

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

**SOFTWARE MULTIMÍDIA PARA AUXILIAR NO PROCESSO
DE ENSINO-APRENDIZAGEM DA INFORMÁTICA A
PESSOAS DA TERCEIRA IDADE.**

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

ITATIANA BÁRBARA NOVAK WENDT

BLUMENAU, JULHO/2002

SOFTWARE MULTIMÍDIA PARA AUXILIAR NO PROCESSO DE ENSINO-APRENDIZAGEM DA INFORMÁTICA A PESSOAS DA TERCEIRA IDADE.

ITATIANA BÁRBARA NOVAK WENDT

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

AGRADECIMENTOS

Agradeço primeiramente a Deus pela realização deste trabalho, por dar-me motivação para que eu pudesse concluí-lo.

Um agradecimento especial aos meus pais, José e Telma, por sempre acreditarem na minha capacidade e perseverança, motivando-me e auxiliando-me em todos os momentos da minha vida, dando-me toda a estrutura necessária para minha formação.

Agradeço aos meus avós, tios e padrinhos, que direta ou indiretamente me auxiliaram para que eu pudesse chegar até aqui.

Agradeço em especial a compreensão de meu esposo Edinei, pelas sucessivas ausências durante a realização deste trabalho, e agradeço-o também, por sempre me motivar e acreditar no meu potencial.

Agradeço especialmente ao Professor Carlos Eduardo Negrão Bizzotto, pela paciência, compreensão e competência demonstrada durante a orientação e realização deste trabalho.

Um agradecimento especial aos chefes e amigos, Maciel, Flávio e Sandro pela compreensão e disponibilidade de recursos durante a realização deste trabalho.

Um agradecimento especial aos meus alunos, que contribuíram para que este trabalho fosse realizado.

SUMÁRIO

AGRADECIMENTOS	III
LISTA DE FIGURAS.....	VI
LISTA DE QUADROS.....	VIII
LISTA DE TABELAS	IX
RESUMO.....	1
ABSTRACT	2
1 INTRODUÇÃO	3
1.1 OBJETIVOS DO TRABALHO.....	4
1.2 JUSTIFICATIVA	5
1.3 LIMITAÇÕES	6
1.4 ESTRUTURA DO TRABALHO	6
2 SOFTWARES EDUCACIONAIS.....	8
2.1 HISTÓRIA DA INFORMÁTICA NA EDUCAÇÃO	8
2.2 EVOLUÇÃO DOS SOFTWARES EDUCACIONAIS.....	9
2.3 TIPOS DE SOFTWARES EDUCACIONAIS	10
2.3.1 TUTORIAL	11
2.3.2 EXERCÍCIO E PRÁTICA	12
2.3.3 AMBIENTE DE AUTORIA	12
2.3.4 SIMULAÇÃO.....	13
2.3.5 JOGOS	14
2.4 DESENVOLVIMENTO DE SOFTWARE EDUCACIONAL	16
2.4.1 DEFINIÇÃO DO AMBIENTE DE APRENDIZAGEM	16
2.4.2 ANÁLISE DE VIABILIDADE	17
2.4.3 SELEÇÃO DO TIPO DE DOCUMENTO.....	17
2.4.4 PLANEJAMENTO DA INTERFACE.....	18
2.4.5 PLANEJAMENTO DO DOCUMENTO	18
2.4.6 SELEÇÃO DO SISTEMA DE AUTORIA E DAS FERRAMENTAS ..	19
2.4.7 CARACTERÍSTICAS PEDAGÓGICAS.....	19
2.4.8 FACILIDADE DE USO	20
2.4.9 INTERFACE	20
2.4.10 ADAPTABILIDADE	21

