

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

**PROTÓTIPO DE UM SOFTWARE EDUCACIONAL QUE
SIMULA UM SUPERMERCADO**

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

DANIELA CARDOZO

BLUMENAU, DEZEMBRO/2001.

2001/2-14

PROTÓTIPO DE UM SOFTWARE EDUCACIONAL QUE SIMULA UM SUPERMERCADO

DANIELA CARDOZO

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. Luiz Bianchi

Prof. Wilson Pedro Carli

DEDICATÓRIA

*A vocês que sempre compartilharam dos
meus anseios e desalentos,
vitórias e derrotas, alegrias e tristezas,
incentivando-me a sonhar e lutar pelos meus sonhos,
torcendo pelo meu sucesso,
minha mais profunda admiração e gratidão.
Obrigada pai! Obrigada mãe!*

AGRADECIMENTOS

Agradeço especialmente ao meu orientador Carlos Eduardo Negrão Bizzotto, que me ensinou que é sempre tempo de aprender e nunca é tempo de desistir, que acreditou na minha capacidade e me ajudou, com paciência e competência, a concretizar este trabalho.

Aos demais professores, que orientaram e dividiram com todos seus saberes e experiências, minha gratidão e carinho.

Agradeço à Andréia Somensi e a José Guidini, pela colaboração e a ajuda prestada nos momentos que precisei para a elaboração deste trabalho.

Aos colegas e amigos que neste período de convivência me animaram e ajudaram a vencer os obstáculos, trilhando pelos caminhos com coragem e perseverança, meu eterno amor.

Enfim, a todos, que de uma maneira ou de outra fizeram e fazem parte da minha vida, dedico este trabalho de conclusão de curso.

SUMÁRIO

DEDICATÓRIA.....	iii
AGRADECIMENTOS	iv
SUMÁRIO.....	v
LISTA DE FIGURAS	vii
LISTA DE QUADROS	ix
RESUMO	x
ABSTRACT	xi
1 INTRODUÇÃO.....	1
1.1 OBJETIVOS DO TRABALHO	2
1.2 ESTRUTURA DO TRABALHO	3
2 INFORMÁTICA NA EDUCAÇÃO.....	4
2.1 MULTIMÍDIA E HIPERMÍDIA	4
2.2 O USO DO COMPUTADOR NA SALA DE AULA.....	5
2.3 SOFTWARE EDUCACIONAL.....	6
2.3.1 TIPOS DE SOFTWARES	8
3 PEDAGOGIA ADOTADA	16
3.1 HISTÓRICO.....	17
3.2 TÉCNICAS DO ENSINO PIAGET	18
3.3 COMO A CRIANÇA APRENDE, SEGUNDO PIAGET	19
4 METODOLOGIA.....	21
4.1 METODOLOGIA DE ORIENTAÇÃO A OBJETOS	21
4.2 OOHDM - OBJECT ORIENTED HIPERMEDIA DESIGN METHOD.	23
4.2.1 PROJETO CONCEITUAL.	24

4.2.2 PROJETO NAVEGACIONAL.....	24
4.2.3 PROJETO DE INTERFACE ABSTRATA.....	26
4.2.4 IMPLEMENTAÇÃO.	28
4.3 AMBIENTE DE AUTORIA.	29
4.3.1 MACROMEDIA DIRECTOR	30
4.3.1.1 LINGUAGEM LINGO	31
5 DESENVOLVIMENTO DO PROTÓTIPO.....	33
5.1 ESPECIFICAÇÃO DO PROTÓTIPO	33
5.2 IMPLEMENTAÇÃO	40
5.2.1 OPERACIONALIDADE DA IMPLEMENTAÇÃO.....	40
6 CONCLUSÃO.....	48
6.1 SUGESTÕES PARA TRABALHOS FUTUROS.	49
REFERÊNCIAS BIBLIOGRÁFICAS	50

LISTA DE FIGURAS

Figura 1 - Interação Professor-Aluno Mediada pelo Computador.	6
Figura 2 - Tela principal do curso de Informática e Sociedade.....	9
Figura 3 - Site sobre questões referentes a qualquer disciplina.....	9
Figura 4 - Tela principal da Enciclopedia Microsoft Encarta.	10
Figura 5 - Tela do jogo Paciência.	11
Figura 6 - Tela do Microsoft Word.	11
Figura 7 - Tela principal do Macromedia Director	13
Figura 8 - Tela principal do Microsoft Power Point.....	13
Figura 9 - Tela principal do Borland Delphi 4.	14
Figura 10 -Tela principal do Supermercado	15
Figura 11 -Anatomia de um objeto.....	22
Figura 12 - Esquema Conceitual.	24
Figura 13 - Classe Navegacional.	25
Figura 14 - Esquema Navegacional.....	26
Figura 15 - Diagrama de Configuração.	28
Figura 16 - Tela Principal do Director.....	30
Figura 17 - Diagrama de Caso de Uso.....	34
Figura 18 - Modelo Conceitual.....	36
Figura 19 - Modelo Navegacional.	38
Figura 20 - Diagrama de Configuração.	39
Figura 21 - Tela de abertura do Protótipo.....	40
Figura 22 - Tela de boas vindas.....	41

Figura 23 - Tela das informações do usuário.	41
Figura 24 - Tela para selecionar a Lista de Compra.....	42
Figura 25 - Tela para criação de novas Listas.	43
Figura 26 - Tela interna do supermercado.....	43
Figura 27 - Tela das compras efetuadas.	44
Figura 28 - Tela de parabéns.	45
Figura 29 - Tela de alerta de falta de produtos.	45
Figura 30 - Tela de alerta quanto ao valor do dinheiro.	46
Figura 31 - Tela de alerta ao somatório dos valores das compras.....	46

LISTA DE QUADROS

Quadro 01- Comparação entre os princípios básicos das diferentes teorias de aprendizagem	16
Quadro 02 - Metodologia OOADM.....	23
Quadro 03 - Termos Lingo.	32
Quadro 04 - Atores do sistema.	35
Quadro 05 - Casos de uso do sistema.	35
Quadro 06 - Código para seleccionar listas.....	42
Quadro 07 – Código referente ao movimento das teclas.....	44
Quadro 08 – Código referente aos resultados.....	47

RESUMO

O presente trabalho de conclusão de curso tem como objetivo o desenvolvimento e implementação de um protótipo de software educacional para a 4ª série do 1º grau, que irá simular um supermercado. Neste software o aprendizado ocorrerá de forma interdisciplinar permitindo o aprendizado de diversos conteúdos. Será permitido que o professor interaja com o software, através de criação de listas de compras, inclusão e exclusão de produtos. Para o desenvolvimento do referido trabalho será utilizado o software de autoria Macromedia Director, versão 8.0, através do paradigma da orientação a objetos.

ABSTRACT

The current essay has the goal of developing and applying a prototype of educational software for 4th grade elementary school, that will simulate a supermarket. In this software the learning will occur in a interdisciplinary way allowing the learning of several contents. It will permit the teacher to interact with the software through the creation of a grocery list, as well as the including and excluding of products. For the development of such work it will be used the software which belongs to Macromedia Director, version 8.0, through the orientation to objects paradigms.

1 INTRODUÇÃO

As últimas décadas vêm se caracterizando por um exponencial crescimento da utilização da informática em todos os setores de nossa sociedade. As indústrias, comércios e profissionais liberais, independente da área de aplicação, estão fazendo uso cada vez mais intenso da informática para melhorar a qualidade, a produtividade e a confiabilidade de suas atividades. Adicionalmente, a utilização dos computadores têm provocado uma grande mudança na educação e nas relações entre pais, alunos e professores (Informática Exame, 1995).

Além de outras ferramentas e tecnologias, os professores poderão se utilizar da tecnologia multimídia para auxiliar na transmissão do conhecimento. Conforme Bizzotto (2000), o termo multimídia tem sido utilizado para designar softwares que utilizam, em conjunto, diversas mídias, como som, texto, imagem e vídeo. A característica distintiva de um software multimídia é a interatividade, através da qual o usuário deixa de ser um receptor passivo, passando a interagir com o software.

Segundo Vaughan (1994), “a multimídia interativa torna-se hipermídia quando seu desenvolvedor fornece uma estrutura de elementos vinculados pela qual um usuário pode mover-se e interagir”.

De acordo com Bizzotto (2000), em um software hipermídia pode-se interligar diferentes tipos de mídias. Com isso, o mercado tem sido inundado com uma grande quantidade de softwares hipermídia para as mais diversas aplicações: relatórios e apresentações empresariais, softwares educacionais, enciclopédias, etc.

A hipermídia, os programas que mesclam jogos e informações educativas, as enciclopédias e outras obras de referência em mídia digital decretam o início de uma nova era da comunicação. Todos estes recursos facilitam a interdisciplinaridade onde o professor atua como “orientador”, estimulando seus alunos a navegar pelo conhecimento e a fazer suas próprias descobertas.

Torna-se importante ressaltar que a informática não deve ser utilizada apenas para automatizar a escola ou a sala de aula, mas sim, contribuir para despertar nos alunos a vontade de aprender sempre. Segundo Angiolletti (1995), um software educacional de qualidade fundamenta-se em dois conceitos: aprender e divertir, transformando a velha e monótona forma de ensinar em um aprendizado estimulante.

Uma das maneiras de se utilizar o computador no processo educacional é torna-lo uma ferramenta didática à disposição do professor. O computador deve ser utilizado para auxiliar o

aprendizado do aluno em todas as áreas. Para esse auxílio, as escolas devem utilizar softwares educacionais (Bianchi, 1997).

Para que um software educacional seja uma ferramenta didática efetiva, é importante que ele permita a participação ativa dos alunos.

Esta participação ativa (construção) está em consonância com o Construtivismo, a teoria do psicólogo Jean Piaget, que descobriu a possibilidade de trabalhar a partir do erro dos alunos. Estudando a lógica dos erros, que os alunos cometiam durante testes de raciocínio, Piaget viu a possibilidade de resgatar o percurso, o processo de desenvolvimento intelectual do ser humano (Coll, 1996).

Assim, o Construtivismo defende que o conhecimento se forma e evolui através de um processo de construção e reconstrução do pensamento. Isso quer dizer, que o professor que ensina um aluno, mostrando a ele diretamente a resposta correta, tira dele o direito de aprender e de compreender as coisas.

Dentro deste contexto é que o presente trabalho propõe o desenvolvimento de um protótipo de software que simulará um supermercado virtual, onde o professor poderá criar tarefas a serem realizadas pelos alunos de acordo com a matéria em questão. O aluno poderá “passear” pelos vários setores do supermercado, onde conhecerá novos produtos, fazendo assim com que a tarefa estabelecida pelo professor seja cumprida, além de que terá total interação com o software. O protótipo proposto será desenvolvido no software Macromedia Director, versão 8.0, sendo utilizada a sua própria linguagem de programação Lingo.

1.1 OBJETIVOS DO TRABALHO

O objetivo principal deste trabalho é o desenvolvimento de um software educacional que simula um supermercado.

Para ser atingido o objetivo geral, alguns objetivos específicos devem ser atendidos:

- a) permitir que o aluno efetue suas compras “passeando” pelo mercado;
- b) possibilitar ao aluno a livre escolha de produtos de diferentes marcas e tipos, desde que cumpra as tarefas exigidas pelo professor;
- c) fazer com que o aluno aprenda as quatro operações matemáticas básicas de uma forma divertida, ampliando assim seu raciocínio numérico;
- d) avaliar se o aluno cumpriu a tarefa estabelecida pelo professor;
- e) possibilitar ao professor o cadastro de novos produtos, bem como sua exclusão;

1.2 ESTRUTURA DO TRABALHO

O trabalho está dividido em seis capítulos. O primeiro capítulo aborda os objetivos do trabalho e sua estrutura.

O segundo capítulo trata da informática na educação, e os tipos de softwares. Sendo que o maior enfoque será do uso do computador no ensino primário.

O terceiro capítulo aborda a Pedagogia de Ensino Piaget, que descobriu a possibilidade de trabalhar a partir do erro, sendo que todo o conhecimento é construído através de um processo contínuo de fazer e refazer. Ele viu a possibilidade de resgatar o percurso, o processo de desenvolvimento intelectual dos seres humanos.

O quarto capítulo apresenta a metodologia utilizada no trabalho, neste caso a metodologia orientada a objetos OOHDM, e também aborda o software Macromedia Director e sua linguagem Lingo.

O quinto capítulo apresenta o desenvolvimento e a implementação do software proposto.

O sexto capítulo cita conclusões referente ao trabalho, e sugestões para futuros trabalhos que poderão ser realizados.

2 INFORMÁTICA NA EDUCAÇÃO

A utilização da informática vem crescendo de forma bastante acelerada no mercado mundial. Muitos países necessitaram desenvolver seus próprios softwares educacionais para atenderem as suas necessidades e especificidades. No Brasil também está ocorrendo o mesmo, onde estão sendo desenvolvidos softwares para os mais diversos conteúdos programáticos (Campos, 1991).

O computador é uma grande ferramenta para auxiliar os professores no processo de ensino – aprendizagem, não só por ser muito atrativo e divertido, mas, principalmente, por possibilitar situações difíceis de serem reproduzidas na sala de aula tradicional (Somensi, 1997).

Para compreender adequadamente as potencialidades e as limitações do uso do computador na sala de aula, torna-se importante conhecer as principais características dos tipos de softwares educacionais existentes no mercado brasileiro, além da tecnologia que eles utilizam: a hipermídia.

2.1 MULTIMÍDIA E HIPERMÍDIA

O termo multimídia vem sendo utilizado para designar softwares que utilizam, em conjunto, diferentes tipos de mídias, como som, imagem, vídeo, etc., de forma interativa. A característica que distingue um software multimídia de outras tecnologias que utilizam múltiplas mídias, como a televisão e o cinema, é a interatividade. A partir da possibilidade de interação, deixa de existir o receptor passivo e surge o participante ativo.

Inicialmente, esta interação ocorria através de *links* em textos (hipertextos), permitindo que o usuário “navegasse” no software em busca de informações mais detalhadas sobre um dado item. Com a evolução da tecnologia, é possível criar *links* não só em textos, mas também em outros tipos de mídias como imagens e vídeos. Com isso surge a hipermídia, possibilitando inúmeras combinações de mídias e *links* que podem tornar a utilização destes softwares mais agradável e produtiva (Bizzotto, 2000).

