Rossi (1996), é composto basicamente de objetos navegacionais (principalmente nós e estruturas de acesso), estados e transições.Um mecanismo de agrupamento, chamado cluster, é utilizado para indicar quais os objetos que podem co-existir no espaço navegacional. Os clusters podem ser pensados como estados (o estado em que todos estes objetos podem ser ativados) ou como uma composição virtual

construída como uma agregação de tais objetos navegacionais. Dado um esquema Navegacional, ele é elaborado em um Diagrama Navegacional pela realização das seguintes etapas:

a) definição dos clusters, isto é, a indicação de quais objetos navegacionais são simultaneamente acessíveis. Em outras palavras, quando um elo é atravessado, se a sua origem permanece aberta;
b) anotação dos elos com ações a serem executadas quando da sua travessia.Os elos contendo um “marcador” em sua origem mantêm o nó de origem aberto, de outro modo, tal nó será fechado. Similarmente, quando um elo sai de um nó em um cluster para um nó em outro cluster, todos os nós no primeiro são fechados;
c) no caso de as transformações navegacionais dependerem dos contextos navegacionais, definição de cada contexto navegacional em que um nó pode ser visitado como um estado interno deste nó e a indicação de transições entre estados.

Há dois casos em que os diagramas de navegação podem ser omitidos:

a) quando o esquema navegacional expressa toda a informação necessária;
b) quando a dinâmica da navegação é completamente controlada pelo usuário final.

Na figura 4.11 demonstra-se os possíveis comportamentos do contexto de navegação.

Figura 4.11 - Demonstração dos possíveis comportamentos do contexto de navegação

Controle \ Restrição de Navegação		Navegação Livre	Navegação Exclusiva	Navegação Parcial
Automático		—	O usuário é guiado automaticamente através dos elos de contexto *	—
Passo a Passo	Acesso Direto (Índice)	O usuário pode acessar quaisquer elementos do contexto de navegação	O usuário só pode acessar elementos do contexto definidos no índice do contexto *	O usuário pode acessar elos “importados” de outros contextos de navegação
	Seqüencial	O usuário pode percorrer quaisquer elos *	O usuário só pode percorrer elos de contexto	O usuário pode percorrer elos “importados” de outros contextos de navegação

Fonte: adaptado de Cerqueira (1997).

4.6 DESIGN DA INTERFACE ABSTRATA

Nesta etapa do processo de modelagem, com a estrutura de navegação já definida, deve-se definir quais objetos irão interagir com o usuário, e principalmente a maneira como objetos de navegação diferentes vão ser visualizados, quais objetos irão ativar a navegação, a maneira como objetos multimídia serão sincronizados e quais transformações de interface irão ocorrer. Na metodologia OOHDM é utilizado o conceito de *Abstract Data View* (ADV), ou visão de dados abstrata, para descrever a interface da aplicação hipermídia.

De acordo com Vaananen (1993), para especificar o modelo abstrato de interface é necessário definir metáforas de interface e descrever suas propriedades estáticas e dinâmicas, assim como seus relacionamentos com o modelo navegacional de uma forma independente da implementação.É preciso especificar:

a) a aparência interfacial de cada objeto navegacional que será percebido pelo usuário, isto é, a representação de seus atributos (incluindo as âncoras). O mesmo objeto navegacional pode ter diferentes representações de interface em diferentes situações. Por exemplo, um nó pode ser representado por uma fotografia de um monumento ou por um ícone em um mapa que atue como um índice para monumentos;
b) outros objetos de interface para oferecer as diversas funções da aplicação, como barras de menus, botões de controle e menus;
c) os relacionamentos entre os objetos de interface e navegacionais, tais como o modo com que um evento externo, (o fato de o usuário "clicar" o "mouse", por exemplo) afetará a navegação;
d) as transformações de interface ocorridas pelo efeito da navegação ou de eventos externos no comportamento de diferentes objetos de interface;
e) e, finalmente, deve-se considerar a sincronização de alguns objetos de interface, especialmente quando há meios dinâmicos, como áudio e vídeo, envolvidos.

4.6.1 ABSTRACT DATA VIEWS(ADV)

De acordo com Carneiro (1994),o modelo de projeto ADV foi criado originalmente para especificar clara e formalmente a separação entre a interface do usuário e os

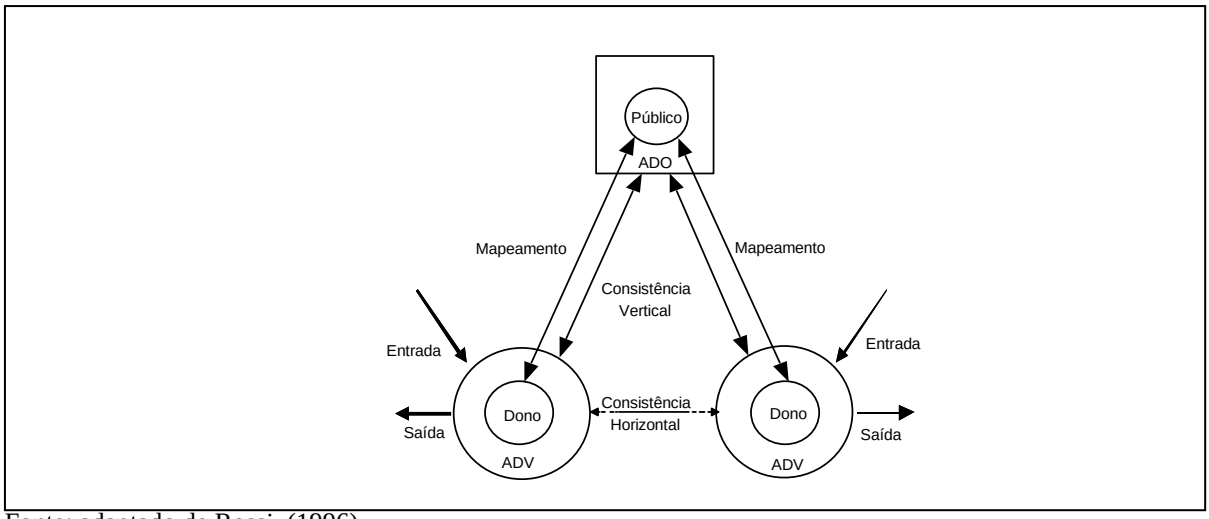
componentes de um sistema de software, e para oferecer um método de projeto independente de implementação, gerando graus mais altos de reuso de componentes de projeto e de inteface.

Os ADV's são objetos, no sentido em que possuem um estado e uma interface, e sua interface pode ser exercitada através de mensagens de outros objetos, ou de eventos de entrada e saída. E são abstratos no sentido em que representam apenas a interface e o estado, e não a implementação. Os ADV's são geralmente usados para representar interfaces entre duas mídias diferentes, tais como um usuário, uma rede ou um mecanismo como um marcador de tempo, ou como uma interface entre dois ou mais *Abstract Data Objects* (ADO's), sendo estes, ADV's que não suportam eventos externos (Rossi, 1996).

Para Rossi (1996), um ADO "possui", tipicamente, um ou mais ADV's que representam "para o mundo externo" algum aspecto de seu estado. Em OOHDM os objetos navegacionais, tais como os nós, os elos, ou as estruturas de acesso irão agir como ADO's e o ADV a eles associado será usado na especificação de sua aparência para o usuário.

Segundo Pinheiro (1999), cada ADV deve ser consistente com seu ADO proprietário e todos os ADV's possuídos por um ADO devem ser consistentes entre si. Estes dois tipos de relacionamentos de consistência são chamados, respectivamente, de consistência vertical e horizontal, representados na figura 4.12.

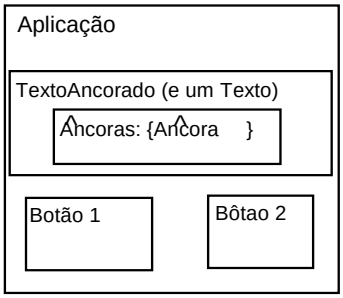
Figura 4.12 – Demonstração dos relacionamentos de consistência entre ADV's e ADO's.



Fonte: adaptado de Rossi (1996).

Segundo Rossi (1996), ADV’s podem ser organizados em hierarquias de generalização/especialização, oferecendo uma estrutura poderosa para a definição de hierarquias de objetos de interface. Na Figura 4.13 aparece o uso de herança em ADV’s.

Figura 4.13 – Exemplo de herança em ADV’s



Fonte: adaptado de Rossi (1996).

4.6.2 DIAGRAMAS DE CONFIGURAÇÃO

Os diagramas de configuração foram definidos originalmente no contexto dos *ObjectCharts* e são úteis na expressão de padrões de comunicação entre objetos, em termos de serviços oferecidos e requeridos. Na abordagem de projeto ADV, são usados para representar eventos externos (iniciados pelo usuário) manejados por um ADV: os serviços oferecidos pelo ADV (como "exibir") e a comunicação entre ADV’s e ADO’s (Coleman, 1992).

No contexto de OOHDM, concentra-se todo o interesse no modo como o usuário irá interagir com a aplicação hipermídia e, especialmente, quais objetos de interface causarão a navegação. Pelo fato de as classes navegacionais definirem uma interface pública, com os serviços oferecidos por seus objetos tem-se um modelo limpo de interação entre objetos de interface e navegacionais (Rossi, 1996).

4.7 IMPLEMENTAÇÃO

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

Para executar a implementação, o *designer* precisa mapear os modelos de interface abstrata e de navegação em objetos concretos disponíveis no ambiente de implementação escolhido. O modelo gerado depois que todas as etapas anteriores foram concluídas pode ser implementado de uma maneira direta utilizando as várias aplicações para este objetivo, já disponíveis no mercado, como por exemplo, Hypercard, Toolbook e Director. O uso de um conjunto de construtores de modelagem (objetos e classes) nesta metodologia permite uma transição suave da modelagem de domínio para o design de navegação e interface. Este passo da implementação não necessita um ambiente orientado à objeto, apesar que o mesmo possa tornar a tarefa mais fácil de ser realizada. No caso do ambiente de execução não ser orientado à objetos, muitas técnicas podem ser usadas para mapear uma especificação orientada à objetos para um ambiente que não seja. Os aspectos dinâmicos da interface lidam tanto com transformações de interface dentro de um ADV (por exemplo, mostrando/ocultando algum atributo), como com transformações de interface envolvendo navegação.

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

Segundo Schwabe (1999), aplicações em *Cd-Rom*, assim como *web-sites*, foram desenvolvidos utilizando a metodologia OOHDM.

Em aplicações hipermídia, a navegação ocorre porque o usuário seleciona uma âncora em um nó. É possível implementar comportamento navegacional mais sofisticado, como, por exemplo, navegação baseada em tempo, ou baseada em mídia (quando um evento determinado ocorre na mesma mídia, podemos proceder de modo a seguir um elo). Ambos os tipos de comportamento navegacional são especificados através de ADVcharts, já que as transições de estado e o evento gerador de cada mudança estão documentados na especificação da transição. Deve-se levar em conta a especificação das transformações navegacionais para saber se é necessário fechar o nó de origem antes de abrir o nó alvo (Rossi, 1996).

Para manter a informação de rastreabilidade acumulada durante o processo do projeto, deve-se documentar, além de todos os passos das etapas anteriores, as decisões de implementação. Apesar de esta documentação poder ser facilmente automatizada e

introduzida no ambiente de implementação, alguns cartões simples podem ser usados ao implementar a aplicação. Estes cartões são basicamente fichas que armazenam todas as informações documentacionais da aplicação, de todas as suas etapas, de uma maneira clara e objetiva (Rossi, 1996).