2.4.11 DOCUMENTAÇÃO	21
3 A INFORMÁTICA NA TERCEIRA IDADE	22
3.1 PORQUE ENSINAR INFORMÁTICA ATRAVÉS DE UM JOGO ÀS PESSOAS DA TERCEIRA IDADE ?	24
4 ORIENTAÇÃO À OBJETOS	26
4.1 CONCEITOS BÁSICOS	26
4.2 USO DA UML	26
4.3 DIFERENTES VISÕES DO SISTEMA.....	28
4.4 DIAGRAMA DE CASOS DE USO	29
4.5 DIAGRAMA DE CLASSES	31
4.6 DIAGRAMA DE SEQÜÊNCIA DE EVENTOS	32
5 AMBIENTES DE DESENVOLVIMENTO	34
5.1 RATIONAL ROSE.....	34
5.2 AMBIENTE DE DESENVOLVIMENTO: DIRECTOR 8.....	35
5.2.1 STAGE	36
5.2.2 JANELA CAST	37
5.2.3 SCORE	38
5.2.4 CONTROL PANEL.....	38
5.2.5 LINGO	39
5.2.5.1 SCRIPTS	40
5.2.5.2 EVENTOS.....	40
5.2.5.3 LISTAS	41
6 SOFTWARE PROPOSTO	43
6.1 ESPECIFICAÇÃO DO SOFTWARE	44
6.1.1 DIAGRAMAS DE CASOS DE USO.....	44
6.1.2 DIAGRAMAS DE CLASSES.....	45
6.1.3 DIAGRAMAS DE SEQÜÊNCIA DE EVENTOS.....	48
6.2 IMPLEMENTAÇÃO E TUTORIAL DO SOFTWARE.....	51
6.3 RELATÓRIO DA UTILIZAÇÃO DO SOFTWARE	60
7 CONCLUSÃO.....	62
7.1 EXTENSÕES.....	62
REFERÊNCIAS BIBLIOGRÁFICAS	63

LISTA DE FIGURAS

FIGURA 1 - TUTORIAL NA INTERNET DO SOFTWARE FRONT PAGE EXPRESS.....	11
FIGURA 2 - FIGURA 2 - TELA DO SOFTWARE - CURSO HJ DE DATILOGRAFIA	12
FIGURA 3 - TELA SOFTWARE MEGALOGO – DESENHO CRIADO COM PROCEDIMENTOS E COMANDOS DA TARTARUGA.....	13
FIGURA 4 - TELA DO SIMULADOR : LABORATÓRIO INTERATIVO DE ELETROMAGNETISMO.....	14
FIGURA 5 - TELA JOGO CARMEM SANDIEGO	15
FIGURA 6 - TELA JOGO CARMEM SANDIEGO	15
FIGURA 7 - TELA DO JOGO GEO-3D	16
FIGURA 8 - MODELO DE CICLO DE VIDA PARA INTERFACE.....	18
FIGURA 9 - RELAÇÕES ENTRE OS DIAGRAMAS DA UML	29
FIGURA 10 - EXEMPLO DE CASOS DE USO	30
FIGURA 11 - EXEMPLO DE DIAGRAMA DE CLASSES	32
FIGURA 12 - EXEMPLO DE UM DIAGRAMA DE SEQÜÊNCIA DE EVENTOS.....	33
FIGURA 13 - TELA INICIAL DO <i>RATIONAL ROSE</i>	34
FIGURA 14 - TELA DO DIRECTOR 8.....	36
FIGURA 15 - A JANELA <i>STAGE</i>	37
FIGURA 16 - A JANELA <i>CAST</i>	37
FIGURA 17 - A JANELA <i>SCORE</i>	38
FIGURA 18 - A JANELA <i>CONTROL PANEL</i>	39
FIGURA 19 - ESTRUTURA DO JOGO	43
FIGURA 20 - DIAGRAMA DE CASO DE USO: ALUNO	44
FIGURA 21 - DIAGRAMA DE CASO DE USO: PROFESSOR.....	45
FIGURA 22 - DIAGRAMA DE CLASSES JOGO MOUSE	46
FIGURA 23 - DIAGRAMA DE CLASSES: JOGO INTERNET.....	47
FIGURA 24 - DIAGRAMA DE INTERAÇÃO: JOGO INTERNET.....	49
FIGURA 25 – DIAGRAMA DE INTERAÇÃO: JOGO MOUSE	50
FIGURA 26 – TELA INÍCIO DO JOGO.....	51
FIGURA 27 - OBJETOS “CAINDO” NA TELA DO JOGO.....	52
FIGURA 28 – PASSANDO PARA O 2º JOGO DO 1º MÓDULO: CLIQUE BOTÃO DIREITO DO MOUSE	55