Estas características despertaram grande interesse no uso educacional destes tipos de softwares, uma vez que os títulos (softwares) que podem ser desenvolvidos permitem a criação de situações que se aproximam muito da realidade vivenciada pelo aluno,

principalmente com a incorporação de ambientes de três dimensões (3D) e de realidade virtual. Neste sentido, os softwares hipermídia, oferecem muito mais atrativos que as publicações em papel, permitindo que o usuário absorva melhor a mensagem transmitida (Cabral, 1994). Este é um dos motivos da grande utilização dos softwares hipermídia como ferramenta auxiliar no processo de ensino-aprendizagem.

A utilização de ferramentas informatizadas (hipertexto, multimídia, hipermídia) na sala de aula é genericamente denominada de “utilização do computador na sala de aula”, numa referência à parte visível do sistema informatizado: o computador (o hardware).

2.2 O USO DO COMPUTADOR NA SALA DE AULA

As escolas sofrem muitas transformações oriundas de aspectos sociais e econômicos, mas principalmente através de inovações agregadas da indústria e do comércio, que devem ser assimiladas pelos alunos, a fim de torná-los mais eficazes quando estes entrarem no mercado de trabalho.

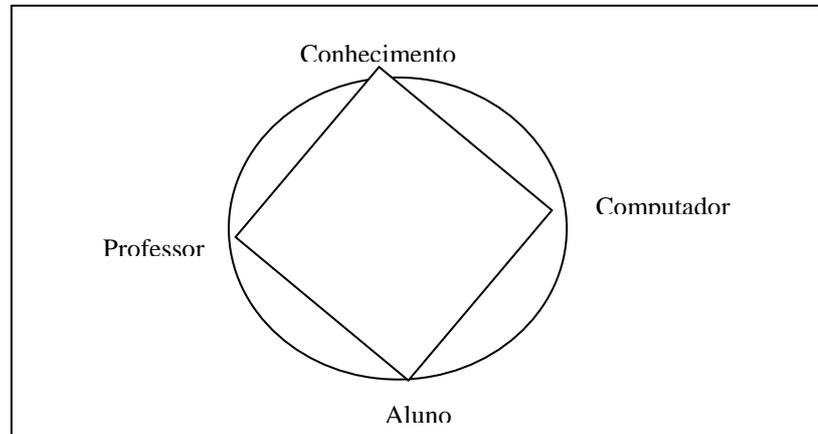
Neste sentido, um dos grandes dilemas que as escolas enfrentam é a identificação da melhor maneira de utilizar o computador na sala de aula. Para resolver este problema, conforme ressalta Angiolletti (1995), uma das melhores alternativas é tornar o computador uma ferramenta auxiliar no processo de ensino-aprendizagem, ou seja, mais uma ferramenta à disposição do professor.

A educação deve favorecer a transformação do contexto social. Os alunos e os professores são sujeitos ativos e a relação do homem com o mundo é feita através do trabalho coletivo. Portanto, um caminho que parece mais adequado é a utilização de softwares que permitam que o professor e o aluno construam seus próprios conhecimentos, desenvolvendo artefatos que estejam dentro de seus interesses. Com isso, o software passa a oportunizar situações que promovam o desenvolvimento e a organização do pensamento, bem como possibilitem despertar a curiosidade e o interesse, ingredientes essenciais ao aprendizado (Bizzotto, 1999).

No entanto torna-se importante ressaltar que este aprendizado não ocorre de forma isolada, sendo muito importante a interação entre os diferentes atores envolvidos no processo: alunos e professores. Desta forma, observa-se que o computador passa a ser um mediador do

processo de interação entre professores e alunos em direção ao conhecimento, conforme figura 1.

Figura 1 - Interação Professor-Aluno Mediada pelo Computador



Fonte: adaptado de Bizzotto (1999).

O círculo, presente na figura 1, indica que as relações entre os elementos do quadrilátero não devem ser entendidas como sendo estáticas, mas sim como uma rede de interações dinâmicas, cujo resultado é a transformação de cada um dos elementos do quadrilátero em função destas interações. A partir destas interações dinâmicas, o professor, o aluno e o computador (na verdade, o software) co-evoluem para um novo nível de conhecimento (Bizzotto, 1999).

Neste sentido, observa-se que o objetivo da utilização da informática na educação não é a substituição do professor pelo computador, uma vez que o professor possui um papel essencial na rede de interações descrita. Com isso, de acordo com Mielke (1991), cabe ao computador o papel de ferramenta auxiliar no processo de ensino-aprendizagem, ficando a cargo do professor e dos alunos a definição de quando e como utilizar o computador .

Esta decisão é decorrência direta da metodologia utilizada pelo professor e do conteúdo a ser trabalhado. Assim, o que diferencia um sistema do outro não é o computador em si, mas sim o software educacional utilizado / disponível.

2.3 SOFTWARE EDUCACIONAL

Quando se fala em software educacional, entende-se programas de computador que possuem uma proposta de ensino-aprendizagem explícita em sua documentação. Este tipo de

software tem despertado questões relacionadas à sua qualidade, ou seja, sua efetividade como ferramenta educacional.

A qualidade do software educacional é de difícil mensuração, uma vez que deve envolver tanto critérios técnicos quanto critérios relacionados ao apoio oferecido na aquisição de habilidades mentais básicas e na aquisição do conhecimento (Campos, 1991). De acordo com Campos (1991), existem três argumentos que são pontos de partida para o desenvolvimento de um software educacional, a saber:

1. A construção de sistemas informatizados dedicados ao ensino necessitam de conteúdos adaptados ao aluno;
2. A gestão da interação entre o sistema e o aluno, isto é, considerar a concepção de interfaces;
3. As tecnologias permitem a construção de um sistema de coleta de dados sobre a aprendizagem e suas condições de realização.

Em termos históricos, esta qualidade vem evoluindo na direção de sistemas que permitem maior interação por parte do professor e do aluno. De acordo com Winn (1993), pode-se agrupar os softwares educacionais em diferentes gerações:

- 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 a forma de apresentação da informação ao estudante. Ocorre um impacto positivo 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 afirma 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.

Independentemente da geração à qual pertença um dado software, torna-se importante identificar as suas características principais, ou seja, aquelas características que diferencia um software dos outros. Cada software educacional contém suas próprias características. Dentro deste contexto, é que Winn (1993), classifica as características em três dimensões, conforme abaixo descritas:

- 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, podem 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 apresentado. 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.

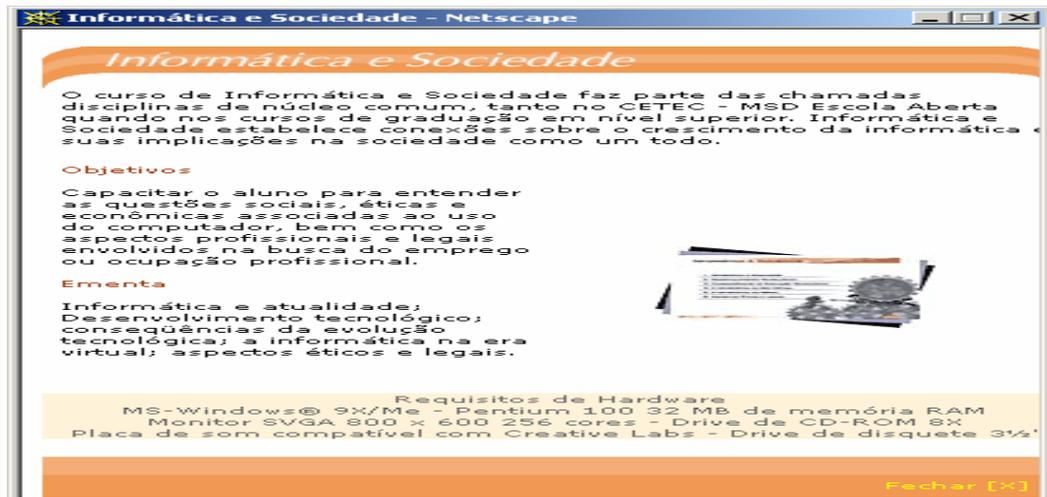
2.3.1 TIPOS DE SOFTWARES

Os softwares educacionais possuem como característica comum o objetivo de auxiliar no aprendizado de um dado conteúdo. No entanto, a abordagem utilizada varia muito entre os diversos tipos de softwares. Em termos gerais, é possível agrupar os softwares educacionais em nove grandes grupos (Tajra, 1998):

- 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. Como exemplo, podemos citar o software Informática e Sociedade do curso MSD Tecnologia Educacional, mostrado na figura 2.

Figura 2 – Tela principal do Curso de Informática e Sociedade



Fonte: Msd Tecnologia Educacional, disponível em <http://www.msd.com.br>

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. Dentro deste contexto, pode-se citar o site <http://www.10emtudo.com.br> onde existe um ícone na qual possibilita que o usuário efetue questões sobre diversas matérias, e no final obtenha o resultado, onde dirá se ele acertou ou errou as perguntas. Parte deste site é demonstrado na figura 3.

Figura 3 – Site sobre questões referente a qualquer disciplina

Fonte: Adaptado do site 10emtudo, disponível em <http://www.10emtudo.com.br>

- c) investigação: neste grupo, encontram-se enciclopédias, em que se pode localizar várias informações a respeito de diversos assuntos. 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. Deste tipo de software pode-se citar o software Enciclopédia Microsoft Encarta demonstrado na figura 4.

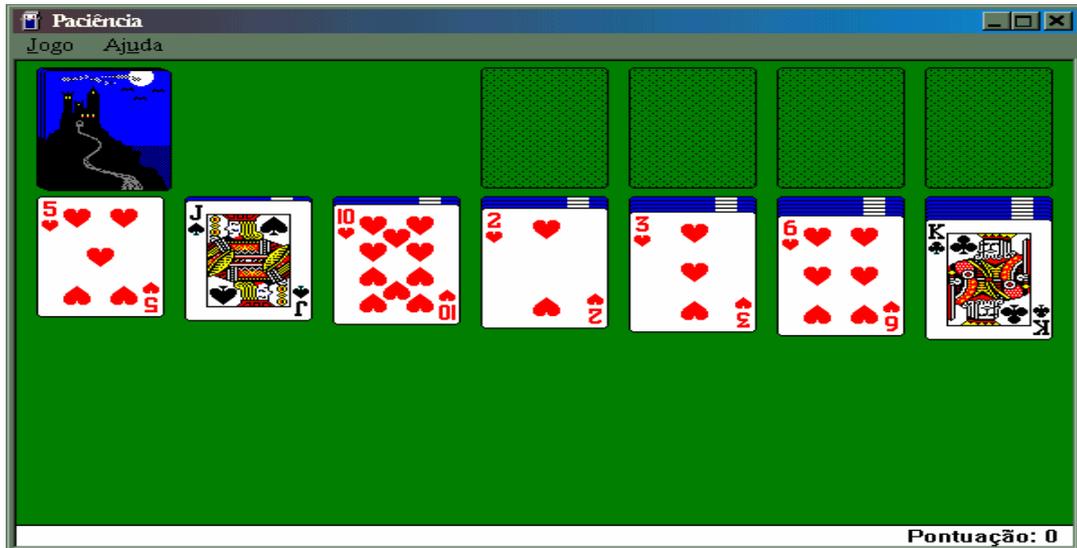
Figura 4 – Tela principal do Enciclopédia Microsoft Encarta



Fonte: Adaptado do software Enciclopédia Microsoft Encarta

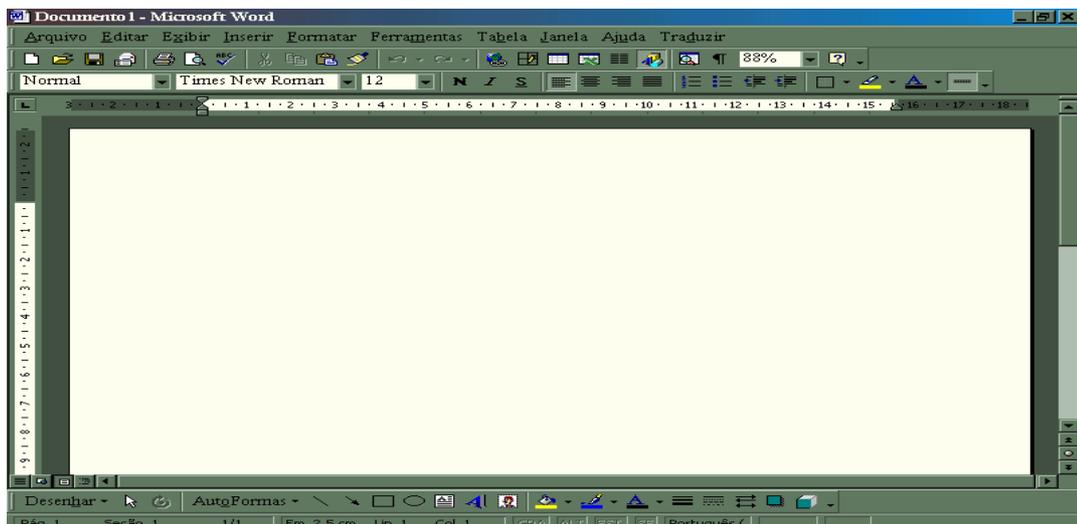
- d) jogos: são os softwares de entretenimento, a sua maior indicação são o lazer e a diversão. Com certeza, os jogos apresentam grande interatividade e recursos de programação muito sofisticados. Os jogos sofrem preconceito na área educacional, pois é comum ouvirmos professores informando aos pais que 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. Como exemplo pode-se citar o software Paciência, abaixo na figura 5, onde através dele o usuário poderá conhecer as cartas do baralho e seus respectivos naipes.

Figura 5 – Tela do jogo Paciência



- e) 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 o Microsoft Word, mostrado na figura 6.