4.8 AMBIENTE MACROMEDIA DIRECTOR

No desenvolvimento do protótipo do ambiente para a criação de material didático multimídia, optou-se pelo software de autoria Macromedia Director, versão 8.0. Esta escolha deveu-se as facilidades oferecidas por este software para o desenvolvimento de aplicações hipermídia. Utiliza-se o Lingo, que é a linguagem do director e que permite a programação orientada a objetos.

No desenvolvimento de uma aplicação o Director utiliza como metáfora para o desenvolvimento de softwares hipermídia, a produção de um filme. Assim, da mesma forma que na produção de um filme deve-se definir (Bizzotto, 2000):

a) cenário (*Stage*): no Director, tudo ocorre no *stage*, ou seja, no palco. No stage são colocados todos os itens (cenário, atores, etc) que irão compor a cena a ser desenvolvida. Assim, o *stage* é o cenário básico, no qual irão ocorrer as cenas;
b) atores (*Cast Member*): um ator no Director é chamado de *cast member*. Assim, deve-se definir quais dos atores (cast *members*) de nosso elenco (Cast) irão participar da cena que se deseja realizar. No entanto o conceito de ator no Director é um pouco diferente. O Director considera como ator (cast member) todo elemento que aparece no stage;
c) texto (*script*): o script define como cada cast member (ator) deve se comportar no *stage* (palco);
d) figurino (propriedades do cast member): através da alteração das propriedades de um cast member (tamanho, cor, opacidade, etc), pode-se alterar sua aparência inicial;
e) seqüência (*score*) : o Director possui uma janela chamada *score*, através da qual podemos definir a seqüência na qual as cenas irão ocorrer. O score é organizado como uma planilha eletrônica, onde cada coluna denominada "*frame*" apresenta todos os cast members que aparecem no *stage* em uma dada unidade de tempo;

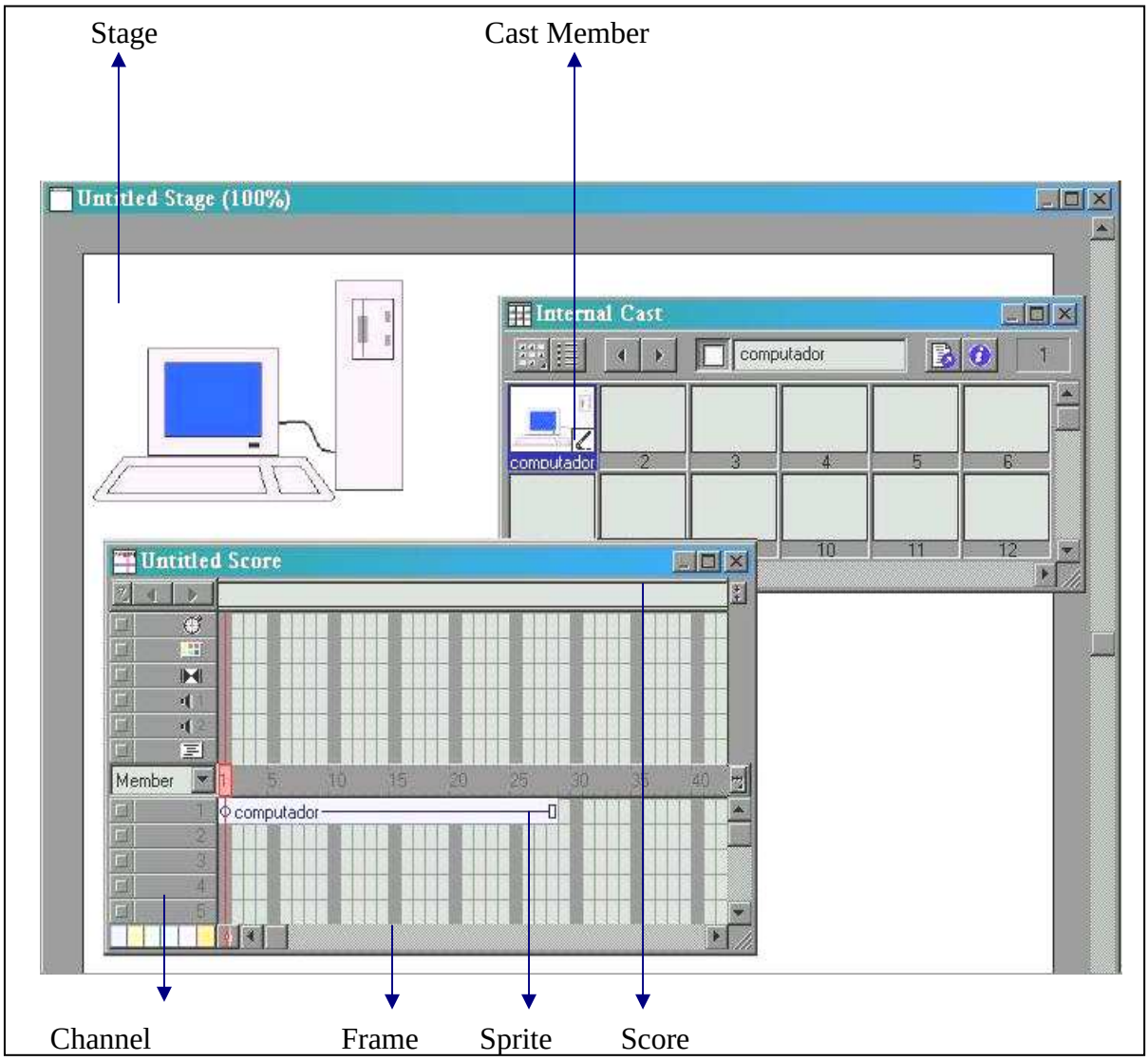
f) *frame*: um instante de tempo no director;

g) *channel*: posição numerada no score. Cada *sprite* ocupa um canal (channel) com uma série de *frames*. Cada linha no *score* é chamada de *channel*;

h) *sprite*: a descrição de cada *member* é mostrada, onde está dentro do score, onde aparece no stage, e muitas outras propriedades . Na figura 4.14 pode-se ver alguns elementos principais do Director;

i) testes (*control panel*): através da janela *control panel*, pode-se rodar o filme para verificar se tudo esta da forma desejada;

Figura 4.14 – Elementos do Director



Lingo é o nome da linguagem utilizada pelo Director para que seja possível introduzir interações mais efetivas com o usuário.Usando a linguagem de programação Lingo, pode-se dizer ao *sprite* o que fazer. Através do Lingo pode-se inserir algumas características a um dado filme (Macromedia, 1997):

a) exploração e navegação: o usuário pode escolher navegar e/ou explorar diferentes segmentos de um dado filme;
b) interatividade: através de *scripts*, o filme pode avaliar e responder às ações dos usuários;
c) “filmes em janelas”: pode-se rodar vários filmes ao mesmo tempo. Os filmes podem rodar independentemente um do outro;
d) manipulação do texto: pode-se combinar e editar um conjunto de caracteres ao longo do filme;
e) interface: pode-se criar menus, botões, checkboxes e beeps;
f) controle de sons: pode-se controlar o volume, entrada e saída de sons.

Um *script* (texto) em lingo, que é ligado a um *sprite* no *score* é chamado *behavior*. Este tipo de script mostra ao sprite como se comportar em diferentes circunstâncias. Existem diferentes tipos de script:

a) behavior script: são scripts associados a sprites (sprite behaviors) ou a frames (frame behaviors);
b) cast member script: associado a um dado cast member, de forma que o script será executado em qualquer sprite do qual aquele cast member faça parte;
c) movie script: são scripts que “valem” para todo o filme. Assim, eles são mais gerais, não se limitando a um dado sprite ou conjunto de sprites ou frames.

Eventos que ocorrem quando o software é inicializado:

a) *prepareMovie*: este é o primeiro evento que ocorre quando um filme é executado, ou seja, o filme, está sendo preparado para ser rodado. É uma boa oportunidade para avaliar se o computador do usuário possui as características para rodar o filme;
b) *beginSprite*: indica que a cabeça de leitura e gravação entrou em um segmento de *sprite*, ou seja, indica que um sprite foi iniciado. É uma boa oportunidade para iniciar as propriedades de um dado *sprite*;

c) *prepareFrame*: ocorre antes do director apresentar um sprite no stage. Assim, nesse evento pode-se incluir ações que alterem as propriedades do *sprite*, uma vez que ele ainda não foi "desenhado";

d) *startMovie*: este evento ocorre no primeiro *frame* do filme.

De acordo com Henderson (1997), da mesma forma que outras linguagens de programação, o Lingo possui regras que devem ser seguidas:

a) todo script é composto por "handlers": geralmente os "handlers" são chamados de subrotinas ou funções em outras linguagens de programação. Quando se chama os handlers, estes executam uma ou mais funções, conforme exemplo:

```
on mouseDown
   beep
end
```

Deve-se observar que o handler começa com a palavra "on" e termina com a palavra "end". Isto deve ser seguido por todo e qualquer handler desenvolvido em Lingo.

b) variáveis locais e globais: o lingo utiliza dois tipos de variáveis:
   - variáveis locais: são variáveis criadas e utilizadas dentro de um handler, não estando disponíveis para outros handlers;
   - variáveis globais: são variáveis que podem ser usadas por diferentes frames em um filme e/ou diferentes filmes;

c) orientação a objetos: segundo Small (1996), o Lingo não pode ser considerado uma linguagem orientada a objetos, como o Delphi, por exemplo, mas sim como uma linguagem que permite a programação orientada a objetos, pois oferece o recurso de utilizar variáveis globais privadas, ou seja, onde cada objeto possui características próprias. O lingo possui três características que permitem a orientação a objeto:

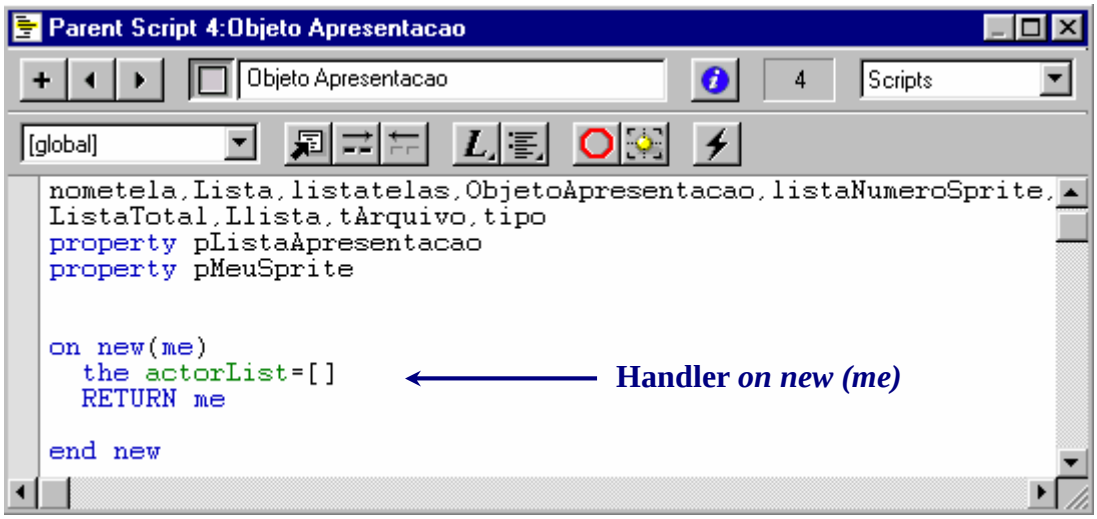
   - a função "NEW": a qual permite que se faça cópias idênticas de um objeto, através de um script. Denominado parent script;
   - "property": a qual permite que os objetos mantenham suas próprias variáveis globais privadas (cada objeto possui características únicas);
   - Ancestor: a qual permite que os objetos dividam propriedades e *handlers*.