FIGURA 29 - TELA JOGO CLICAR E MOVER OBJETOS.....	56
FIGURA 30 - TELA JOGO INTERNET: DIGITAÇÃO DE ENDEREÇOS PARA ACESSAR OUTRAS PÁGINAS.....	58
FIGURA 31 - TELA JOGO INTERNET: BAIRROS	59
FIGURA 32 – TELA JOGO INTERNET: CASAS	59

LISTA DE QUADROS

QUADRO 1 - DETALHANDO CASOS DE USO DO DIAGRAMA DE CASOS DE USO: ALUNO	45
QUADRO 2 - CÓDIGO DE IMPLEMENTAÇÃO PARA ADICIONAR OBJETOS A LISTA.....	52
QUADRO 3 - SOMA DOS OBJETOS CLICADOS	53
QUADRO 4 – RETIRANDO OBJETOS DA TELA E ADICIONANDO-OS A LISTA DE SPRITES	53
QUADRO 5 - CÓDIGO PARA VERIFICAR MUDANÇA DE FASE E JOGO.....	54
QUADRO 6 – LISTA COM AS FASES E QUANTIDADE DE OBJETOS	55
QUADRO 7 – CRIAÇÃO DA LISTA DE EVENTOS DO MOUSE.....	56
QUADRO 8 - VERIFICANDO ENCAIXE DOS OBJETOS NAS CAIXAS	57
QUADRO 9 - VERIFICAÇÃO DA DIGITAÇÃO CORRETA DOS ENDEREÇOS...58	
QUADRO 10 - CÓDIGO DE IMPLEMENTAÇÃO JOGO INTERNET CASAS	60

LISTA DE TABELAS

TABELA 1 - DIAGRAMAS UML.....	28
TABELA 2 - TERMINOLOGIA UTILIZADA.....	40
TABELA 3 - SINTAXE PARA A CRIAÇÃO DE LISTAS.....	42

RESUMO

Este trabalho de conclusão de curso apresenta o desenvolvimento de um software na forma de um jogo para auxiliar no processo de ensino-aprendizagem da informática a pessoas da terceira idade. O software foi desenvolvido com o intuito de atender as necessidades relatadas pelos alunos em atividades desenvolvidas no ensino da informática, especificamente a utilização do mouse e a “navegação” na internet. Para a realização do trabalho foram utilizados o software de autoria Macromedia Director 8 e a ferramenta Rational Rose. A efetividade do software desenvolvido na facilitação da aprendizagem do manuseio do mouse e de um navegador da Internet foi avaliada através de sua utilização por cinco usuários em potencial.

ABSTRACT

This course conclusion research presents the development of a software with the configuration of a game, to aid in the teaching-learning process of the computer science to people of the third age. The software was developed with the intention of assisting the needs related by the students, in the activities developed in the teaching of the computer science, specifically the use of the mouse and the “navigation” in the Internet. For the accomplishment of the work it were used the authorship software Macromedia Director 8 and the tool Rational Rose. The effectiveness of the software was developed in the facilitation of the learning of the mouse handling and of an Internet navigator. It was evaluated through the application of five potential users.

1 INTRODUÇÃO

O início desse novo século tem se caracterizado por uma expressiva difusão da informática por toda a sociedade. Na área empresarial, a utilização da informática deixou de ser um diferencial competitivo para se tornar uma necessidade para a sobrevivência. Na educação, o número, qualidade e diversidade de softwares educacionais tem aumentado exponencialmente e a quantidade de escolas brasileiras que possuem computadores para o ensino, com acesso à internet, também tem crescido.