Figura 6 – Tela principal do Microsoft Word

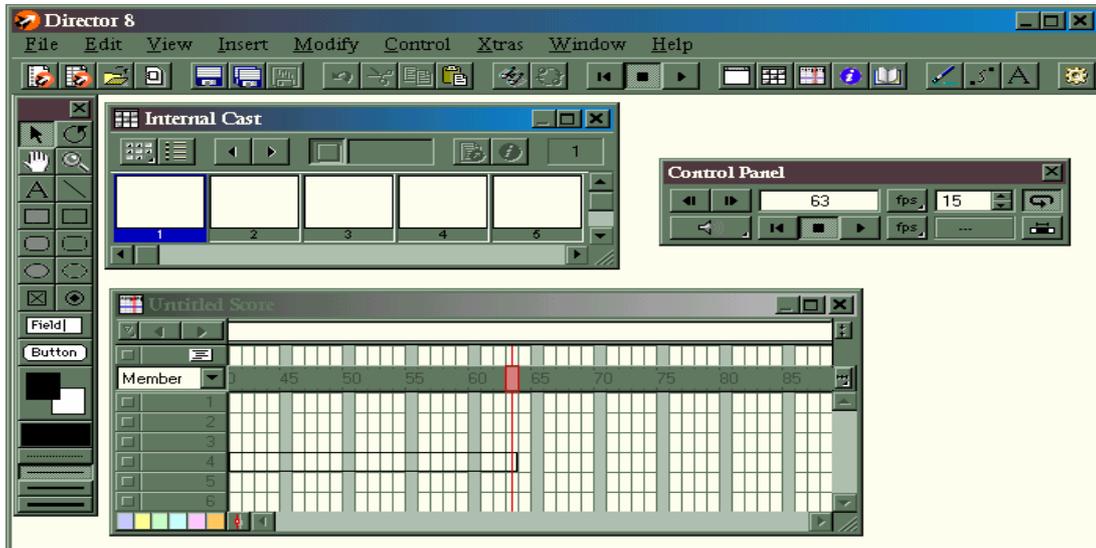


- editores de textos: são softwares que apresentam vários recursos de elaboração de textos, tornando mais fácil e rica a produção de trabalhos, visto que por meio deles é possível incluir diversos tipos de fontes, estilos, bordas, figuras, margens, parágrafos. Com ele pode-se elaborar diversas

atividades de criações de relatórios, cartas, poesias, músicas, entrevistas e, inclusive, produzir jornais;

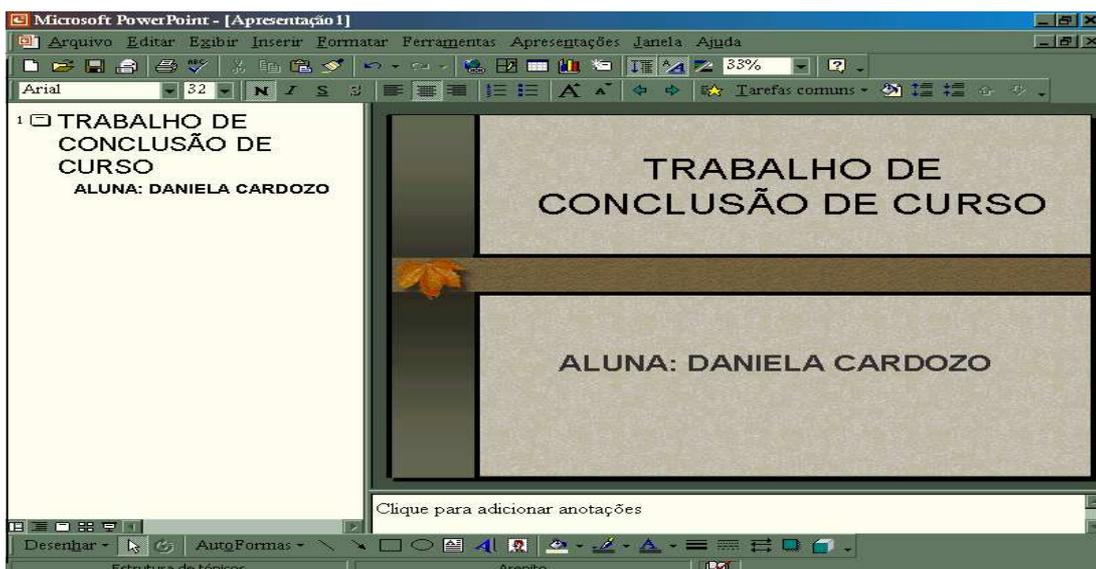
- banco de dados: possibilitam o arquivamento de informações que poderão posteriormente ser relacionadas para diversos tipos de análises e ordenações conforme o interesse do usuário.
 - planilhas eletrônicas: possibilitam a realização de cálculos, de uma forma rápida, a partir de dados informados e posteriormente, a elaboração de gráficos que facilitam a visualização das informações;
 - softwares gráficos: são aqueles voltados para a elaboração de desenhos e produções artísticas. Por meio 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 seus próprios desenhos de acordo com sua imaginação e criatividade.
- f) 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 sem grandes complicações. Estes softwares funcionam como um aglutinador de produções elaboradas em outros softwares. A grande vantagem dos softwares de autoria, 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. Dentre os softwares de autoria existentes, pode-se citar o software Macromedia Director 8, mostrado na figura 7.

Figura 7 – Tela principal do Macromedia Director



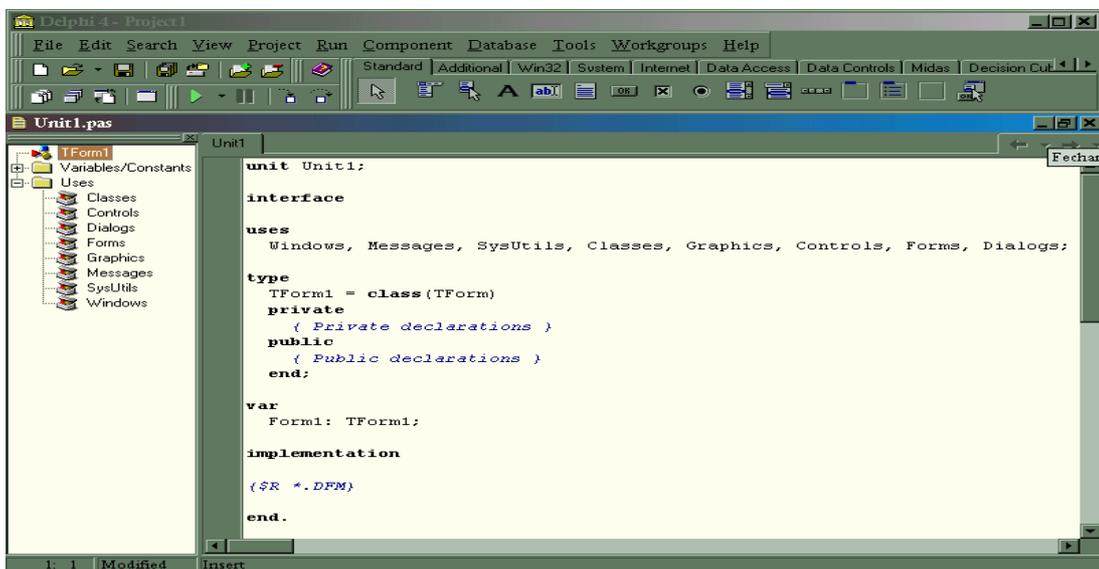
- g) softwares de apresentação: estes programas são utilizados para elaborar apresentações de palestras e aulas. Possuem recursos de visualização de telas, bem como, permitem produções de slides e transparências. Estes softwares, também, são muito bem aceitos pelos alunos, pois estes poderão elaborar seus trabalhos em forma de apresentação diferentemente de textos impressos. Dentro deste contexto, pode-se citar o software Power Point, citado na figura 8.

Figura 8 – Tela principal do Microsoft Power Point



- h) softwares de programação: são os que permitem a criação de outros programas. Estes programas são ótimos para estimular o 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. Pode-se citar como exemplo o software Borland Delphi 4, mostrado na figura 9.

Figura 9 - Tela principal do Delphi 4



- i) 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. Estes softwares exigem uma maior habilidade por parte dos professores para que eles possam analisar os possíveis acontecimentos havidos. Os softwares simulados são recursos significativos para o aprendizado e atrativos para os alunos e os professores. Sendo assim, pode-se citar o software Supermercado – Usando a Matemática na Prática, mostrado na figura 10.

Figura 10 – Tela principal do Supermercado



No mercado atual existem muitos softwares educacionais, entre eles está o “Supermercado, Usando a matemática na Prática”, o qual foi desenvolvido pela escola Positivo de Curitiba. É um jogo educativo que tem por principal objetivo estimular a aprendizagem e desenvolver o raciocínio de crianças de 7 a 10 anos.

Este software além de ser educativo, prende muita a atenção das crianças, pelo fato de ele ser em três dimensões (3D), e “colocar” a criança dentro do supermercado. Dentro deste ambiente lúdico, a criança deverá executar várias tarefas, como por exemplo:

- a) interpretar e procurar o produto indicado na lista de compras;
- b) calcular quantidades;
- c) verificar o prazo de validade;
- d) comparar e conferir preços;

3 PEDAGOGIA ADOTADA

Conforme ressaltado anteriormente, tem havido um expressivo aumento no número e tipo de softwares educacionais disponíveis aos professores e alunos brasileiros. O que se observa, no entanto, é que a grande maioria destes softwares tem como objetivo a “automação da sala de aula tradicional”. Assim, estes softwares podem ser considerados como as versões eletrônicas dos livros didáticos.

Isto tem ocorrido porque, muitas vezes, não há uma reflexão sobre os princípios pedagógicos que fundamentam o desenvolvimento do software. Assim, não se analisa a pedagogia que dá suporte ao processo de ensino-aprendizagem dos alunos.

Independente da teoria pedagógica que irá fundamentar o desenvolvimento do software educacional é essencial que se defina qual das teorias existentes que será utilizada como base para o desenvolvimento.

De acordo com Bizzotto (1999), “... as teorias de aprendizagem diferem entre si em função dos princípios sobre os quais elas se fundamentam. Assim, é fundamental que estes princípios sejam conhecidos claramente, para que não se utilize uma abordagem que se baseie em princípios conflitantes com aqueles que serão utilizados como base para o desenvolvimento do software”. No quadro 01 são apresentados os princípios básicos de três abordagens muito conhecidas e utilizadas.

Quadro 01 - Comparação entre os Princípios básicos das diferentes Teorias de Aprendizagem.

Abordagem	Princípios
Tradicional	Heteronomia (sem autonomia) Reprodução Massificação
Behaviorista	Heteronomia (sem autonomia) Auto-didatismo Reprodução Massificação
Construtivista	Autonomia Interação Construção Personalização

Fonte: adaptado de Bizzotto (1999)

Com base no apresentado no quadro 01, observa-se que a abordagem construtivista é aquela cujos princípios se adequam melhor aos objetivos do protótipo proposto no presente trabalho. Dentre os representantes desta abordagem podemos destacar o trabalho de Jean Piaget.

Embora Piaget não tenha desenvolvido uma teoria de aprendizagem, sua teoria epistemológica de como, quando e por que 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 equilíbrio, como condições básicas para 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 interrelacionamento contínuo do organismo com o ambiente em que se encontra inserido, através de um processo de trocas de influências equivalentes*”.

Em função desta escolha, a teoria de Jean Piaget será apresentado em maiores detalhes no item a seguir.

3.1 HISTÓRICO

Jean Piaget, especialista em psicologia evolutiva e epistemologia genética, filósofo e educador, nasceu em Neuchâtel, na Suíça, em 09 de agosto de 1886 (Piaget, 1998). Desde criança interessou-se por mecânica, fósseis e zoologia. Enquanto terminava seus estudos secundários, trabalhou como assistente voluntário do Laboratório do Museu de História Natural, de Neuchâtel, sob a direção de Paul Godet, especialista em malacologia. Com a morte de Godet, em 1911, continuou trabalhando no laboratório e escreveu vários trabalhos, alguns publicados pelo Museu de História Natural de Genebra, na Lamarck Collection e na Revue Suisse de Zoologie (Piaget, 1998).

Registrando-se na divisão de ciência da Universidade de Neuchâtel, dela recebeu o título de Doutor em Ciências (1918), seguindo depois para Zurique, onde estudou nos laboratórios de psicologia de G.F.Lipps e estagiou na clínica psiquiátrica de E. Bleuler. Em 1919 ingressou na Sorbonne, onde estudou psicopatologia, e em 1923, assumiu a direção do Instituto Jean Jacques Rousseau, de Genebra, passando a estudar, sistematicamente, a inteligência (Piaget, 1998).

Jean Piaget foi o responsável por uma das maiores contribuições no campo da psicologia científica contemporânea, na área específica do comportamento cognitivo. As aplicações de sua teoria do desenvolvimento encontram-se muito difundidas, no campo pedagógico e na explicação da evolução da conduta cognitiva.

3.2 TÉCNICAS DO ENSINO PIAGET

Ser um professor piagetiano difere muito de ser um professor tradicional. O professor precisa aprender a brincar, a trabalhar o simbolismo (faz de conta), ter disposição para fazer exercícios físicos para ter agilidade nos trabalhos motores (Piaget, 1998).

Segundo Piaget (1998), dentro da pedagogia Piaget, há algumas técnicas relevantes para se tornar um professor piagetiano, nas quais estão abaixo descritas:

- a) ter prestígio entre os alunos, pois esses só obedecem a quem admiram (prestígio obtido pela capacidade intelectual);
- b) saber a matéria não como coisa vaga sobre a qual se pode fazer um discurso. O professor deve ser capaz de fazer sínteses dos temas expostos;
- c) graduar, rigorosamente, as dificuldades dos problemas propostos;
- d) as relações afetivas do professor-aluno tem profunda influência no trabalho escolar;
- e) o professor piagetiano não trabalha pelo aluno: o que o aluno pode fazer sozinho ou em grupos jamais deve ser feito pelo professor. O professor é um criador de "obstáculos" transponíveis, jamais um modelo para ser imitado;
- f) entusiasmar os alunos pelo trabalho que está sendo realizado. É fundamental saber dramatizar;
- g) elogiar tudo quando for elogiável. Isso sinaliza os acertos e estimula a continuidade da ação;
- h) "vender" ao aluno o que vai trabalhar. Ninguém se esforça num determinado sentido se achar que isso não vale a pena;
- i) o professor deve ter claro o objetivo que deseja atingir.

Piaget aconselha que se proponha à criança tarefas, progressivamente, mais difíceis (gradação e seriação) para que ela seja estimulada a reorganizar seu comportamento sensório-motor, verbal e mental, em nível superior (desenvolvimento da inteligência).