Em Lingo, define-se um modelo de objeto, escrevendo um *parent script. Parent scripts* são como modelos para códigos de objetos. Pode-se criar um objeto daquele *parent script* enviando uma mensagem "*new*" para o *parent script*. Para criar um objeto, em primeiro lugar precisa ser criado um parent script. Um parent script define os métodos e as propriedades de objetos criados a partir dele.

Neste sentido, o handler "*on new me*" (figura 4.15), cria um novo objeto a partir de um *parent script*. Da mesma forma como se pode criar vários números de identidade de documentos a partir de um modelo, pode-se criar quantos objetos de identificação quiser, chamando o método *new* de um *parent script*. Toda vez que se manda uma mensagem para um *parent script*, ele cria um novo objeto a partir daquele *parent script*.

É importante usar a palavra *me* na linha que começar com *on*. A palavra *me* reserva um ponteiro para a localização na memória do objeto criado a partir do *parent script*. Importante também é incluir a palavra *return* dentro do método "*on new*". Esta linha retorna um ponteiro para o objeto. Se não for incluída a palavra *return*, não haverá comunicação com o objeto depois de criado.

Figura 4.15 *: Handler on new*



A figura 4.16 demonstra um exemplo de criação de objeto. Esta linha de código *tObjetoApresentacao= new(script "Objeto Apresentacao")* cria um novo objeto assim que o programa é iniciado, e coloca o valor do ponteiro para o objeto na variável tObjetoApresentacao. A linha de comando *(the actorlist).append(tObjetoApresentacao)* adiciona este valor especifico, no caso a posição da memória onde foi criado o objeto, no final

da lista *the actorlist*. Esta lista (the actor List) contém todos os objetos que foram criados em um dado executável, mantendo-se ativa enquanto o executável estiver ativo na memória. Assim, em qualquer parte do executável é possível enviar uma mensagem para um objeto que está na *actorList*, de forma que um método específico seja realizado. Para isso, pode ser utilizado o script mostrado na figura 4.17.

Figura 4.16: Criando objeto

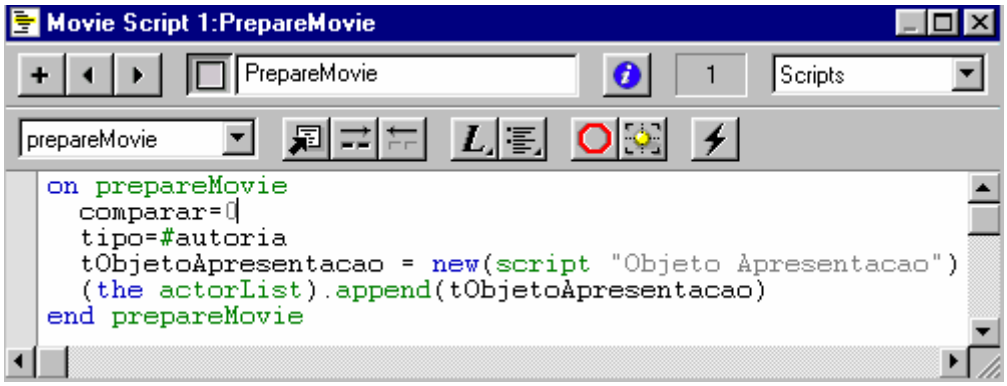
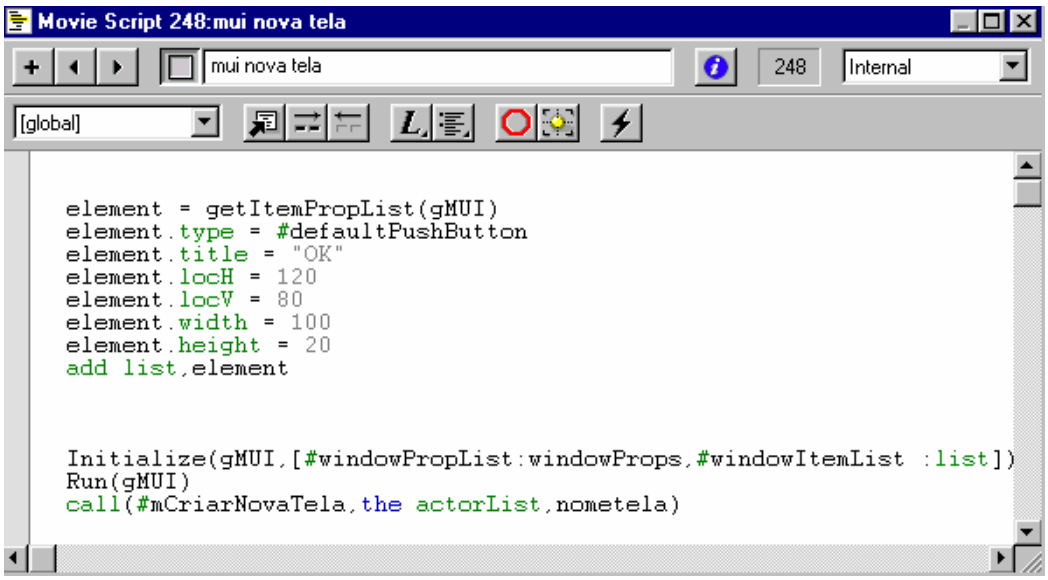


Figura 4.17 – Enviando mensagem para um objeto que está na actorList



5 DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo será apresentada a especificação do protótipo, com a utilização da ferramenta *Rational Rose*, serão apresentados os diagramas de caso de uso, diagrama de classes, esquema conceitual e esquema navegacional. Será apresentada também a descrição da utilização do protótipo.

5.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Nos dias atuais, os professores encontram uma grande dificuldade no uso da informática. Até a simples escolha de um software pode ser frustrante. Primeiro porque muitos professores ainda têm um certo receio em relação à informática, imaginam que para utilizar um programa, precisam antes de tudo fazer um curso, e se aprofundar mais com o software. O que muitas vezes pode ser considerado verdade. A maioria dos softwares é de difícil compreensão, o que se torna uma barreira para os professores, que estão interessados em encontrar um ambiente de fácil utilização, que possa se adequar as suas necessidades.

Para satisfazer a estas dificuldades encontradas pelo professor, o presente trabalho de conclusão de curso, é um protótipo de um ambiente para criação de material didático multimídia. Este protótipo possibilita o professor inserir um texto, inserir uma imagem, que pode ser importada de um arquivo, ou pode ser desenhada no próprio protótipo, pois o mesmo possui um editor de imagens tipo paint do windows, inserir vídeo, e inserir botões para navegação. Possibilita também que o professor mande E-mail’s para seus alunos, através do protótipo, sem precisar abrir um outro programa para fazer esta operação.

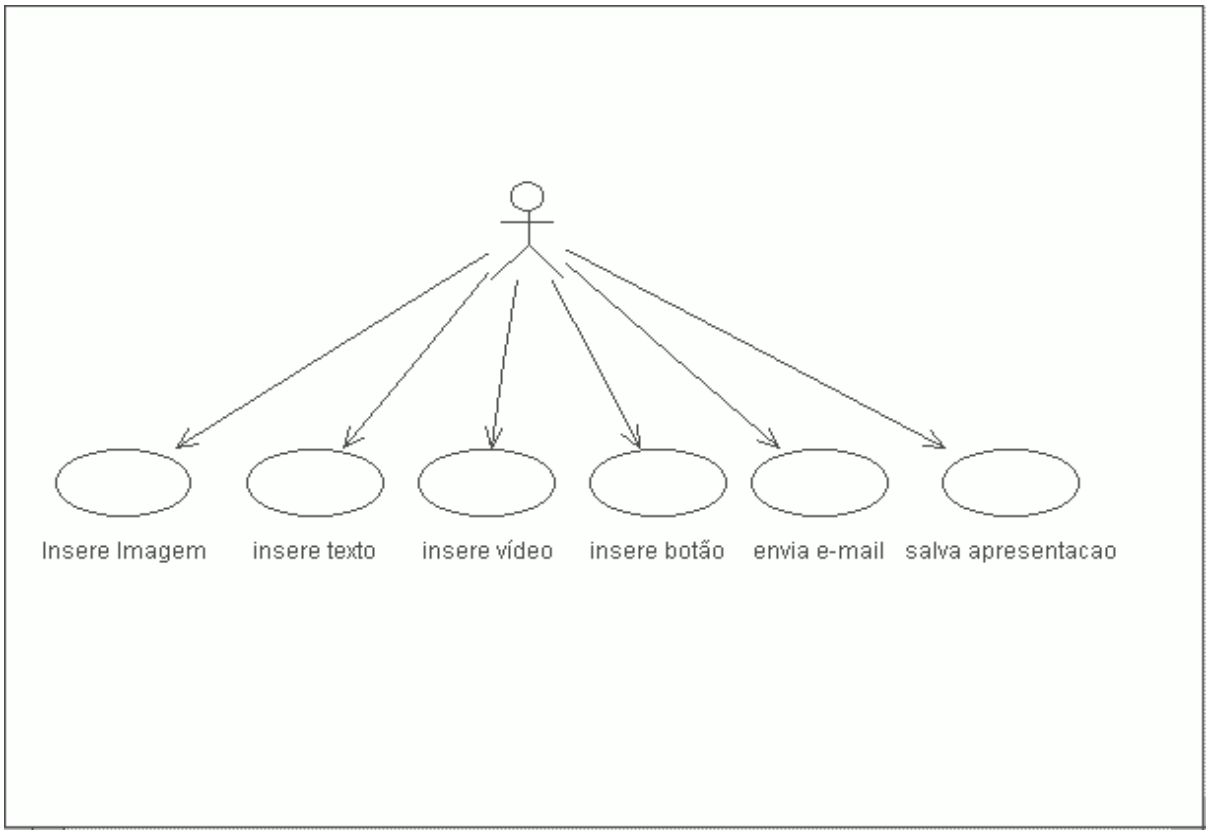
Este software permite a inclusão de mídias e criação de link’s através de botões para ligar telas diferentes.

5.2 ESPECIFICAÇÃO

Visando a utilização da metodologia orientada a objeto OOHDM, citada no capítulo 4, foi desenvolvido este protótipo, de forma a realizar um trabalho de análise e especificação para posterior implementação.