A década de 90 do século passado foi testemunha de uma expressiva ampliação do uso de computadores nas residências, no mundo do lazer, na cultura, na medicina, na arquitetura, no trabalho. As novas gerações já estão totalmente socializadas com a informática, através de sua utilização nas escolas, nas pesquisas científicas etc. (GOLDMAN, 2001).

Dentro dessa realidade, o domínio das novas tecnologias da informação e da comunicação é requisito fundamental para que os profissionais das diversas áreas tenham maiores oportunidades. No entanto, não é só na área profissional que as novas tecnologias vêm criando oportunidades e novas formas de comunicação e interação. No lazer, já é bastante forte a necessidade de se conhecer as ferramentas fornecidas pela informática.

O que se observa, no entanto, é que a grande maioria da população ainda não tem acesso a essas possibilidades. Até o final de 2002, estima-se que cerca de 3 milhões de brasileiros estarão conectados à internet. Apesar desse número parecer grande, ele é muito menor do que o número de pessoas que possuem uma linha telefônica (49 milhões) e corresponde a menos de 2% da população brasileira (BRASIL, 2001).

Evidentemente que a principal causa desse pequeno número de usuário é econômica. No entanto, existe uma grande quantidade de pessoas que mesmo possuindo capacidade econômica para adquirir um computador, não o faz. Essa resistência na utilização dos computadores decorre do fato de que a adoção de uma nova tecnologia exige uma mudança de paradigma, o que, em muitos casos, é muito difícil de ser feita.

Nesse contexto, os instrumentos que permitem o acesso aos recursos básicos oferecidos pela informática assumem uma importância muito grande, uma vez que permitem

uma “migração” para o novo paradigma de forma menos traumática. Dentre esses recursos destacam-se os softwares educacionais, ou seja, softwares que, geralmente, possuem recursos multimídia e permitem o aprendizado de ferramentas informatizadas.

Apesar da grande quantidade de softwares disponível para o aprendizado da informática, o público-alvo principal são crianças e jovens. A quantidade de softwares direcionados à terceira idade ainda é muito pequena. Isso ocorre porque, conforme ressalta Goldman (2001), para os mais maduros, o uso do computador parece uma barreira difícil de transpor, mesmo que a cada dia, eles sejam solicitados a responder às demandas deste mundo digital.

É dentro desse contexto que se enquadra o presente trabalho, o qual propõe o desenvolvimento de um software educacional, do tipo jogo, para auxiliar pessoas da terceira idade no manuseio do mouse e navegação na internet.

1.1 OBJETIVOS DO TRABALHO

A presente proposta de trabalho tem como objetivo o desenvolvimento de um software educacional, na forma de jogo, que auxiliará o aprendizado no uso do *mouse* e em um navegador da internet por pessoas da terceira idade.

Além desse objetivo, o trabalho possui também os seguintes objetivos específicos:

- a) possibilitar o aprendizado no uso do *mouse*, o que inclui o clique, clique duplo e arrastar-soltar;
- b) permitir o aprendizado dos recursos básicos de um navegador da internet, o que inclui a digitação de endereços, navegação através de cliques em botões e *links*;
- c) monitorar o desempenho do aluno em cada um dos módulos e jogos incluídos no software;
- d) análise/teste da efetividade do software produzido, junto ao público alvo.

1.2 JUSTIFICATIVA

O avanço da informática nos últimos anos tem modificado o panorama em diversos campos da atividade humana. O cenário para um futuro próximo apresenta perspectivas de amplas e profundas modificações em praticamente todos os setores da sociedade. A Internet, especialmente, apresenta possibilidades extraordinárias de mudanças no paradigma da sociedade, onde pessoas da terceira idade começam a participar cada vez mais do acesso a esta tecnologia dentro de suas casas.