A criança é que determina como o professor deve apresentar as situações didáticas, pois, em cada estágio de desenvolvimento, a criança tem maneira diferente de aprender (esquemas de assimilação).

3.3 COMO A CRIANÇA APRENDE, SEGUNDO PIAGET

Segundo Lagoa (1994), para o pesquisador suíço Piaget, que descobriu a possibilidade de trabalhar a partir do erro, todo o conhecimento é construído através de um processo contínuo de fazer e refazer. Nesse processo, o conhecimento seguinte é sempre mais sofisticado que o anterior e depende dos estímulos do meio ambiente. Da sua teoria, conhecida pelo imponente nome de epistemologia genética, partiram as teses construtivistas atuais e as práticas escolares que respeitam o universo do aluno.

Trabalhar a partir do erro é a sua grande descoberta. Na lógica dos erros é que as crianças cometiam durante testes de raciocínio, ele viu a possibilidade de resgatar o percurso, o processo de desenvolvimento intelectual do ser humano.

No que diz respeito à educação e à pedagogia, a resposta encontrada por ele é aparentemente simples: o conhecimento se forma e evolui através de um processo de construção e reconstrução do pensamento. Piaget quer dizer que o professor que ensina a uma criança, mostrando a ela diretamente a resposta correta, tira dela o direito de aprender, de compreender as coisas.

A teoria de Piaget inspira, na prática, três princípios que podem ajudar o professor na sala de aula:

- a) respeito à produção do aluno: uma idéia consensual, mas que ainda é difícil de ser praticada, porque o professor precisa estar seguro de que qualquer indivíduo pode aprender, independentemente da sua idade, sexo, cor, classe social. Conhecer é inerente ao ser humano, porque disso depende a sua sobrevivência;
- b) o aluno precisa de espaço para testar suas hipóteses: o conhecer parte da ação. O ser humano adquire conhecimentos agindo sobre o objeto a conhecer, seja ele

pertencente à realidade concreta ou à realidade abstrata. A criança pensa que basta dar “petelecos” nos objetos para que eles se movam, porque deu um peteleco na bola de papel e ela se moveu. Bate igualmente na mesa de madeira e ela não se move. A criança aprende que alguns objetos se movem, outros não, mas para exercitar isso ela precisa de espaço para fazer perguntas e testar respostas, interpretar a realidade a si mesma;

- c) o trabalho em grupo facilita o aprendizado: o conhecer se dá na integração dos indivíduos com os objetos a conhecer e entre os indivíduos. Por isso, o trabalho cooperativo é mais fértil.

4 METODOLOGIA

Para o desenvolvimento do protótipo de software proposto foram utilizadas as seguintes tecnologias: metodologia orientada a objetos, neste caso a metodologia OOHDm e o ambiente de desenvolvimento Macromedia Director, através da linguagem de programação Lingo.

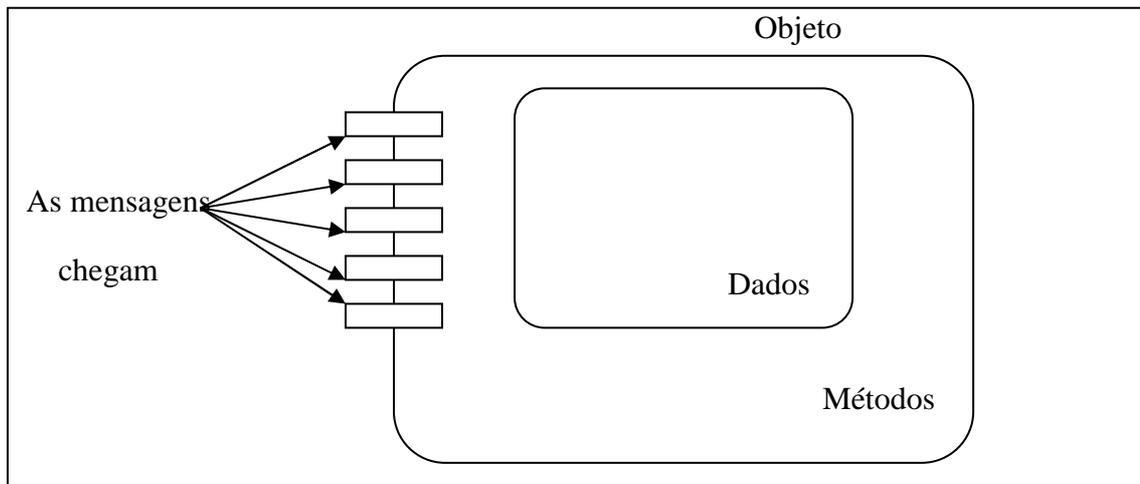
4.1 METODOLOGIA DE ORIENTAÇÃO A OBJETOS

Segundo Winn (1993), a programação orientada a objeto, acompanhada de suas ferramentas e arquiteturas, está mudando o universo da programação. As ferramentas de desenvolvimento, os ambientes dos sistemas e as aplicações para dar suporte a este tipo de informação multimídia, com participação direta do usuário final via processamento distribuído, são as tendências que indicam a orientação a objetos.

Dentro desta metodologia de orientação a objeto existem mecanismos básicos, ou seja, esses mecanismos são compostos de: objetos, mensagens, classes e hereditariedade. Abaixo serão descritos esses mecanismos separadamente (Winblad, 1993):

- a) objetos: são módulos que contém dados e instruções para operar sobre estes dados. Dentro destes objetos, residem os dados das linguagens tradicionais, como números, matrizes, strings e registros, bem como funções, instruções ou sub-rotinas que as operam. Os objetos são entidades que têm atributos específicos (dados) e maneiras de comportamento (procedures);
- b) mensagens: diferente dos dados passivos em sistemas tradicionais, os objetos possuem habilidade para agir. A ação ocorre quando um objeto recebe uma mensagem, isto é, uma solicitação para que se comporte de uma determinada maneira. Quando os programas orientados ao objeto são executados, os objetos estão recebendo, interpretando e respondendo mensagens de objetos (Winblad, 1993). A figura 11 ilustra melhor como funciona o mecanismo mensagem na metodologia orientado a objetos;

Figura 11 - Anatomia de um objeto



Fonte: Adaptado de Winblad (1993).

- c) métodos: segundo Martin (1996), os métodos especificam a maneira pela qual os dados de um objeto são manipulados;
- d) classes: é a descrição de um conjunto de objetos quase idênticos. Uma classe consiste em métodos e dados que resumem as características comuns de um conjunto de objetos. Definir classes significa posicionar codificação reutilizável em um depósito comum em vez de expressá-lo várias vezes (Winblad, 1993);
- e) hereditariedade: é o mecanismo de compartilhamento automático de métodos e dados entre classes, subclasses e objetos. Segundo Rumbaugh (1994), a hereditariedade permite aos programadores criar novas classes programando somente a diferença entre elas e a classe-pai;
- f) encapsulamento: é o 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 a objeto pode operar diretamente em um dado do objeto (Winblad, 1993);

4.2 OOHDM – OBJECT ORIENTED HIPERMEDIA DESIGN METHOD

O OOHDM utiliza mecanismos de abstração e composição numa arquitetura orientada a objetos, que permite uma descrição concisa de informações complexas e uma especificação de padrões de navegação e interface (Schwabe, 1992).

No OOHDM, uma aplicação é construída em um processo de quatro passos, para suportar um modelo incremental ou de protótipo. O quadro 02 mostra os quatro passos e as informações relevantes de cada um.

Quadro 02 – Metodologia OOHDM

Atividades	Produtos	Formalismo	Mecanismos
Projeto Conceitual	Classes, Sub-sistemas, relacionamentos, perspectivas de atributos	Modelagem orientada a objetos	Classificação, agregação, generalização e especialização
Projeto Navegacional	Nós, Links, estruturas de acesso, contexto navegacional	Views, State charts, classes de contexto	Classificação, agregação, generalização e especialização
Projeto de Interface Abstrata	Objetos de interface abstrata, respostas a eventos externos, transformações de interface	Abstract Data View (ADV), diagramas de configuração, ADV-charts	Mapeamento entre navegação e Objetos perceptíveis
Implementação	Aplicação		

Fonte: adaptado de Pompermaier (1999).

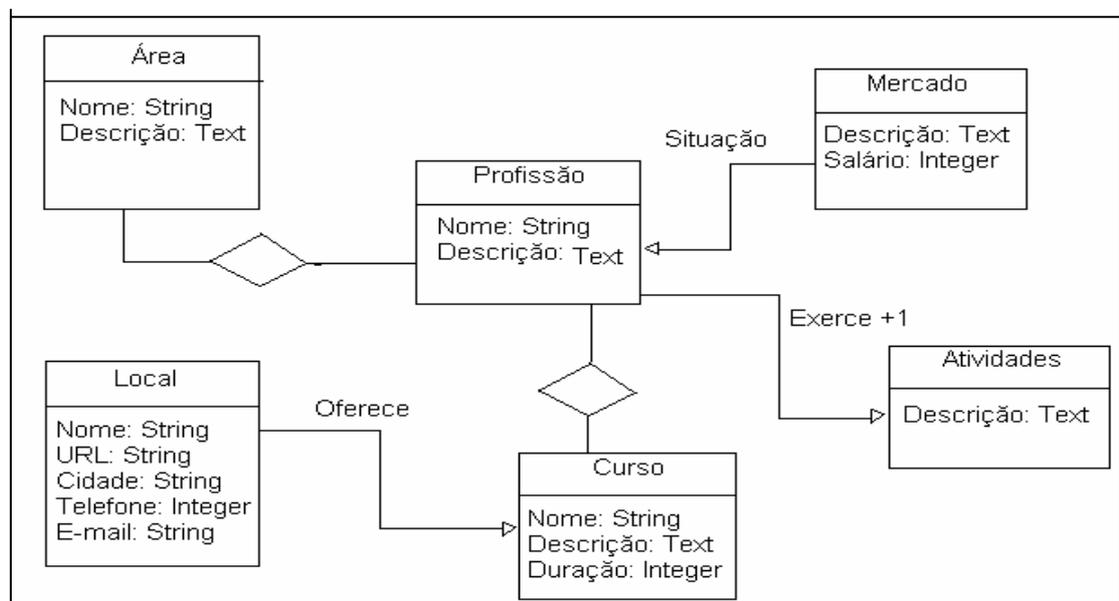
As seções seguintes abordarão cada um dos passos para a realização de uma modelagem de aplicação hipermídia utilizando OOHDM.

4.2.1 PROJETO CONCEITUAL

Em uma modelagem orientada a objetos procura-se representar os aspectos estruturais e comportamentais que serão mapeados para o ambiente de implementação como estruturas de classes com seus métodos (Rumbaugh, 1991).

Nesta fase o domínio da aplicação é criada utilizando os princípios de modelagem orientada a objetos apresentada em Rumbaugh (1991). Serão utilizados os conceitos de classes, relacionamentos, agregação, generalização e especialização. A figura 12 apresenta um exemplo de esquema conceitual.

Figura 12 - Esquema Conceitual



Fonte: adaptado de Pompermaier (1999).

4.2.2 PROJETO NAVEGACIONAL

As aplicações hipermídias são projetadas para realizar navegação através de um espaço de informações (Schwabe, 1992). A navegação é uma característica das aplicações hipermídia em que o usuário pode percorrer as informações que se apresentam na tela do computador. Esta é uma das principais características destes tipos de sistemas.

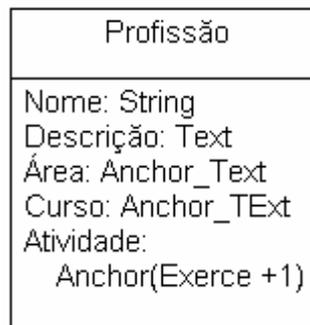
Diferentes modelos navegacionais podem ser construídos a partir de um projeto conceitual (Rossi, 1995). A estrutura navegacional de uma aplicação hipermídia é definida

através de especificações de classes de navegação que refletem as visões sobre o domínio da aplicação.

Em OOHDm este projeto é um conjunto de tipos pré-definidos de classes de navegação: nós, elos, estruturas de acesso, os quais são organizados em Contextos de Navegação (Schwabe, 1992).

As classes de navegação ou nós são definidas por um conjunto de atributos provenientes das classes conceituais, como visões orientadas a objetos sobre estas classes. Os nós são uns recipientes de informações básicas. A figura 13 mostra um exemplo de classe de navegação.

Figura 13 - Classe Navegacional

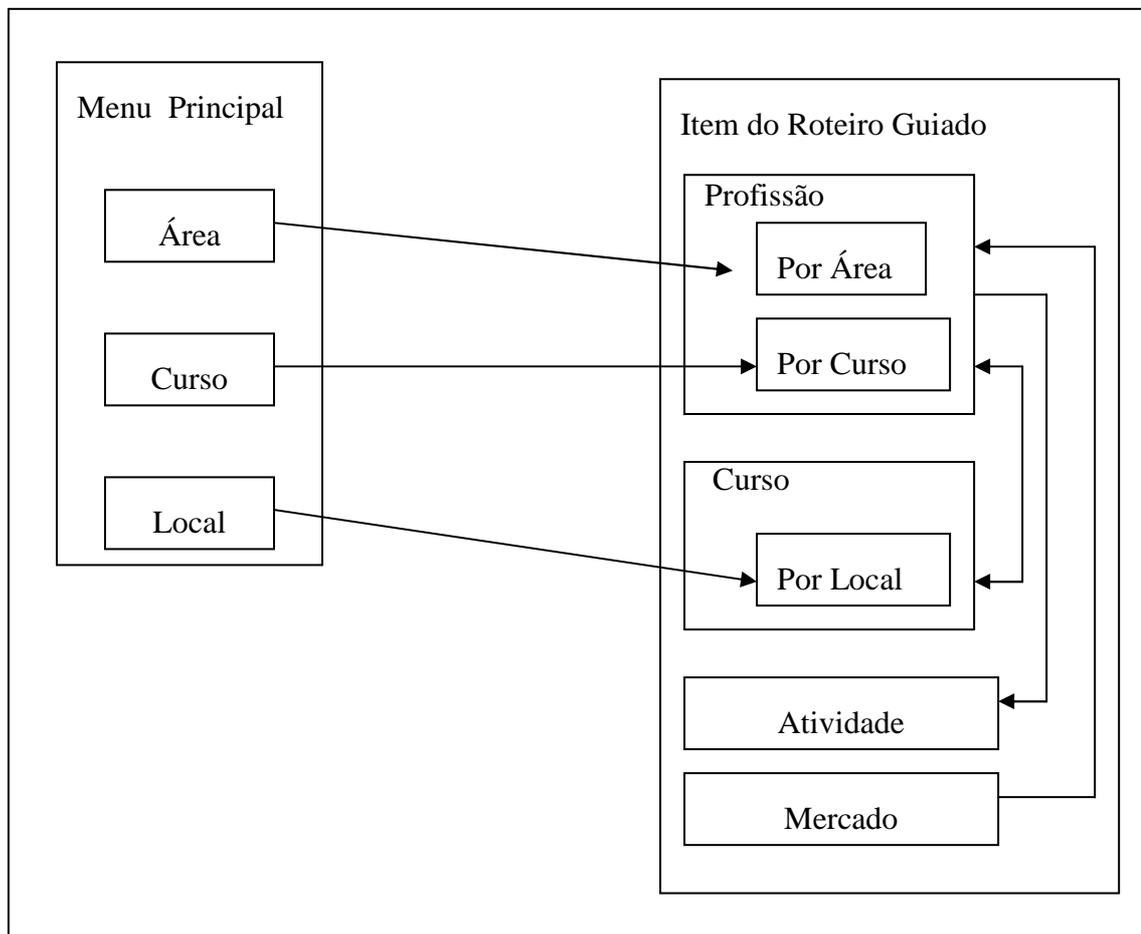


Fonte: adaptado de Pompermaier (1999).