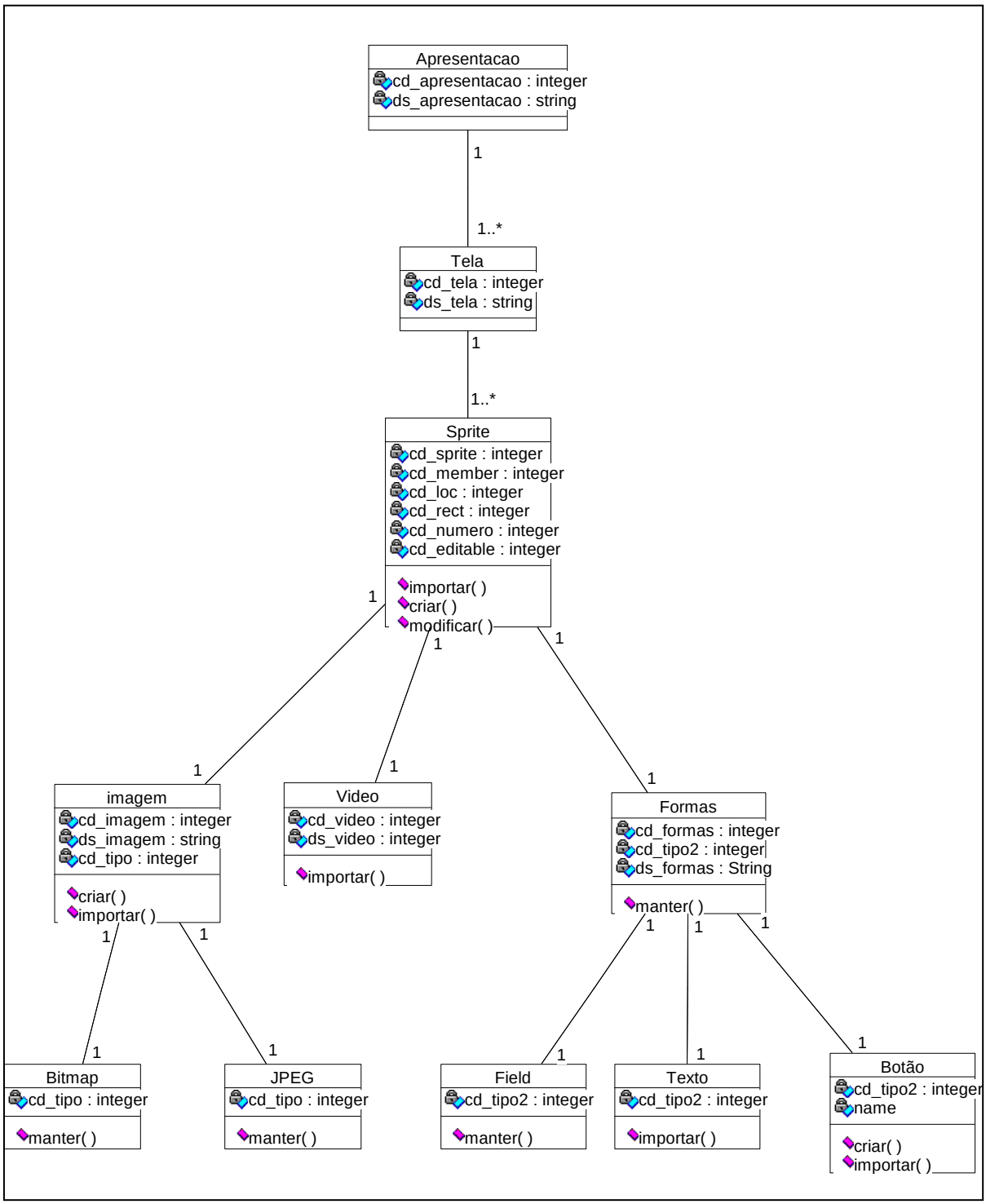
Através da ferramenta *Rational Rose*, é apresentada a especificação formal do problema utilizando o diagrama de caso de uso (Figura 5.1). Para o desenvolvimento deste diagrama é necessário saber quais as principais tarefas relacionadas com o problema e quem irá executar estas tarefas.

Figura 5.1– Diagrama de caso de uso



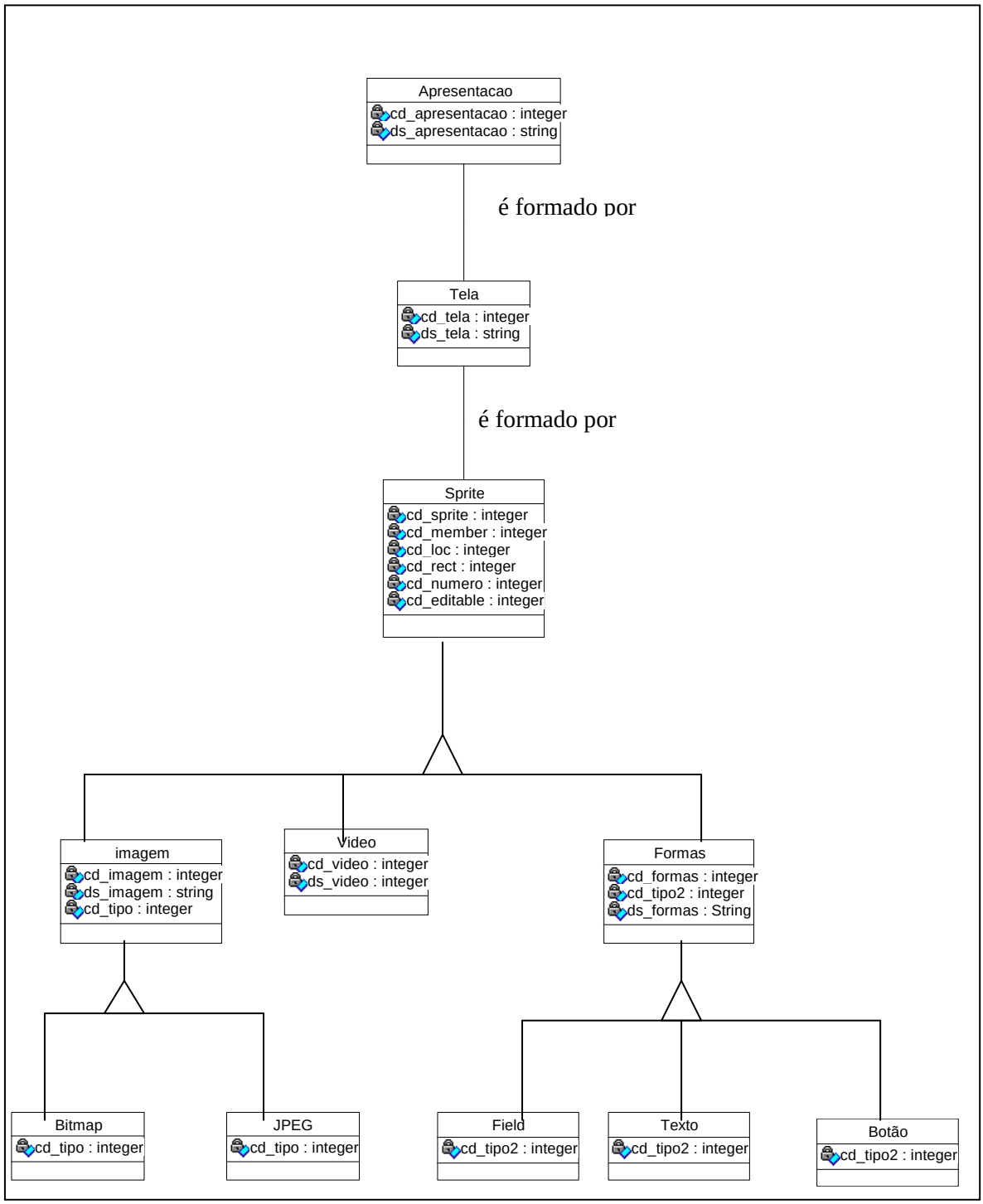
Na figura 5.2, é apresentado o diagrama de classes, que demonstra os relacionamentos entre as classes. Através deste diagrama são demonstradas quais informações precisam ser guardadas e de quais objetos.

Figura 5.2 – Diagrama de classes.



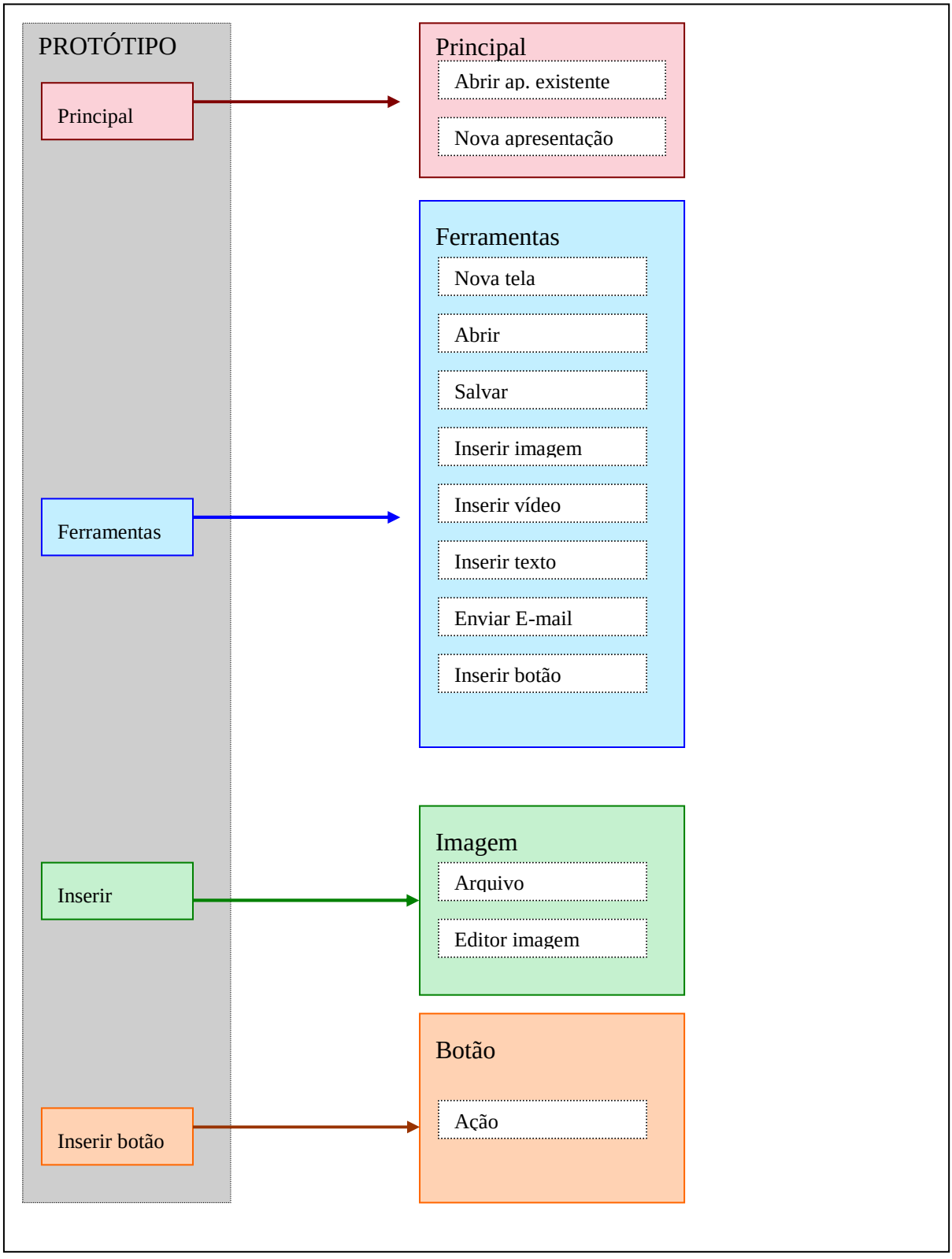
Através da Metodologia OOHDM, uma outra maneira para demonstrar o relacionamento entre as classes é apresentado. Chamado de esquema conceitual é constituído basicamente por classes, relações e subsistemas, conforme demonstrado na figura 5.3.