Segundo Halfoun (2002), 629 milhões de pessoas com mais de 60 anos já constituem 10% do total da população. Esse número deve dobrar até 2050, o que significa dizer que dentro de no máximo 50 anos o número de idosos em todo o mundo pode ultrapassar o de crianças e adolescentes. Os dados são de um levantamento feito pela Divisão Populacional da ONU (Organização das Nações Unidas). A ONU está convencida de que até 2050 a população mundial de idosos deve chegar a dois bilhões de pessoas.

Sendo assim, conforme Tessari (2001), para as novas gerações do terceiro milênio não foi preciso a adaptação à informática; mas e os mais velhos, pais, avós, bisavós? O fato é que, não se pode pensar que as novas ferramentas são exclusivas dos jovens. Ao contrário, nunca é tarde para experimentar, conhecer e aprender. Pelo exposto, observa-se que a informática traz inúmeros benefícios aos mais velhos: descoberta e reaproximação de amigos e parentes, ampliação de novos horizontes, reencontro do gosto pela vida, além de ser excelente exercício mental, pois há sempre uma coisa nova a se aprender.

Com a introdução da informática no cotidiano e dentro das casas das pessoas, como também, a evolução da internet, despertou a curiosidade dos idosos ao verem seus netos “brincando” no computador, e com isso, surgiu uma nova necessidade: adaptar o treinamento de informática a pessoas da terceira idade. Iniciou-se, então, uma pesquisa informal com usuários da terceira idade, que resultou na constatação de que essas pessoas encontram barreiras para o aprendizado da informática por receio ou medo de não conseguirem aprender. Uma das maiores dificuldades para a utilização do computador é o manuseio do *mouse*, devido à falta de coordenação motora ou até mesmo de não conseguirem associar o movimento do *mouse* com o ponteiro na tela.

O grande interesse das pessoas da terceira idade é a utilização da internet para se comunicar com outras pessoas. Em geral, esse aprendizado é realizado através do uso de tutoriais ou cursos presenciais. No entanto, as dificuldades e interesses do referido público são muito específicos.

Ao invés da apresentação linear dos recursos, como ocorreria em um tutorial, os jogos irão desenvolver a capacidade do aluno em utilizar o mouse, clicar sobre botões, utilizar teclas de atalho e menus. Com isso, espera-se que sejam desenvolvidas as habilidades cognitivas necessárias ao uso do mouse e à navegação na internet.

1.3 LIMITAÇÕES

O desenvolvimento do software limitar-se-á ao ensino do uso do mouse, além de apresentar os recursos mais utilizados na navegação na internet.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está dividido em sete capítulos.

O primeiro capítulo trata dos objetivos do trabalho, justificativa e sua estrutura.

O segundo capítulo faz uma abordagem sobre Software Educacional. Aborda o histórico, a evolução dos jogos educacionais e os tipos de softwares educacionais.

O terceiro capítulo trata da Informática na Terceira Idade. É comentado sobre as limitações que as pessoas da terceira idade possuem para aprender informática e sobre o interesse delas pela informática na atualidade.

No quarto capítulo, faz-se uma análise sobre o paradigma de orientação a objetos focando conceitos básicos, metodologia de desenvolvimento, principais diagramas e documentos.

O quinto capítulo faz uma breve abordagem da ferramenta CASE Rational Rose utilizada para especificação do software propriamente dito. Adicionalmente é feita uma abordagem geral sobre o ambiente de desenvolvimento utilizado: o Director 8.

O sexto capítulo trata do software desenvolvido, detalhando a especificação do sistema, seu processo de desenvolvimento e algumas telas representativas do software.

No sétimo capítulo, são apresentadas a conclusão do trabalho e algumas sugestões para futuros trabalhos.

2 SOFTWARES EDUCACIONAIS

Software Educacional é geralmente classificado como subconjunto de sistemas de treinamento. Existe claramente uma interseção entre os dois conjuntos, mas a maior parte dos softwares educacionais não pode ser exatamente descrita como sistemas de treinamento, e vice-versa (FERREIRA, 1995).