Os elos são mapeados dos relacionamentos do esquema conceitual e compõem-se dos nós de origem, nós de destino, cardinalidade e outros atributos decorrentes da relação. As estruturas de acesso são utilizadas para auxiliar o usuário final na procura de uma informação desejada. Menus, Índices e Roteiros Guiados são alguns exemplos de estruturas de acesso.

Em OOHDm a principal estrutura do esquema navegacional é a noção de Contexto de Navegação (Schwabe, 1992). Este contexto constitui-se de um conjunto de nós organizados segundo algum critério e de elos que dão a ordenação a esses nós, formando um caminho de navegação. A figura 14 mostra o esquema de navegação.

Figura 14 - Esquema Navegacional



Fonte: adaptado de Pompermaier (1999).

4.2.3 PROJETO DE INTERFACE ABSTRATA

Muitas das metodologias atuais para a modelagem de aplicações hipermídia tendem a negligenciar a construção de interface homem-aplicação e vice-versa (Rossi, 1995). Estas costumam ser totalmente construídas com uso de ferramentas dependentes da implementação e do ambiente.

Esta fase tem a pretensão de projetar uma interface, visando o usuário final, sem relacionar-se com o ambiente e a implementação da aplicação. Uma vez construído o projeto navegacional, pode-se criar as possíveis interfaces que a aplicação terá.

Para cada especificação de um nó é possível definir diferentes visões de interface de usuário, este especificando o que será mostrado em cada nó, como, por exemplo, imagens, informações e *links*.

Em OOHDm a especificação da interface do usuário é realizada utilizando a abordagem das Visões de Dados Abstratos (ADV - *Abstract Data View*) (Cowan, 1995). Estes são modelos orientados a objetos utilizados para representar os aspectos estruturais e dinâmicos de uma interface de usuário.

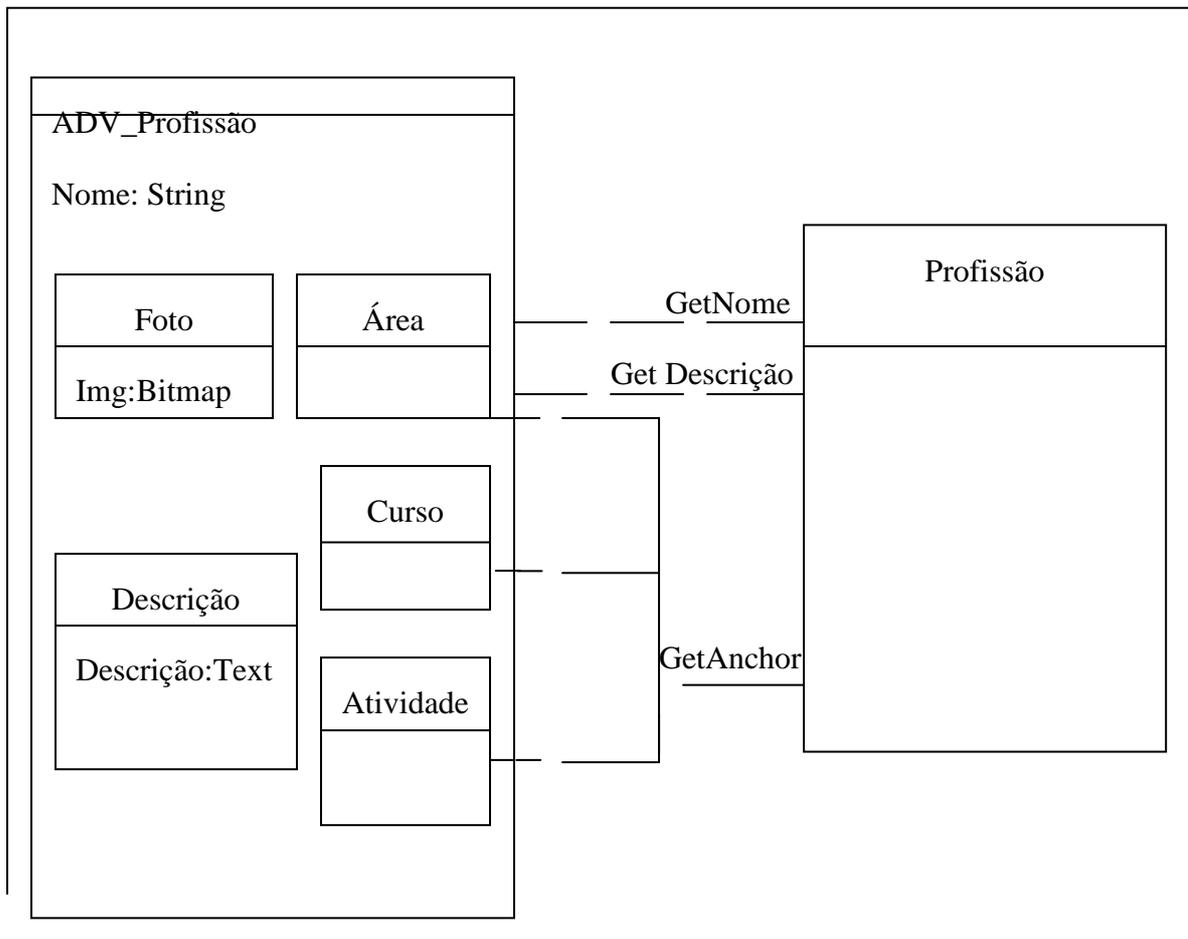
Os ADVs são objetos no sentido em que possuem um estado e uma interface, e sua interface pode ser exercida através de mensagens de outros objetos, ou de eventos de entrada e saída. São abstratos no sentido em que representam apenas a interface e o estado, e não a implementação.

Para a especificação das interfaces do usuário, utilizando ADVs é necessário, também, a utilização de *Abstract Data Object* (ADO), sendo estes ADVs que não suportam eventos externos, como um *click* de *mouse*. Os ADOs são utilizados para gerenciar as estruturas de dados e controles dentro de uma aplicação.

Em OOHDm os objetos navegacionais, tais como os nós, os elos, ou as estruturas de acesso agirão como ADOs e o ADV a eles associado será utilizado na especificação de sua aparência com o usuário.

Os Diagramas de Configuração são úteis na expressão de padrões de comunicação entre objetos em termos de serviços oferecidos e requeridos (Schwabe, 1995). No contexto de OOHDm é importante o modo como usuário irá interagir com a aplicação e, especialmente, quais objetos de interface causarão a navegação. A figura 15 mostra o diagrama de configuração da classe.

Figura 15 - Diagrama de Configuração



Fonte: adaptado de Pompermaier (1999).

4.2.4 IMPLEMENTAÇÃO

Para obter a implementação deve-se mapear os modelos navegacionais e interface abstrata em objetos concretos, em um ambiente de implementação. É nesta fase que é realizado o mapeamento dos objetos perceptíveis e suas transformações, definidos no projeto de interface abstrata para os objetos disponíveis no sistema de hipermídia escolhido para a implementação (Pompermaier, 1999).

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 (Rossi, 1996).

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. Como será visto, há muitas implementações possíveis, mas nenhuma delas destaca-se por alcançar tanto maior desempenho como maior flexibilidade e facilidade de manutenção. Daremos ênfase às soluções que ofereçam uma transição inconsútil do projeto à implementação. Algumas características indesejáveis do *Toolbook* (e de ambientes hipermídia similares) devem ser consideradas: os objetos de interface e os objetos de navegação correspondentes são normalmente definidos como o mesmo objeto, isto é, não há uma separação clara entre os interesses da interface e os das entidades navegacionais; além disto, o suporte de composição dinâmica é reduzido: no *Toolbook*, apenas um *background* é permitido por cada página; etc.

4.3 AMBIENTE DE AUTORIA

No mercado brasileiro existem ambientes de autoria. Observando-as detalhadamente pode-se chegar a basicamente três estilos de ferramentas. Estes estilos podem ser classificados em ferramentas baseadas em: linha de tempo, pilha de cartões e icônicas.

Segundo Badgett (1994), pode-se caracterizá-las pelos seguintes aspectos:

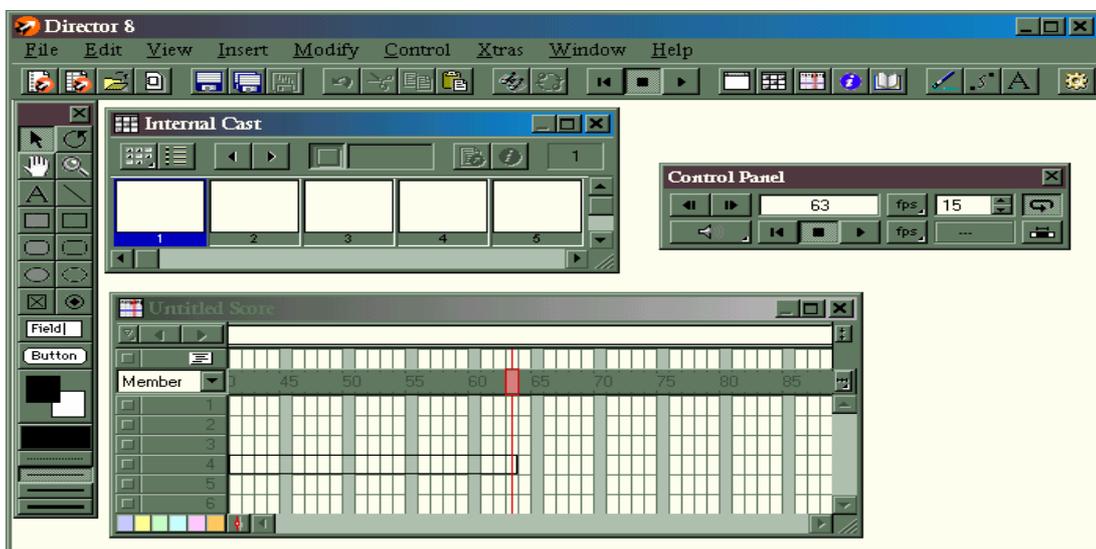
- a) Linha de tempo: este tipo de ferramenta permite que o desenvolvedor trabalhe com os objetos em uma linha de tempo, ou seja, alinha-se os quadros, figuras e sons para serem executados em um determinado instante e por um certo período. Dentro desta filosofia destacam-se as seguintes ferramentas: Tempra Gold, ToolBook e Director;
- b) Pilha de cartões: segue a filosofia de que se tem uma pilha de cartão e que para cada cartão indique qual é o próximo cartão a ser apresentado. No mercado destacam-se: HyperCard e Hyper Studio;
- c) Icônica: desenvolve-se a aplicação através de ícones que indicam qual é a ação a ser executada e qual é o próximo ícone a ser executado. Estas ferramentas apresentam o programa em forma de fluxograma. Dentro deste enfoque pode-se achar as seguintes ferramentas: IconAuthor, AuthoWare e HSC Interact.

4.3.1 MACROMEDIA DIRECTOR

O director é uma ferramenta bastante complexa com um amplo conjunto de recursos que poderá ser utilizado para criar apresentações, animações e aplicações hipermídia.

É composto por quatro elementos principais, são eles: o *stage*, o *cast window*, o *score* e o *control panel*. A figura 16, mostra a tela principal do Director, composta pelos seus componentes.

Figura 16 - Tela Principal do Director



Abaixo, os quatro elementos principais serão explicados individualmente:

- a) Stage: o stage (palco) é a janela que aparece o filme que está sendo criado (Principia, 1996). O stage é a tela mais visível do Director. Com o comando Window/Stage pode-se ocultar todas as janelas, menos o stage, que vai ficar sempre visível;
- b) Cast (elenco) contém os cast members (membros do elenco). Os cast members são os elementos básicos de qualquer filme do director (Miller, 1996). Os cast podem conter imagens, texto, sons, palhetas de cores, etc. O director fornece uma janela cast com 1000 espaços em branco (cast members). Mas é possível utilizar mais de 32.000 cast members em um único filme (Henderson, 1997);
- c) Score (grade) contém todos os cast members, e os outros fatores que controlam o filme. O Score é que vincula os dados exportados do cast. Nele as animações

são feitas colocando-se um gráfico no cenário, e mudando levemente a sua posição, em vários quadros. O Score oferece efeitos de transições, ajuste de palhetas de cores e controle de tempo. As colunas do score representam os frames e as linhas representam os canais;

- d) Control Panel (painel de controle) proporciona um conjunto de controles similares aos de um vídeo cassete. Pode ser usado para tocar, parar ou voltar um filme ao seu início.

4.3.1.1 LINGUAGEM LINGO

A linguagem de programação utilizada pelo Director é o Lingo. O Lingo não se faz necessário em uma simples aplicação, o Director possui bons recursos para animação, sem precisar associar a linguagem Lingo a eles. Usa-se o Lingo nas animações mais complexas.