Figura 5.3- Esquema conceitual.



Para o melhor entendimento do protótipo, a figura 5.4 demonstra o esquema navegacional. Através deste diagrama, que faz parte da metodologia OOHDM, pode-se entender como cada função do protótipo está relacionada.

Figura 5.4 – Esquema navegacional.



Através da ferramenta *Rational Rose*, pode-se desenvolver a especificação do protótipo. A partir deste momento, inicia-se a descrição da implementação do protótipo, através da ferramenta Macromedia Director 8.0, utilizando a programação orientada a objetos.

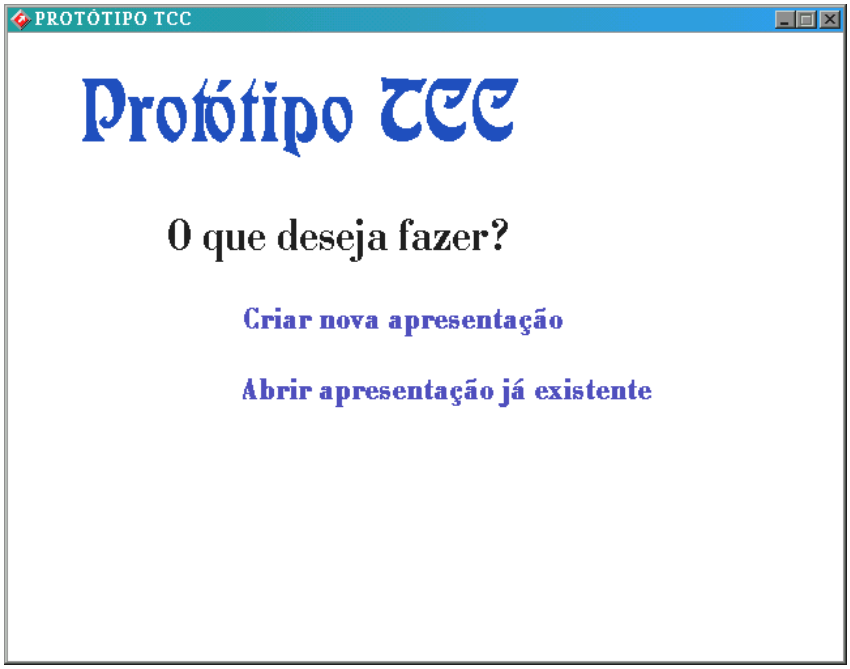
5.3 IMPLEMENTAÇÃO

5.3.1 OPERACIONALIDADE DA IMPLEMENTAÇÃO

A seguir será apresentada a operacionalidade da implementação. Onde serão mostradas as telas do protótipo, bem como as características de cada uma delas, e alguns códigos. Foi utilizada para implementação deste protótipo o software Macromedia Director 8.0, descrito no capítulo 4.8.

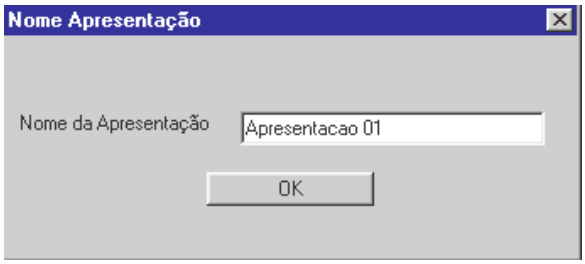
A figura 5.5 é a primeira tela, sendo a abertura do ambiente. Nesta tela o usuário deve decidir se deseja criar uma nova apresentação, ou se deseja abrir uma apresentação já existente, ou seja, uma apresentação que já foi criada anteriormente.

Figura 5.5 - Tela Apresentação



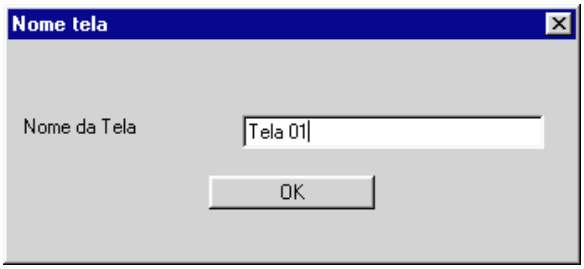
Se o usuário, neste caso, o professor, desejar criar uma apresentação, então ele irá clicar em “Criar nova apresentação”, uma “caixa de diálogo” irá surgir pedindo para o usuário digitar o nome da apresentação que ele irá criar, conforme figura 5.6.

Figura 5.6 – Caixa de diálogo para digitar o nome da apresentação



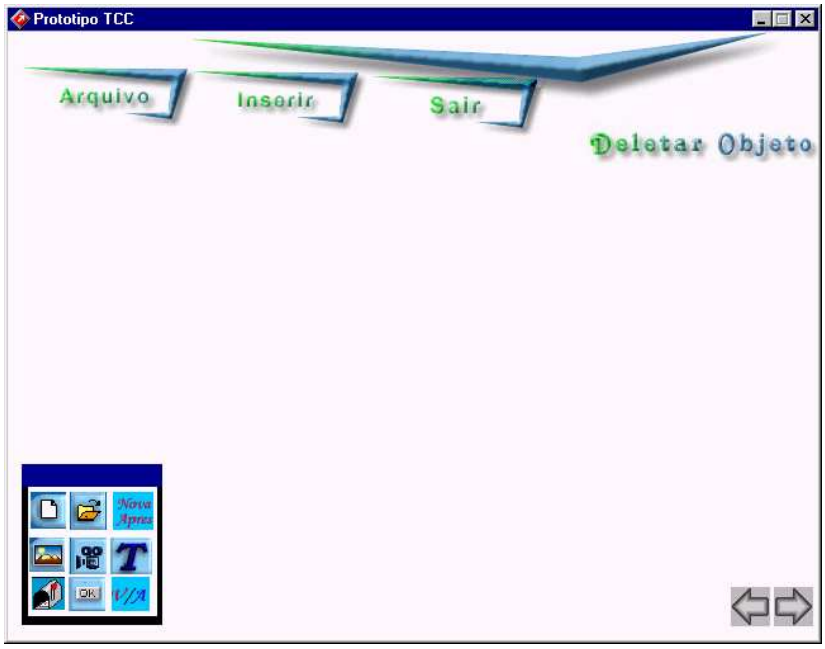
Quando o usuário terminar de digitar o nome da apresentação, e clicar “OK”, outra “caixa de diálogo” irá se abrir, esta agora solicitando o nome da primeira tela que será criada, conforme figura 5.7.

Figura 5.7 – Caixa de solicitação do nome da primeira tela.



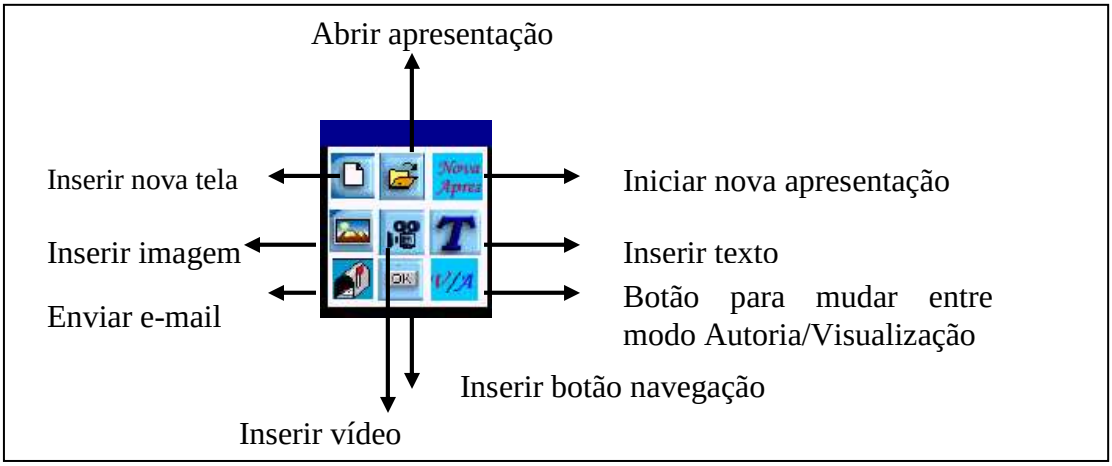
Após digitar o nome da tela, e clicar sobre “OK”, o programa irá abrir a tela do protótipo, conforme figura 5.8.

Figura 5.8 – Tela principal do protótipo



Na tela principal encontra-se o menu, com as mesmas funções disponíveis na barra de ferramentas, um botão “Deletar Objeto”, que possibilita ao usuário deletar imagens, textos, vídeos e botões que desejar, e duas setas que permitem ao usuário percorrer as telas criadas. Na barra de ferramentas (Figura 5.9), pode-se ver as funções do protótipo, como inserir nova tela, abrir apresentação, iniciar nova apresentação, inserir imagem, inserir vídeo, inserir texto, enviar E-mail, inserir botão de navegação e mudar entre modo Visualização/Autoria.

Figura 5.9 – Barra de ferramentas



Clicando na opção inserir imagem  , irá abrir uma “caixa de diálogo” (figura 5.10) onde o usuário (professor), terá as opções de inserir uma imagem que já está no seu computador, ou desenhar utilizando um editor de imagens tipo paint do Windows. O código correspondente à função inserir imagem pode ser visto na Figura 5.11.

Figura 5.10 – Inserir imagem

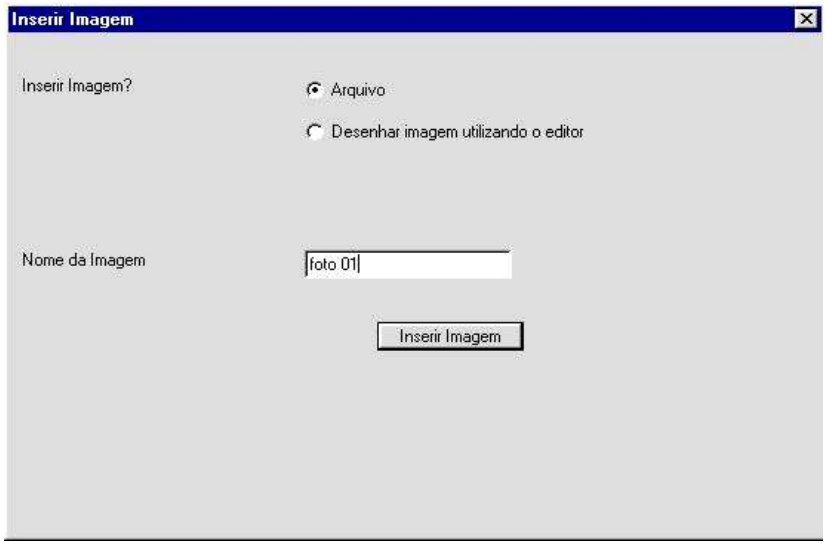
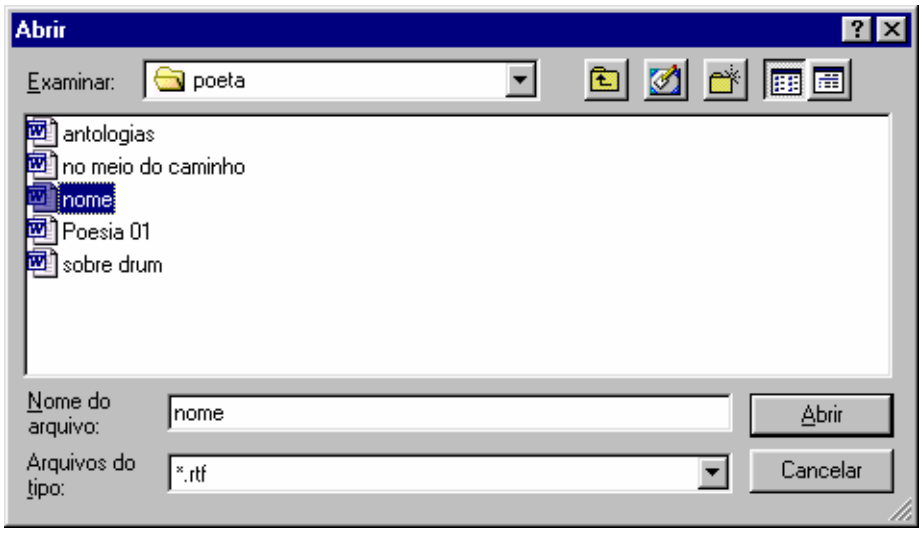