O Lingo fornece a habilidade de controlar a ação do filme mais precisamente, também fornece atributos adicionais para os cast members e adiciona interatividade aos filmes (Miller, 1996).

O director possui ainda os seguintes recursos (Vaughan, 1994):

- a) animações, gráficos, imagens do *QuickTime*, efeitos sonoros e clipes musicais;
- b) MacroMedia *KnowledgeBase*, que possui vários avisos técnicos, perguntas e respostas comuns ao suporte técnico, dicas e truques;
- c) modelos de trabalho do Swivel 3D_Pro, ModelShop_II e SoundEdit_Pro;
- d) exemplos avançados de programas Lingo, prontos para serem copiados, colados e personalizados.

Segundo Small (1996), o Lingo não pode ser considerado uma linguagem orientada a objetos, mas sim como uma linguagem que permite a programação orientada a objetos.

Os termos utilizados pelo Lingo, são diferentes na literatura relacionada à orientação a objetos, como mostra o quadro 03.

Quadro 03 - Termos Lingo

Termo/Lingo	Equivalente Orientação Objeto
Parent script	Classe (class)
Child object	Instância de classe (class instance)
Property variable	Variável de Instância (instance variable)
Handler	Método
Ancestor script	Super classe

Fonte: adaptado de Macromedia (1997).

O lingo possui três características que permitem a orientação a objeto (Small, 1996):

- a) A função New: esta função permite fazer cópias idênticas de um objeto, através de um script, denominado parent script;
- b) Property: esta função permite que os objetos mantenham suas próprias variáveis globais privadas;
- c) Ancestor: permite que os objetos dividam propriedades ou handlers.

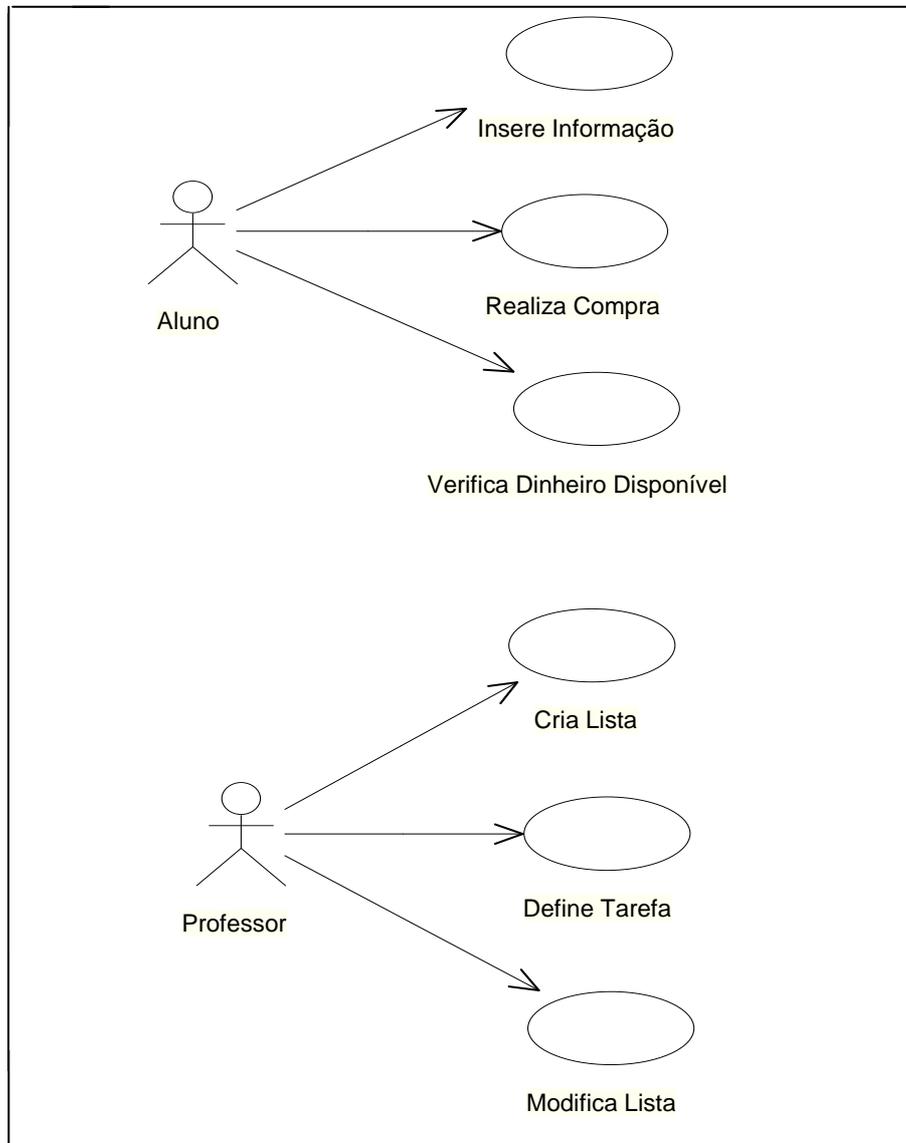
5 DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo será apresentada a especificação do protótipo utilizando a ferramenta *Rational Rose*, mostrando os diagramas de caso de uso, esquema conceitual, esquema navegacional e diagrama de configuração.

5.1 ESPECIFICAÇÃO DO PROTÓTIPO

O propósito principal do trabalho desenvolvido é proporcionar a interação com o software por parte do aluno e do professor. O protótipo simula um supermercado, onde o aluno irá interagir com o software desde a primeira tela, onde ele deverá informar dados, como nome, idade e a quantidade de dinheiro disponível. Feito isso, o professor entrará em ação escolhendo a lista de compras que o aluno deverá cumprir. O professor poderá optar por escolher uma lista já cadastrada ou criar uma lista nova. Quando selecionada a lista pelo professor, o aluno começará suas compras entrando assim no supermercado. O objetivo principal é fazer com que o aluno cumpra a tarefa, ou seja, o aluno deverá comprar todos os produtos da lista selecionada, e no final de suas compras, ele precisará acertar o valor de suas compras. A cada resposta do aluno, o sistema dará uma mensagem dizendo se ele acertou o valor das compras, ou se faltou algum produto para ser comprado, ou se o dinheiro que ele informou no início do programa é insuficiente, ou enfim, se ele errou o valor total das compras. Sendo assim, este trabalho foi desenvolvido com intuito de fazer com que o aluno adquira um melhor raciocínio matemático.

Para a especificação deste protótipo, será utilizada a metodologia orientada a objetos OOADM, apresentada no Capítulo 4. Segundo Castro (1998), cada vez mais as equipes responsáveis pelo desenvolvimento de sistemas são pressionados a criarem softwares da mais alta qualidade e da forma mais rápida possível. A ferramenta *Rational Rose* facilita este processo, tornando possível a modelagem de desenvolvimento. O padrão chamado *Unified Modeling Language* (UML), é o mais utilizado hoje em dia, pois fornece uma documentação interativa e automática dos processos de modelagem, sendo também possível a geração de códigos a partir desta mesma modelagem. Através da figura 17 é apresentada a especificação formal do problema utilizando o diagrama de caso de uso, onde foram identificados dois atores e seis casos de uso principais do sistema.

Figura 17 – Diagrama de caso de uso

Depois de identificados os atores, segue a definição de cada um deles no quadro 04.

Quadro 04 – Atores do sistema

Aluno	É o ator primário para o referido sistema. Ele é o principal ator que utilizará o software e obterá benefícios.
Professor	É o ator secundário, que tem por função definir as tarefas que deverão ser cumpridas pelo ator primário, bem como fazer mudanças nas tarefas se achar necessário.

Seis casos de uso foram definidos para o sistema e são definidos no quadro 05.

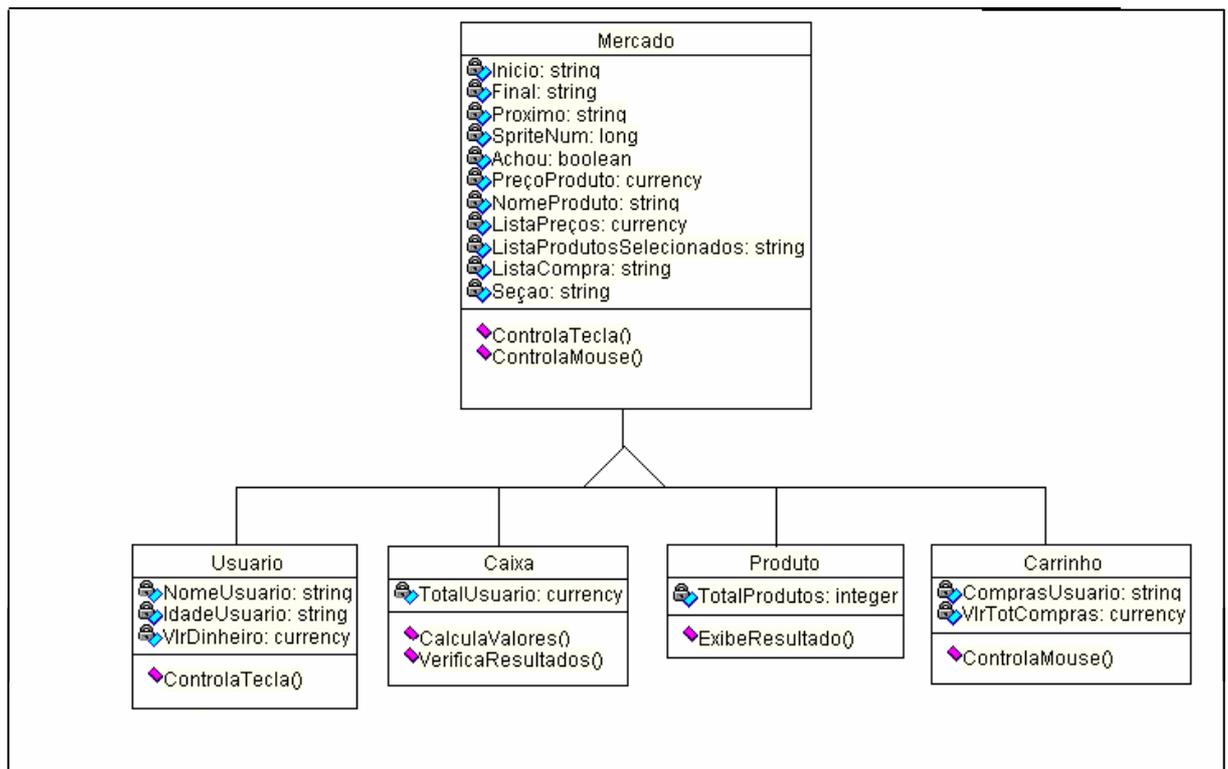
Quadro 05 – Casos de uso do sistema

Inserir informação	O aluno insere as informações que o sistema solicita antes de ele realizar as compras.
Realiza compra	Depois de definida a tarefa, o aluno tem por obrigação cumprir a tarefa criada pelo professor, sendo esta tarefa uma compra.
Verifica dinheiro disponível	No final das compras, o aluno terá que verificar se o dinheiro que ele tem é suficiente para a referida compra que ele efetuou.
Cria / Modifica lista	Na hora do professor definir a tarefa a ser cumprida, ele poderá fazer a criação de novas listas, bem como usar alguma lista já pronta.
Define tarefa	Quando já criada as listas, o professor poderá escolher a lista que deseja usar como tarefa para seus alunos.

Modifica lista	O professor poderá interagir com o software através de mudanças dentro das listas de compras, podendo incluir ou excluir os produtos já colocados nas prateleiras, além de poder modificar os preços.
-----------------------	---

Para demonstrar o relacionamento entre classes, a metodologia OOADM propõe a elaboração do Modelo Conceitual. Este modelo é constituído basicamente de classes com seus métodos. Existem cinco classes, cada uma com seus atributos, e ligadas entre si, através de agregação, relacionamento, generalização e especialização, conforme demonstrado na Figura 18.

Figura 18 – Modelo Conceitual



Depois de definido o Diagrama Conceitual com suas classes, é mostrado a descrição de cada uma delas.

Classe Mercado: é a classe principal do diagrama, pois esta classe contém relação com as outras quatro classes. Contém os seguintes atributos:

- Início: indica o início do caminho (frame) que poderá ser feito pelo aluno dentro do supermercado;
- Proximo: indica o próximo frame do deslocamento dentro do supermercado;
- Fim: indica o fim do caminho (frame) do deslocamento dentro do supermercado;
- SpriteNum: contém o número do sprite referente ao produto na prateleira;
- Achou: indica se o aluno achou o produto que esta na lista de compras;
- PreçoProduto: contém o preço do produto que está no supermercado;
- NomeProduto: contém o nome do produto que está no supermercado;
- ListaPreços: contém o preço de todos os produtos existentes no supermercado;
- ListaProdutosSelecionados: contém os produtos comprados pelo aluno;
- ListaCompra: contém os produtos que deverão ser comprados pelo aluno;
- Seção: indica a seção onde o produto se encontra dentro do supermercado.

Classe Usuario: esta classe inclui como atributos os dados do usuário, além de que ela herda também todos os atributos da classe mercado. Descrevendo os atributos:

- NomeUsuario: contém o nome do aluno que irá interagir com o software;
- IdadeUsuario: contém a idade do aluno que irá interagir com o software;
- VlrDinheiro: contém a quantidade de dinheiro disponível informado pelo aluno, no qual é o valor que ele poderá usar para fazer suas compras;

Classe Caixa: como a classe usuário esta classe também herda todos os atributos da classe mercado. Como atributo esta classe contém:

- TotalUsuario: contém o total das compras efetuadas pelo aluno;

Classe Produto: como as outras classes, esta classe também herda todos os atributos da classe mercado. Como atributo esta classe contém:

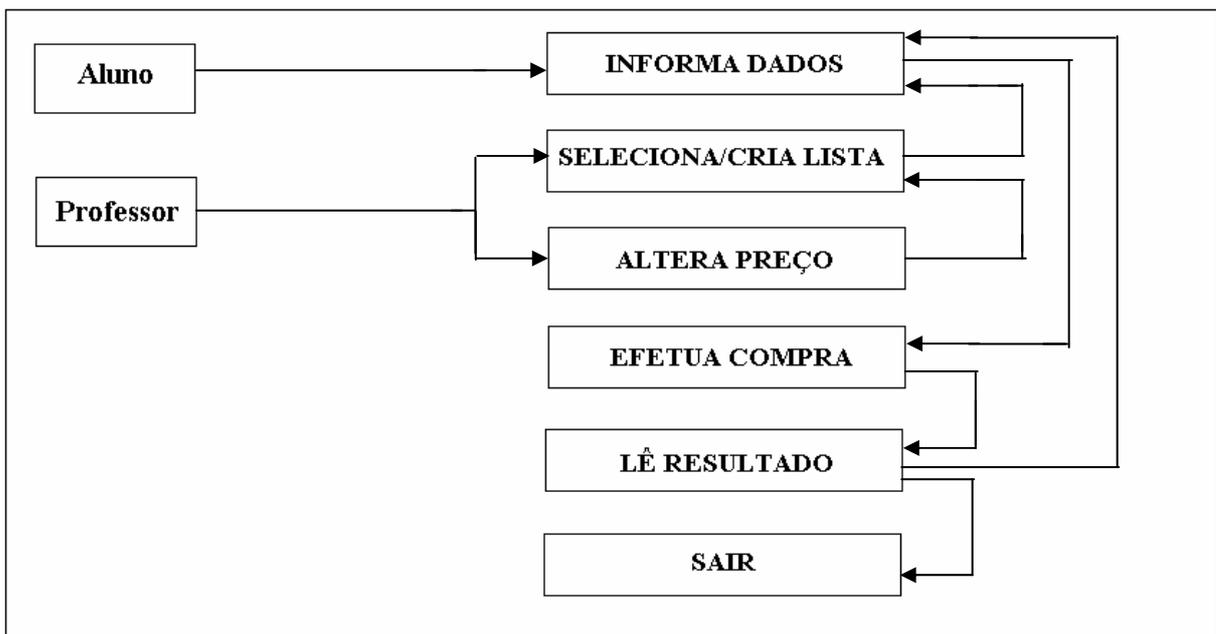
- TotalProdutos: contém a quantidade de produtos que estão na lista de compras;

Classe Carrinho: esta classe herda também os atributos da classe mercado. Seus atributos são:

- ComprasUsuario: indica as compras efetuadas pelo aluno;
- VlrTotCompras: indica o valor total das compras efetuadas pelo aluno.

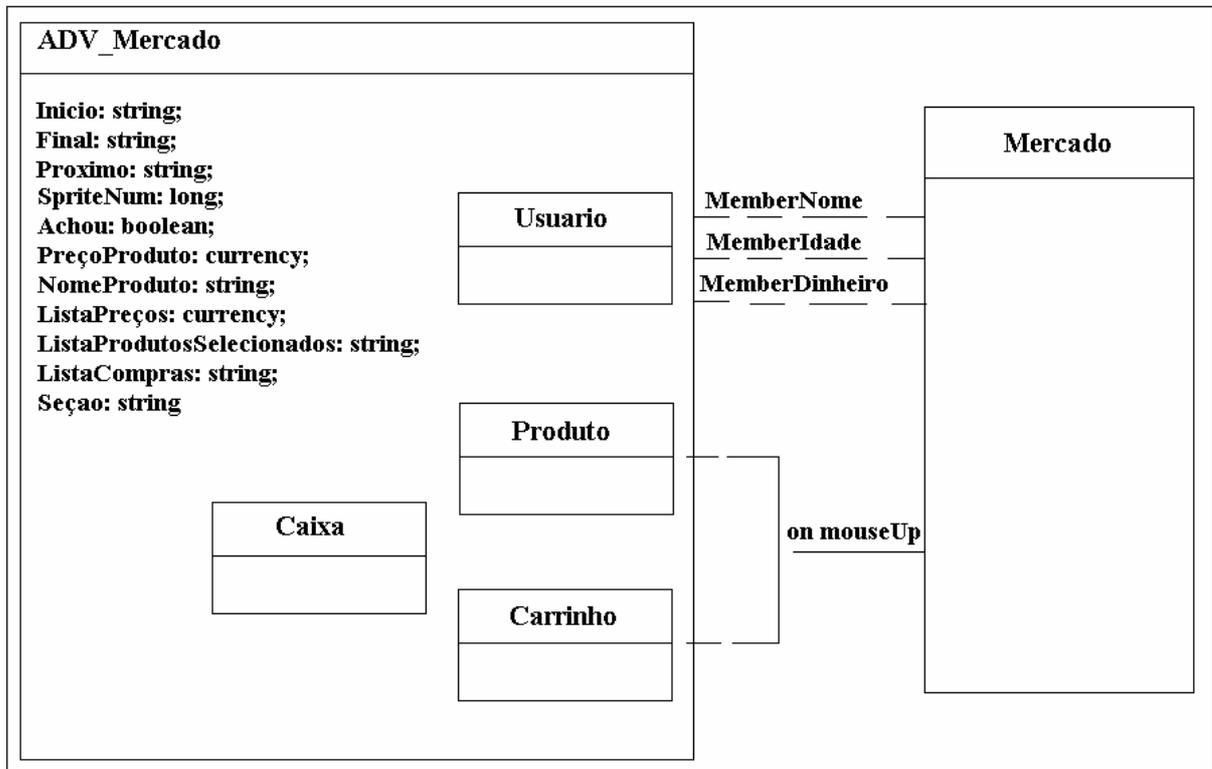
Ainda através da metodologia OOADM, será apresentada na figura 19 o Modelo Navegacional, onde ajudará o usuário na procura da informação desejada.

Figura 19 – Modelo Navegacional



Em seguida, na figura 20, é mostrado o Diagrama de Configuração da classe Mercado.

Figura 20 – Diagrama de Configuração



5.2 IMPLEMENTAÇÃO

5.2.1 OPERACIONALIDADE DA IMPLEMENTAÇÃO

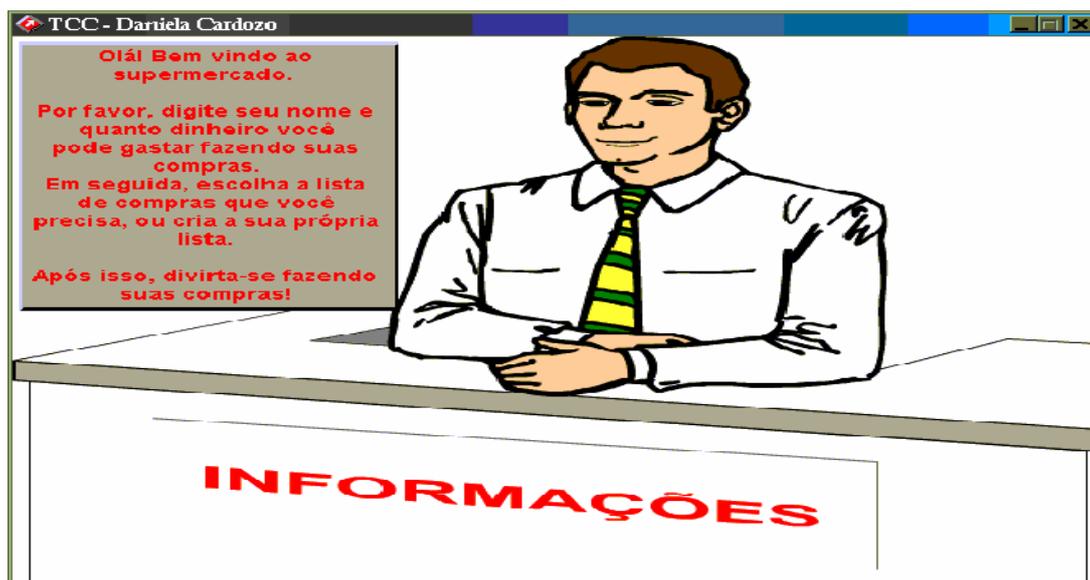
Neste item será apresentada a operacionalidade da implementação. Serão mostradas as telas, bem como suas características e alguns códigos. Para a implementação deste protótipo foi utilizado o software Macromedia Director, versão 8.0, citado no capítulo 4.3.1. A figura 21, apresenta a tela de abertura do protótipo.

Figura 21 – Tela de abertura do protótipo



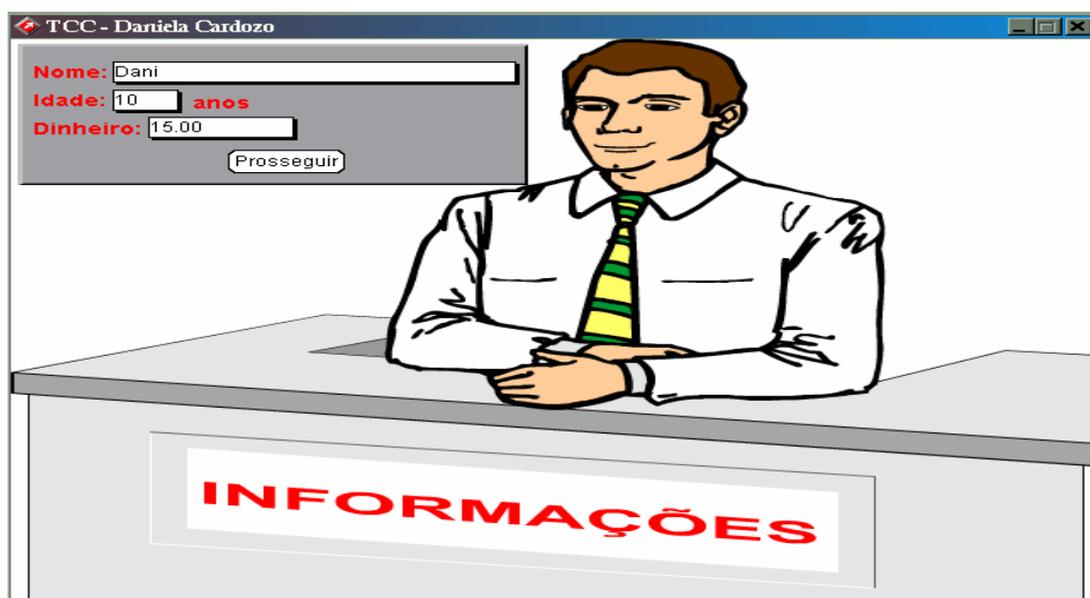
Após a tela de abertura, o protótipo, através da figura 22, apresenta a tela de boas vindas ao usuário.

Figura 22 – Tela de boas vindas



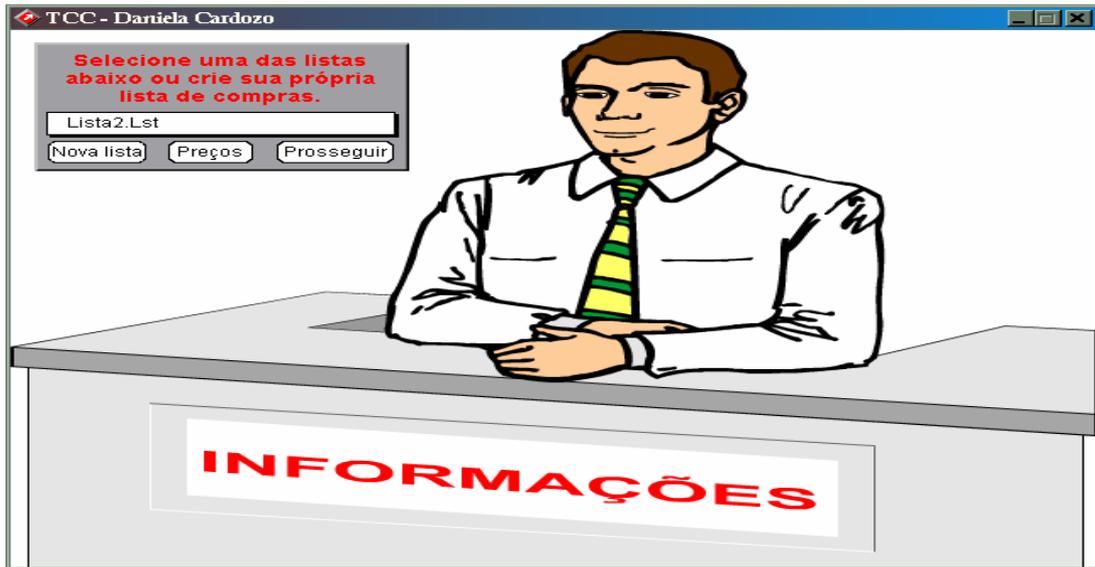
A figura 23 representa a tela onde o usuário deverá informar o seu nome, sua idade e quantidade de dinheiro disponível. Nota-se que sem essas informações o usuário não poderá avançar para a próxima tela.

Figura 23 – Tela das informações do usuário



A próxima tela, a figura 24, solicita ao usuário que ele escolha uma das opções: uma lista já cadastrada ou crie outra lista. Poderá ainda mudar os preços dos produtos que achar necessário. Abaixo, no quadro 06, é demonstrado o código para selecionar lista.