Figura 5.11 - Código correspondente à função inserir imagem

```
on muiFileOpen me
  fileIO=new(xtra "fileio")
  setFilterMask fileIO,"*.jpg,*.jpg,*.gif,*.gif,*.png,*.png,*.bmp,*.bmp"
  displayOpen(fileIO)
  ii=the result
  new(#bitmap)
  teste=new(#bitmap,castLib "Salvar")
  member(teste).filename=ii
  call(#mverificarsprite,the actorList)
  sprite(varsprite).member=member(teste).member
  sprite(varsprite).rect=member(teste).rect
  updatestage
end mouseUp me
```

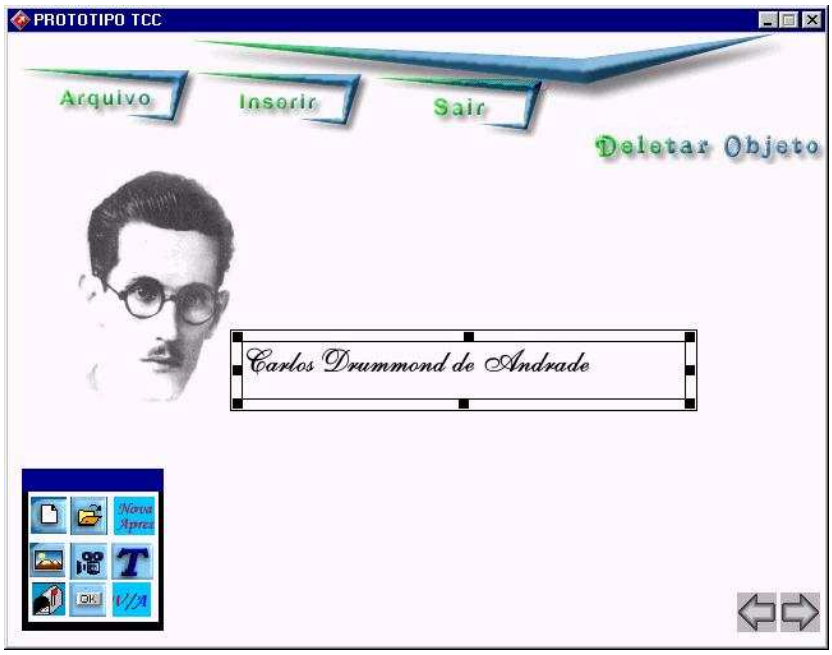
Outra opção será inserir um texto na apresentação. Assim que clicar na barra de ferramentas na opção para inserir texto  , uma “caixa de diálogo” para inserir textos do tipo (.txt, .rtf, .htm) irá se abrir (Figura 5.12), e o usuário poderá importar o texto que desejar.

Figura 5.12 – Inserir texto



A figura 5.13 demonstra como ficou a primeira tela construída pelo professor.

Figura 5.13 – Primeira tela



O usuário pode então criar uma segunda tela. Clicando em inserir nova tela  , uma "caixa de dialogo" aparecerá (Figura 5.14) e o usuário então digita o nome da nova tela. Na figura 5.15 é possível ver parte do código utilizado para inserir uma nova tela.

Figura 5.14 – Propriedades nova tela

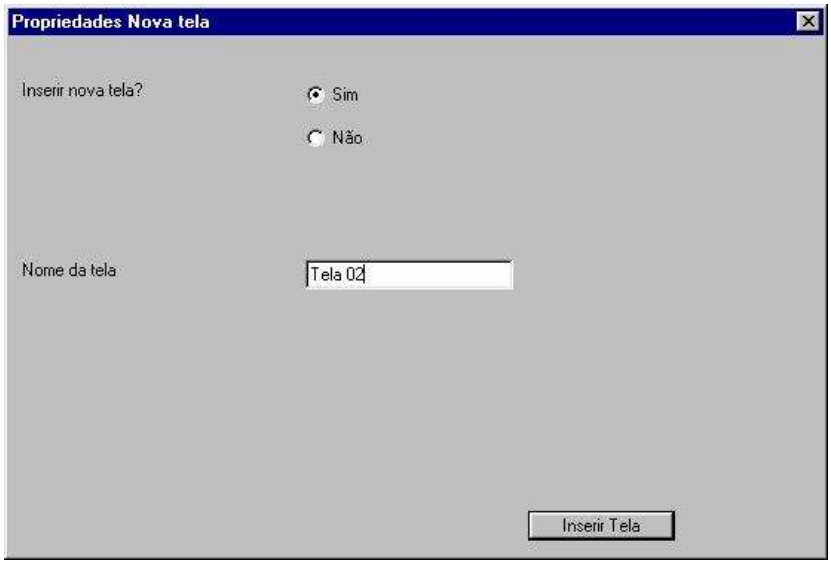


Figura 5.15 – Parte do código utilizado para inserir uma nova tela.

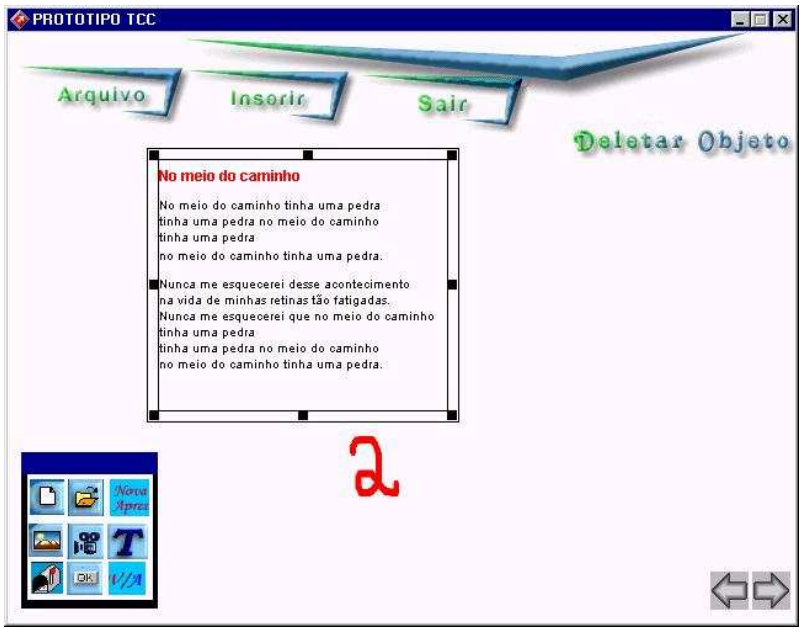
```
if action = #itemChanged and  elementList.tip="novapg" then

  newval5= elementList.title
  if newval5="Sim" then
    a=1
    call(#mLimparSprite,the actorList)
    repeat with k=40 to 49
      sprite(k).locv=10000
      sprite(k).locH=10000
    end repeat
    telaatual=nometela
    m=getOne(listatelas,telaatual)
    proximatela=listasprite[nomeap][m+1]
    telaanterior=listasprite[nomeap][m-1]
  end if
  if newval5="Não" then
    a=0
    Stop(gMUI,0)
    gMUI = VOID
  end if
```

Uma nova tela em branco irá aparecer e o usuário poderá então inserir o que desejar. Nesta segunda tela o usuário então pode optar por inserir um texto e uma imagem utilizando o ambiente tipo “paint” do Windows, neste caso foi desenhado o numero 2, como mostra a figura 5.16.

Figura 5.16 – Ambiente para criação de imagens.

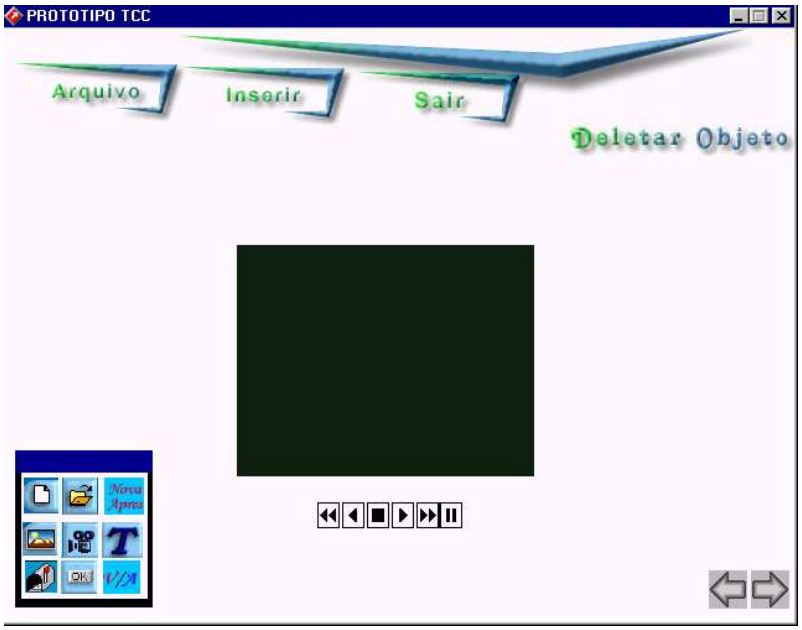
Quando clicar sobre “retornar” esta imagem que foi desenhada irá aparecer na tela juntamente com o texto inserido anteriormente, como mostra a figura 5.17.

Figura 5.17 – Tela 02



O usuário pode então inserir uma nova tela, e nesta tela inserir um vídeo, como mostra a figura 5.18. O espaço em preto demonstra o local onde o vídeo irá aparecer.

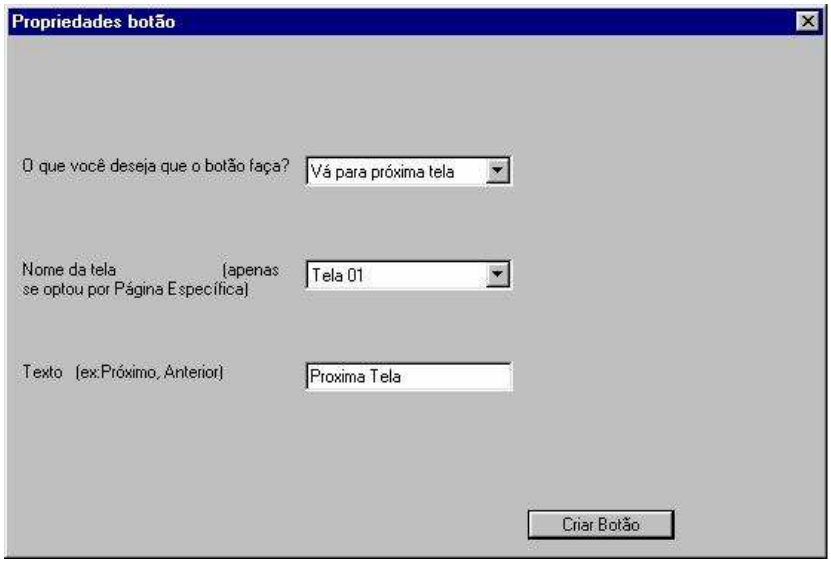
Figura 5.18 – Tela 03



Através dos botões  , o usuário poderá percorrer todas as telas que criou, e modificar o que desejar. Poderá então inserir os botões que auxiliarão na navegação. Na

primeira tela, pode inserir um botão, com a característica de que quando o mesmo for clicado, irá passar para a próxima tela, neste caso “Tela 02”. A “caixa de diálogo” contendo as informações necessárias para a criação de um botão pode ser vista na figura 5.19.

Figura 5.19 – Propriedades do botão



A figura 5.20 demonstra quais as funções disponíveis em um botão (Vá para próxima tela, Vá para tela anterior, Vá para tela específica, Vá para primeira tela).

Figura 5.20 – Funções do botão.

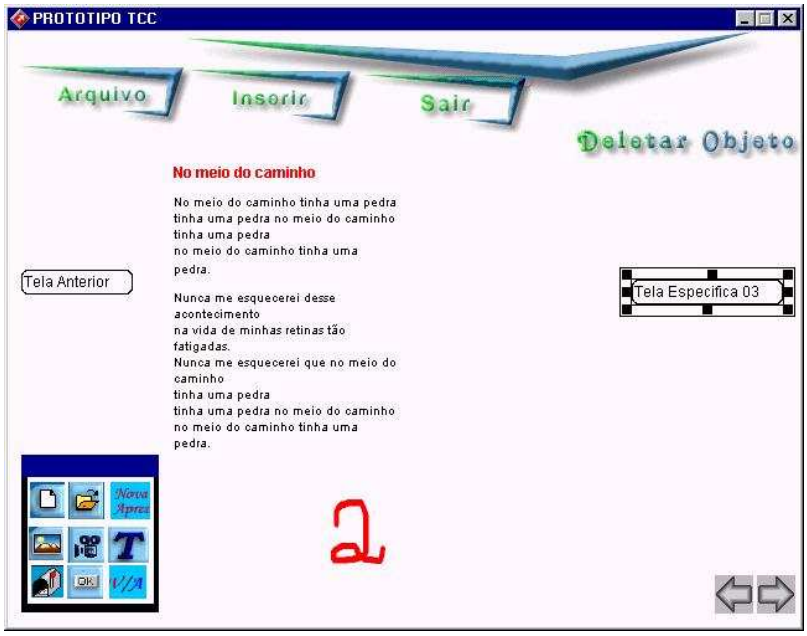
A figura 5.21 demonstra parte do código utilizado para executar as funções do botão, quando o mesmo for clicado durante a apresentação.

Figura 5.21 – Parte do código para executar as funções do botão.

```
on acao me
  case b of
    "proxima"   : proxima
    "anterior"  : anterior
    "especifica": especifica
    "primeira"  : primeira
  end case
end acao me
on proxima me
  call(#mLimparSprite,the actorList)
  repeat with k=40 to 49
    sprite(k).locH=10000
    sprite(k).locV=10000
  end repeat
  m=getOne(listatelas,nometela)
  n=tLista[nomeap][m+1]
  c=count(n)
  t=m+1
  nometela=listatelas[t]
  repeat with i=1 to c
    if tLista[nomeap][m+1][i]<>[] then
      call(#mverificarsprite,the actorlist)
      sprite(varsprite).type=tLista[nomeap][m+1][i][1]
      sprite(varsprite).member=tLista[nomeap][m+1][i][2]
      sprite(varsprite).width=tLista[nomeap][m+1][i][5]
      sprite(varsprite).height=tLista[nomeap][m+1][i][6]
      sprite(varsprite).rect.top=tLista[nomeap][m+1][i][7]
      sprite(varsprite).rect.left=tLista[nomeap][m+1][i][8]
      sprite(varsprite).rect.right=tLista[nomeap][m+1][i][9]
      sprite(varsprite).rect.bottom=tLista[nomeap][m+1][i][10]
      sprite(varsprite).loch=tLista[nomeap][m+1][i][3]
      sprite(varsprite).locv=tLista[nomeap][m+1][i][4]
    end if
  end repeat
  updatestage
end proxima me
```

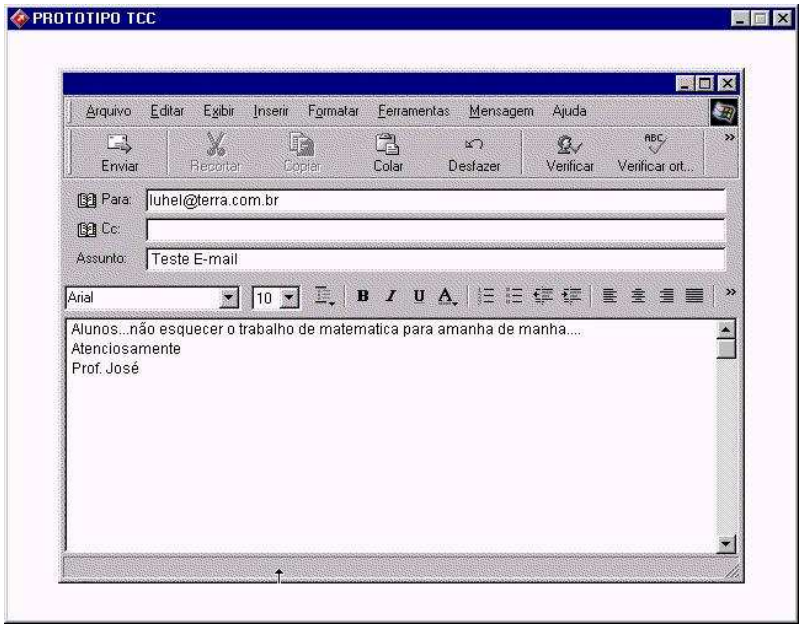
A figura 5.22 demonstra a tela 02 com os botões para ir para tela anterior “Tela 01” e para a tela específica “Tela 03”.

Figura 5.22 – Tela 02 com botões, no modo visualizar.



A partir daí o usuário poderá incluir quantas telas desejar, com seus textos, imagens, vídeos, botões, deletar o que inseriu e poderá inclusive enviar E-mail, como mostra a figura 5.23. Quando clicar sobre o botão para enviar E-mail esta tela irá abrir, dentro do protótipo, e o usuário poderá então enviar E-mail's para quem desejar.

Figura 5.23 – Envio de E-mail



Para que os botões que foram inseridos para facilitar a navegação estejam habilitados a executar as funções, a apresentação deve estar no modo visualização, para mudar entre modo autoria/visualização é necessário clicar no botão  .

Outra opção é abrir uma apresentação já foi criada anteriormente. A figura 5.24 demonstra o código necessário para a função “abrir apresentação”.

Figura 5.24 – Parte do código para abrir apresentação

```
on mAbrirApresentacao(me,v)
  repeat while v=1
    fileXtra = new(xtra "fileXtra3")
    tArquivo = fx_FileOpenDialog(fileXtra, "c:\Windows\Desktop\Apresentações",
"Apresentacoes/*.*","Abrir Apresentação",0,0)
    if tArquivo = "" then
      alert "Este tipo de formato não é válido"
      abort
      t=false
    else
      tLista=[:]
      tLista = loadPropsFromDisk(tArquivo)
      if listP(tlista) then
        pListaApresentacao = tLista
        v=2
      end if
    end if
  end repeat
  listatelas=[]
  listasprite=[:]
  nomeap=Tlista.getpropat(1)
  nometela=Tlista[nomeap].getpropat(1)
  cont=count(Tlista[nomeap])
  repeat with i=1 to cont
    nomet=Tlista[nomeap].getpropat(i)
    add listatelas,nomet
  end repeat
  tipo=#visualizacao
  repeat with k=40 to 49
      sprite(k).locv=10000
      sprite(k).locH=10000
    end repeat
  restaurar4 -→ chama a função que irá restaurar todos os sprites da apresentação que
                desejo abrir
  abort
  end mAbrirApresentacao
```

Outra função disponível neste protótipo é a função para redimensionar textos e imagens(Figura 5.25).

Figura 5.25 – Imagem mostrando redimensionar.

Parte do código criado para que seja possível este redimensionamento pode ser visto na Figura 5.26.

Figura 5.26 – Parte do código para função redimensionar

```
on mouseDown(me)
  pSpriteTexto=numerosprite
  tCliquePos = the mouseLoc
  tDiferenca = tCliquePos - pMeuLoc
  sendSprite(sprite pSpriteTexto,#mRedimensionando)
  repeat while the stillDown
    tMousePos = the mouseLoc
    case pNumeroAlca of
      #1:
        repeat while the stillDown
          -- monitorar posição do mouse
          tMousePos = the mouseLoc
          pDistancia = tMousePos - tCliquePos
          tCliquePos = tMousePos
          dif = [pDistancia.locH,pDistancia.locV]
          rectPos = [1,2]
          tRect = sendSprite(sprite pContornoInterno,#mAlteraContorno,rectPos,dif)
        end repeat
      #2:
        repeat while the stillDown
          tMousePos = the mouseLoc
          pDistancia = tMousePos - tCliquePos
          tCliquePos = tMousePos
          dif = [pDistancia.locV]
          rectPos = [2]
          tRect = sendSprite(sprite pContornoInterno,#mAlteraContorno,rectPos,dif)
        end repeat
       -- para todas as oito alças
  sendSprite(sprite pSpriteTexto,#mFimRedimensionamento,\
             tRect,pMargem)
end mouseDown
on mCalcularFatores(me)
  pSpriteTexto=numerosprite
  case pNumeroAlca of
    #1:
      pFatorCorrecaoHor = -(member(pMeuMember).width)
      pFatorCorrecaoVert = -(member(pMeuMember).height)
    #2:
      pFatorCorrecaoHor = 0
      pFatorCorrecaoVert = -(member(pMeuMember).width)
-- para todas as oito alças
  end case
end mCalcularFatores

on mPosicionarNoStage(me,tListaPosAlcas,tTextoRect)
  pSpriteTexto=numerosprite
  pTextoRect = tTextoRect
  if voidP(tListaPosAlcas) then
    RETURN #Error
  else
    pMeuLoc = tListaPosAlcas[pNumeroAlca]
    pMeuLoc = point(pMeuLoc.locH + pFatorCorrecaoHor,\
                    pMeuLoc.locV + pFatorCorrecaoVert)
```

```
    pMeuOrigLoc = pMeuLoc
  end if

  sprite(pMeuSprite).loc = pMeuLoc
  UpdateStage

end mPosicionarNoStage
```

5.4 RESULTADOS E DISCUSSÃO

Foi utilizada para a criação deste software a versão 8.0 do software Macromedia Director.