Figura 24 – Tela para selecionar a lista de compra



Quadro 06 – Código para selecionar lista

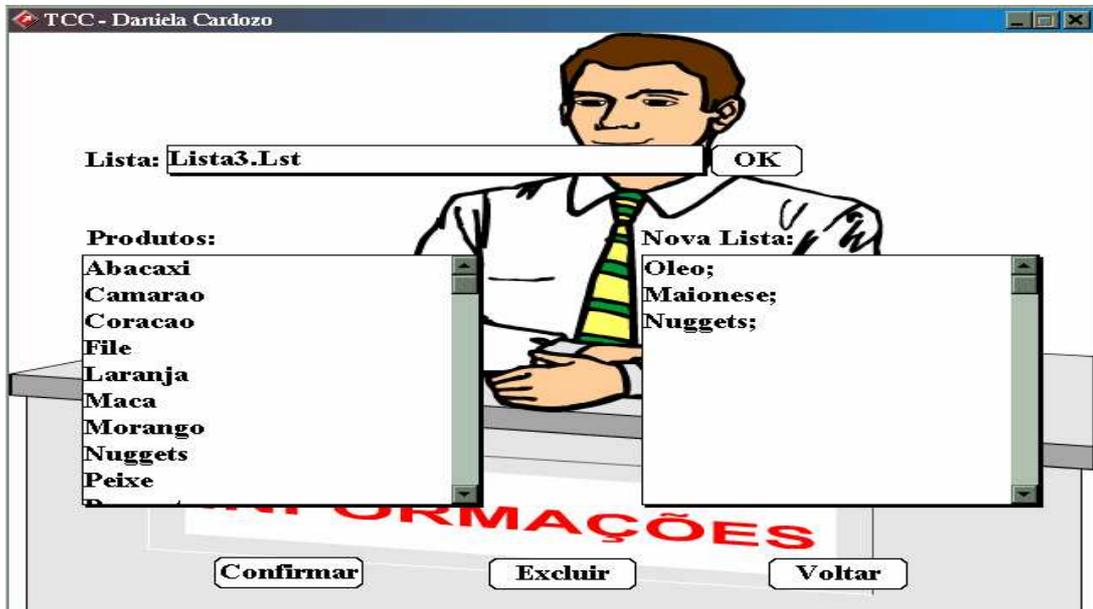
```

on SeleccionaListas
  NomePasta = the MoviePath
  set ListaItens = []
  Contador = 0
  repeat while TRUE
    Contador = Contador + 1
    NomeArquivo = getNthFileNameInFolder (NomePasta, Contador)
    if NomeArquivo = EMPTY then
      sendSprite (12, #DropList_SetContents, ListaItens)
      return
    else
      Tamanho = offset(".lst",NomeArquivo)
      if Tamanho > 0 then
        ListaItens.add(NomeArquivo)
      end if
    end if
  end repeat
end

```

Na figura 25, é apresentada a tela onde o usuário poderá criar listas novas.

Figura 25 – Tela para criação de novas listas



Depois do usuário ter escolhido uma lista cadastrada ou ter criado uma nova lista, ele passará para a figura 26, que é a tela da área interna do supermercado, onde o usuário começará a fazer suas compras de acordo com a lista escolhida. O código correspondente a movimentação das teclas dentro do supermercado poderá ser visto no quadro 07.

Figura 26 – Tela interna do supermercado



Quadro 07 – Código correspondente ao movimento das teclas

```

on keyDown me
  Tecla = the keyCode
  member("Tip").Text = EMPTY
  if (Tecla = 126) and (the frameLabel <> pFinal) then
    play the frame + 1
  else if (Tecla = 125) and (the frameLabel <> pInicio) then
    play the frame - 1
  else if (Tecla = 118) then
    if sprite(pSpriteLista).visible = TRUE then
      sprite(pSpriteLista).visible = FALSE
    else
      sprite(pSpriteLista).visible = TRUE
    end if
  else if (the Key = RETURN or the Key = ENTER) and (the frameLabel = pFinal) then
    cursor 4
    play frame pProximo
  end if
end keyDown

```

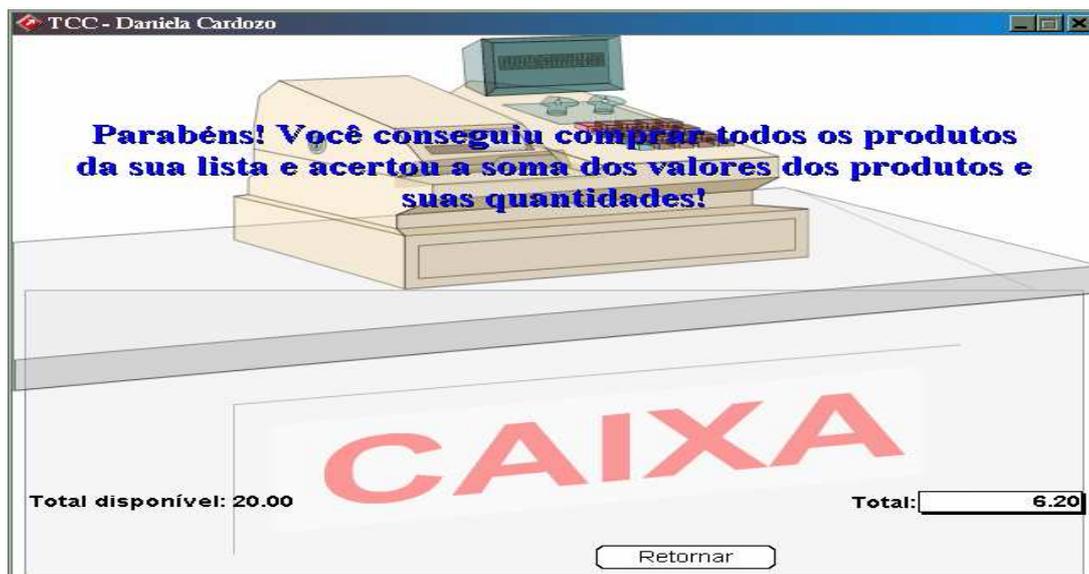
Depois de concluída as compras, o usuário passará para a tela dos resultados. A figura 27 abaixo, apresenta os produtos que o usuário comprou.

Figura 27 – Tela das compras efetuadas



Depois do usuário ter visto o resultado de suas compras, ele deverá calcular quanto foi o total e se ele acertar o valor uma mensagem será dada a ele, como a mostrada na figura 28.

Figura 28 – Tela de parabéns



Quando o usuário fizer as compras, e tiver esquecido de comprar algum produto aparecerá uma mensagem, como a mostrada na figura 29.

Figura 29 – Tela de alerta de falta de produtos



Quando o usuário tiver comprado todos os produtos da lista, e, na hora de colocar o total das compras, o valor das compras for maior que o valor do dinheiro que ele informou, o sistema dará uma mensagem, como mostra a figura 30.

Figura 30 – Tela de alerta quanto ao valor do dinheiro



Quando do final das compras, depois do usuário ter comprado todos os produtos solicitados, o usuário deverá informar o valor das compras, se acaso ele errar, o sistema dará uma mensagem, como mostra a figura 31.

Figura 31 – Tela de alerta ao somatório dos valores das compras



No quadro 08 é mostrado o código que calcula os resultados.

Quadro 08 – Código referente aos resultados

```

on CalculaResultados
  global ListaCompra
  global ListaProdutosSelecionados
  global ListaPrecos
  member("SaudacaoCaixa").text = EMPTY
  member("PrecoFinal").editable = FALSE
  ListaCompra.sort()
  ListaProdutosSelecionados.sort()
  ListaPrecos.sort()
  ListaFinal= [:]
  Contador = 1
  ValorFinal = 0
  ValorLista = 0
  repeat with I = 1 to ListaCompra.count
    Quantidade = 0
    Preco = 0
    repeat with H = 1 to ListaProdutosSelecionados.count
      if ListaProdutosSelecionados[H] = ListaCompra[I] then
        Quantidade = Quantidade + 1
      end if
    end repeat
    if Quantidade > 0 then
      Preco = ListaPrecos[ListaCompra[I]]
      Total = value(Preco) * Quantidade
      ListaFinal.addProp (ListaCompra[I], Total)
      ValorFinal = ValorFinal + Total
      Contador = Contador + 1
    end if
  end repeat
  repeat with I = 1 to ListaCompra.count
    ValorLista = ValorLista + value(ListaPrecos[ListaCompra[I]])
  end repeat
  if ValorFinal > value(member("Dinheiro").text) then
    Temp = member("Nome").text && ", infelizmente o dinheiro que você trouxe"
    Temp = Temp && "é insuficiente para pagar todos os produtos que você"
    Temp = Temp && "selecionou. Volte mais tarde com mais dinheiro para poder"
    Temp = Temp && "levar todos os produtos."
  else
    if value(member("PrecoFinal").text) <> ValorFinal then
      Temp = member("Nome").text && ", você infelizmente errou a soma total das compras."
      Temp = Temp && "Tente novamente!"
    else
      if ValorFinal = ValorLista then
        Temp = "Parabéns! Você conseguiu comprar todos os produtos da sua lista e"
        Temp = Temp && "acertou a soma dos valores dos produtos e suas quantidades!"
      else
        Temp = member("Nome").text && ", infelizmente você não conseguiu comprar todos"
        Temp = Temp && "produtos da sua lista. Tente novamente!"
      end if
    end if
  end if
  member("Resultados").text = string(Temp)
end CalculaResultados

```

6 CONCLUSÃO

Como o uso da informática na educação hoje está cada vez maior, percebeu-se com este trabalho a necessidade da criação de muitos outros softwares hipermídia no mercado. Este trabalho foi concluído com intuito de complementar o ensino dentro de uma sala de aula, podendo ajudar o professor e o aluno na disciplina de matemática.

Depois de concluído este trabalho pode-se perceber que os objetivos foram atingidos, permitindo assim a total interação do professor e do aluno com o software. São eles:

- a) permitir que o aluno efetue suas compras passeando pelo supermercado;
- b) possibilitar ao aluno a escolha de produtos, desde que cumpra a tarefa exigida pelo professor;
- c) o aprendizado das quatro operações matemáticas básicas;
- d) avaliar se o aluno cumpriu a tarefa estabelecida;
- e) possibilitar ao professor o cadastro de novas listas, bem como poder modifica-las.

Contudo, ao término do desenvolvimento do protótipo, chegou-se à conclusão de que a ferramenta utilizada, o software Macromedia Director, versão 8.0, não seria apropriado para desenvolver o produto final, pelos motivos citados abaixo:

- a) Para cada novo produto a ser inserido no software, seria necessário fazer vários cálculos para dimensionar e posicionar o produto na tela, sendo que um mesmo produto poderia aparecer mais de uma vez;
- b) Seria necessário calcular a posição de cada prateleira a ser exibida na tela;
- c) Todos os novos cálculos consumiriam um tempo muito grande, fazendo com que o software não atendesse às necessidades do usuário.

Com isso, como sugestão para poder chegar ao desenvolvimento do produto final, seria interessante utilizar o software Direct X da Microsoft, pelo fato de que este software contém rotinas do windows somente para gráficos.

6.1 SUGESTÕES PARA TRABALHOS FUTUROS

Algumas sugestões podem ser citadas para futuros trabalhos a serem desenvolvidos:

- a) ampliar ainda mais a área do supermercado, com inclusão de mais produtos;
- b) permitir o acesso simultâneo via internet de diversos usuários;
- c) incluir agentes inteligentes para sugerir ao usuário produtos que devem ser comprados com preços menores;

REFERÊNCIAS BIBLIOGRÁFICAS

ANGIOLETTI, Jocir Leandro. **Protótipo de um software educacional 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

BADGETT, Tom; SANDLER, Corey. **Criando Multimídia em seu PC**. São Paulo: Makron Books, 1995.

BIANCHI, Daniela. **Estudo ergonômico de software educacional para crianças de 5 a 7 anos**. 1997. 120f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau

BIZZOTTO, Carlos Eduardo Negrão. **Director 8 Rápido e Fácil**. São Paulo: Makron Books, 2000.

BIZZOTTO, Carlos Eduardo Negrão. **Concepção, Desenvolvimento e Avaliação de um Ambiente extensível para o Aprendizado cooperativo distribuído**. 1999. 130 f. Projeto submetido para Qualificação do Programa de Pós-graduação em Engenharia de Produção - Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis.

CABRAL, Pedro Manso. **A hora da Multimídia**. Informática Exame, São Paulo: Abril, Ano 9, n.98, maio/1994, p. 144.

CAMPOS, Gilda H. B. de; ROCHA, Ana Regina C. **Manual para Avaliação da Qualidade do Software Educacional**. Rio de Janeiro: Publicações Técnicas, 1996.

CASTRO, Alexandre R.; COSTA, Elenio Pereira. **Um breve estudo sobre a ferramenta Rational Rose**. Rio Grande do Sul. [1998?]. Disponível em <<http://www.fsb.com.br/base/Rational/F75.htm>>. Acesso em: 17 set 2001.

COLL, César. **Construtivismo na Sala de Aula**. Tradução Cláudia Scilling. São Paulo: Atica, 1996, 221p.

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

HENDERSON, Jason. **Director 6.0 Demystified**. Califórnia: Macromedia, 1997

INFORMÁTICA EXAME, **Home PC**. São Paulo: Abril, n.3, mar/1995. Edição Especial.

LAGÔA, Ana. **Como a criança aprende segundo Piaget**. Nova Escola, São Paulo. Fundação Victor Civita, ano 9, n. 76, junho/1994, p.20-25.

MACROMEDIA, **Macromedia director 6.0: using director**. São Francisco: Macromedia, 1997.

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

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

MILLER, Débora M.; MILLER, Michael D. **Macromedia Director 5.0 Power Toolkit**. São Francisco: Ventana, 1996.

PIAGET, Centro Educacional Jean Piaget. **Escolinha “A chave do tamanho”**. Rio de Janeiro. [1998?]. Disponível em <<http://www.jeanpiaget.com.br>>. Acesso em: 27 ago 2001.

PRINCIPIA, Tecnologia da Informação. **Macromedia Director 5.0. apostila do curso de software director**. Rio de Janeiro:[s.n], 1996

POMPERMAIER, Leandro B. **Modelagem de aplicações hipermídia: uma experiência com OOHDM**. [S.l]. [1999?]. Disponível em < <http://inf.ufrgs.br/~pomper/oohtm.html>>. Acesso em: 27 ago 2001.

ROSSI, Gustavo. **Projeto de aplicações hipermídia**. [S.l]. [1996]. Disponível em <<http://sol.info.unlp.edu.ar/~fer/oohtm/pg001.htm>>. Acesso em: 27 ago 2001.

ROSSI, Gustavo et al. **An Object-oriented modeling and design the human-computer interface of hypermedia applications**. Proceedings of the International workshop on Hypermedia Design, 1995.

RUMBAUGH, M. B. et al. **Object-oriented modeling and design**. Englewood Cliffs: Prentice-Hall, 1991

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

SCHWABE, Daniel et al. **Hypertext development using a model-based approach**. Software and Experience. New York, v.22, n.11, p.937-962, nov. 1992.

SMALL, Peter. **Lingo Sorcery: The Magic of Lists, Objects and Intelligent Agents**. London: John Wiley and sons, 1996.

SOMENSI, Andréia. **Aplicação da Tecnologia de Sistemas Especialistas, em um Software Educacional baseado na Pedagogia Freinet**. 1997. 76f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau

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

VAUGHAN, Tay. **Multimídia na Prática**. Tradução Elaine Andrade Pizzoli; Revisão técnica Equipe Makron Books de Informática. São Paulo: Makron, 1994.

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**. Seattle: University of Washington, 1993.