Obteve-se como resultado as seguintes funções:

a) Importação: o software permitirá a importação de diversos tipos de arquivos, de forma a satisfazer as necessidades de grande parte dos professores. Dentre os tipos de arquivos que poderão ser importados, temos:

- Textos no formato RTF, TXT e HTML.
- Imagens no formato BMP, GIF, JPG, TIFF e PNG.
- Vídeos no formato AVI, Quick Time e MPEG.

b) Autoria: Neste *módulo* destacam-se as seguintes funções:

- Botões, com rotinas associadas (ir para a próxima página, voltar para a página anterior, ir para uma determinada página etc.).
- Criação de desenhos (tipo Paint do Windows 98).

c) Interação: neste módulo foi incluída uma função que permita a interação efetiva entre professor e alunos. Esta função é:

- E-mail: alunos e professor poderão trocar e-mail entre si.

6 CONCLUSÕES

O objetivo principal do trabalho, qual seja, a implementação de um ambiente baseado em computador para a criação de material didático hipermídia, foi atingido. Durante a implementação do referido ambiente, observou-se que o Director 8.0 mostrou-se adequado para o desenvolvimento de ambientes de autoria.

O ambiente de autoria hipermídia desenvolvido, permite ao professor criar seu próprio material didático multimídia. Pode-se observar que o protótipo possui recursos necessários para a criação de materiais didáticos hipermídia, por parte de professores que não são da área de informática.

Com relação aos objetivos específicos apresentados no início do trabalho, observa-se que todos foram cumpridos permitindo que o professor insira textos, sons, vídeos e imagens, botões e tornando possível a comunicação entre professor e alunos através de *E-mail*.

O ambiente proposto utiliza os princípios da cooperação (as apresentações criadas podem ser compartilhadas entre alunos e professores), mas neste protótipo apenas permite o envio de E-mail, as funções para o recebimento não foram concluídas, construção (permite criar imagens, botões) e interação (o sistema não só reage ao usuário, ele se modifica em função do histórico de interações). Com isso, o ambiente proposto diminui os principais problemas apresentados pela maioria dos softwares educacionais atuais: centralização e inflexibilidade.

6.1 EXTENSÕES

A seguir serão apresentadas sugestões para aperfeiçoar o trabalho desenvolvido:

a) criação de um ambiente para utilizar os princípios da cooperação através da internet, onde os professores poderão trocar seus materiais, ou disponibilizar na rede para quem desejar;
b) implementação das funções para a criação de animações;
c) implementação das funções para permitir a criação e correção de testes desenvolvidos pelo professor.

REFERÊNCIAS BIBLIOGRÁFICAS

ALENCAR, Eunice Soriano de. **Novas contribuições da psicologia aos processos de ensino e aprendizagem**. São Paulo: Cortez, 1995.

ANGIOLETTI, Joacir Leandro. **Protótipo de um software para educação primária usando recursos multimídia. 1995**. 69 f .Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

BELLO, José Luiz de Paiva. **Paulo Freire e uma nova filosofia para a educação**.[2000?]. Disponível em < http://www.pedagogia.click2site.com/pfreire.htm>. Acesso em: 14 mar 2001.

BIEBER, M ;KACMAR, C. **Designing hypertext support for computation applications**. Communications of the ACM. Volume 38, Number 8, August 1995. Disponível em < http://www.informatik.uni-trier.de/~ley/db/journals/cacm/index.html>

BIZZOTTO, Carlos Eduardo Negrão Bizzotto. **Concepção, desenvolvimento e validação de um ambiente extensível para o aprendizado distribuído**. Trabalho submetido ao Exame de Qualificação ao Doutorado.Dezembro,1999.

BIZZOTTO, Carlos Eduardo Negrão. **Director 8** – rápido e fácil. São Paulo: Makron Books,2000.

CAMPOS, Gilda H. B. de; ROCHA,Ana Regina C. **Manual para avaliação da qualidade do software educacional**. Rio de Janeiro: Publicações Técnicas, 1991.

CAMPOS, Gilda Helena Bernardino de. **Metodologia para avaliação da qualidade de software educacional**. Diretrizes para desenvolvedores e usuários. 1994. Dissertação de Doutorado. Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia – Universidade Federal do Rio de Janeiro, Rio de Janeiro.

CARNEIRO, L. M. F. et al. **ADVcharts**: a visual formalism for highly interactive systems. Cambridge: Cambridge University Press, 1994.

CERQUEIRA, Alessandro de Almeida Castro. Tese submetida para o grau de mestre em Ciência em Engenharia de Sistema e Computação. Rio de Janeiro,1997. Disponível em <http://www.nce.ufrj.br/~castro/tese/download.html >. Acesso em 03 abr. 2001.

CHAIBEN, Hamilton. **Hipermídia na educação**. Universidade Federal do Paraná. [2000?]. Disponível em< http://www.cce.ufpr.br/~hamilton/hed/hed00000.htm >. Acesso em 03 abr. 2001.

COLEMAN, D. et al. **Introducing object-charts or how to use statecharts in object-oriented design**. IEEE Transactions on Software Engineering, Janeiro, 1992.

CONKLIN, J. **Hypertext**: an introduction and survey. IEEE Computer. Setembro,1987.

DUPUY Jean-Pierre. **Nas origens das ciências cognitivas**.São Paulo: Ed Unesp, 1995. Disponível em< http://www.montreal.com.br/~casslear/autor/resenha.htm >. Acesso em 05 abr. 2001.

FEITOSA, Sonia Couto Souza. **Método Paulo Freire**: princípios e práticas de uma concepção popular de educação. Dissertação de Mestrado da Faculdade de Educação – Universidade São Paulo,1999. Disponível em < http://www.paulofreire.org/metodo.htm>. Acesso em 10 abr. 2001.

FIALHO, Francisco Pereira**. Sistemas de ensino a distância**. Apostila de curso. Universidade Federal de Santa Catarina, 1998.

HENDERSON, Chuck. **Mastering macromedia director 6**. California : Macromedia, 1997.

IZAKOWITZ, T. et al. **RMM**: a methodology for structured hypermedia design. Communications of the ACM. Outubro, 1995. Disponível em < http://www.informatik.uni-trier.de/~ley/db/journals/cacm/index.html>

LOLLINI, Paolo. **Didática e computador**. São Paulo: Loyola, 1991.

LUCENA, Carlos; FULKS, Hugo. **Professores e aprendizes na Web**: a educação na era da internet. Rio de Janeiro: Ed. Clube do Futuro, 2000.

MACROMEDIA. **Macromedia director 6**: using director.São Francisco:Macromedia, 1997.

MARTIN, James. **Análise e projeto orientados a objetos**. São Paulo: Makron Books, 1996.

MATUI, Jiron. **Construtivismo**: teoria construtivista sócio-histórica aplicada ao ensino. São Paulo: Ed. Moderna, 1996.

MIELKE, Fernando Luiz. **Ensino assistido por computador**: algumas considerações teóricas da ergonomia e da inteligência artificial num ambiente hipertexto. 1991. 112 f..Universidade Regional de Blumenau, Blumenau.

MIZUKAMI, M. da Graça Nicoletti. **Ensino**: as abordagens do processo. SãoPaulo: EPU, 1986. Disponível em < http://www.ufv.br/dpe/edu660/teorias.htm>. Acesso em 02 abr. 2001.

MODESTI, João. **Os fundamentos psicológicos da educação behaviorística**. Rio de Janeiro: Ed. Vozes, 1959.

PARKES, A. P. **A study of problem solving activities in a hypermedia representation.** Journal of Educational Multimedia and Hypermedia, Vol 3, N. 2, 1994.

PINHEIRO, Marco Antonio. **OOHDM**. 1999. Trabalho de Conclusão de Curso – Universidade do Vale do Itajaí, Itajaí.

PROINFO. **Programa nacional de informática na educação**. [1999?]. Disponível em <http://www.proinfo.gov.br/>. Acesso em 18 abr. de 2001.

RAMOS, Edla Maria Faust. **Análise ergonômica do sistema hipernet buscando o aprendizado da cooperação e da autonomia**.Florianópolis, 1996.

REYNOLDS, Angus; IWINSKI, Thomas. **Multimedia Training**. New York: McGraw-Hill, 1996.

RICH, John Martin. **Bases humanísticas da educação**. Rio de Janeiro: Zahar editores, 1975.

ROSENZWEIG, Gary. **Special edition using macromedia director 8**. Indianápolis- Indiana: QUE, 2000.

ROSSI, Gustavo. **Projeto de aplicações hipermídia**. [1996]. Disponível em <http://sol.info.unlp.edu.ar/~fer/oohdm/pg001.htm> . Acesso em 02 abr 2001.

RUMBAUGH, James et al. **Modelagem e projetos baseados em objetos**. Rio de Janeiro: Ed. Campus, 1994.

SALGADO, Ana Carolina et al. **Sistemas hipermídia**: hipertexto e banco de dados. Porto Alegre: Instituto de Informática da UFRGS, 1992.

SCHWABE, Daniel et al. **Abstraction, composition and lay-out definition mechanisms in OOHDM**.[1995?]. Disponível em <http://www.cs.tufts.edu/~isabel/schwabe/MainPage.html> Acesso em 03 abr. 2001.

SCHWABE, Daniel et al. **Systematic hypermedia application design with oohdm**. [1996?]. Disponível em < http://www.cs.unc.edu/~barman/HT96/P52/section1.html>. Acesso em 03 abr. 2001.

SCHWABE, Daniel et al. **The object-oriented hypermedia design model**: OOHDM. [1999?]. Disponível em < http://www.telemedia.puc-rio.Br/oohdm/site_oohdm/oohdm.html>. Acesso em 03 abr. 2001.

SMALL, Peter. **Lingo sorcery**: the magic of lists, objects and intelligent agents. London: John Wiley and sons, 1996.

SMITH, J. B.; WEISS, S. F. **Hypertext:** introduction to the special issue. Communications of the ACM, Vol 31, N.7. Julho, 1988.
Disponível em < http://www.informatik.uni-trier.de/~ley/db/journals/cacm/index.html>

TAJRA, Sanmya Feitosa. **Informática na educação**: professor na atualidade. São Paulo: Ed. Érica, 1998.

TAYLLER, Yves de la et al.**Piaget, Vygotsky e Wallon**: teorias psicogenéticas em discussão.São Paulo: Ed. Summus, 1993.

UNIVERSAL. In: **DICIONÁRIO** da língua portuguesa. Brasil. Priberam Informática, 1999. Disponível em < http://www.priberam.pt/DLPO> . Acesso em 23 abr. 2001.

VAANANEN, K. **Metaphor-based user interfaces for hyperspaces**. Workshop in methodological issues on the design of hypertext-based user interfaces. Darmstadt, Germany. Julho, 1993.

VALENTE, José Armando. **Computadores e conhecimento**: repensando a educação. Campinas: Gráfica Central da UNICAMP, 1993.

VARELA, Francisco et al. **The embodied mind**. Cabridge: MIT. 1997.

WINBLAD, Ann L. et al. **Software orientado ao objeto**. São Paulo: Makron Books, 1993.

WINN, William. **A conceptual basis for education applications of virtual reality**. Seatle: University of Washinghton, 1993.

WINN, William. **The impact of the three-dimensional immersive virtual environments on modern pedagogy.** Seattle: University of Washinghton, 1997.