O Software Educacional vem entrando no mercado mundial de forma muito acelerada. Inúmeros países como Inglaterra, França e EUA, entre outros, desenvolvem projetos de uso de microcomputador em educação e, conseqüentemente necessitaram desenvolver produtos de software específicos para suas necessidades. O mesmo tem ocorrido no Brasil, onde diversos projetos de pesquisa vêm sendo desenvolvidos não só relacionados ao uso do microcomputador e 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 (CAMPOS, 1991).

O objetivo do presente capítulo é apresentar a evolução histórica da informática na educação, o conceito de multimídia, as etapas do desenvolvimento de um software educacional, qualidade de software educacional, além de descrever e exemplificar os principais tipos de softwares educacionais existentes.

2.1 HISTÓRIA DA INFORMÁTICA NA EDUCAÇÃO

Desde a década de 70, a educação no Brasil tenta alcançar o *status* de tratamento científico com o uso de tecnologia instrucional. Muito se tentou, para realizar o ensino individualizado, onde cada aluno seguiria de acordo com seu próprio ritmo e, tivesse à sua disposição um *feedback* sobre as respostas desejadas (FAGUNDES, 1992).

No Brasil, o primeiro evento que tratou do tema Informática e Educação foi um seminário sobre a utilização de computadores no ensino de física, sob a assessoria de um especialista da Universidade de Dartmouth (EUA), realizado em 1981, na Universidade de São Carlos, São Paulo. No entanto, ações de abrangência nacional só ocorreram a partir de

1981, com a realização do 1º Seminário Nacional de Informática na Educação, em Brasília (PEIXOTO, 1995).

O Projeto LOGO (descrito no tópico 2.3 deste capítulo) foi operacionalizado a partir de 1978, ano em que, de fato, a informática chegou a escola pública brasileira, em Campinas. O interesse despertado pelo Projeto LOGO fez com que a UNICAMP criasse em 1983, o Núcleo Interdisciplinar de Informática Aplicada a Educação – NIED (PEIXOTO, 1995).

A realização de pesquisas sobre o ensino por computador aparece como resultado dos subprojetos financiados pelo Projeto EDUCOM. Criado em 1983 pela Secretaria Especial de Informática da Presidência da República e pelo Ministério da Educação, sua implantação definitiva ocorreu apenas em 1985, com a instalação de cinco centros de pesquisa em universidades: Universidade Federal de Pernambuco, Universidade Federal de Minas Gerais, Universidade Federal do Rio de Janeiro, Universidade de Campinas e Universidade Federal do Rio Grande do Sul (PEIXOTO, 1995).

O Projeto EDUCOM ¹ originou o Programa Nacional de Informática Educativa – PRONINFE, lançado em 1989, para apoiar o desenvolvimento e a utilização de novas tecnologias de informática no ensino fundamental, médio e superior e na educação especial. Tanto o EDUCOM quanto o PRONINFE não chegaram à escola de ensino básico, permanecendo no campo experimental em universidades, secretarias de educação e escolas técnicas (MEC, 2001).

A divulgação dos resultados de pesquisas e a valorização do uso do computador como sinal de modernidade pelas escolas criam condições para o aparecimento de novos termos relativos ao ensino por computador (PEIXOTO, 1995).

2.2 EVOLUÇÃO DOS SOFTWARES EDUCACIONAIS

Muitos softwares disponíveis no mercado que são ditos educacionais, não apresentam relevância pedagógica. Possuem, muitas vezes, apenas uma interface agradável,

¹ Nome dado ao projeto criado em 1983 pela Secretaria Especial de Informática (SEI) visando a implantação de centros-piloto em universidades públicas, voltados à pesquisa no uso de informática educacional.

colorida, mas são pobres no seu conteúdo. Além disso, exploram, na maioria das vezes, a repetição de exercícios, caracterizando os chamados CAI (Computer Aided Instruction) - Instrução Auxiliada por Computador, que são classificados como os mais simples softwares educacionais. Um dos tipos de software educacional que tenta fugir à monotonia dos CAI's são os jogos educativos computadorizados. Através de um jogo, o aluno pode “aprender” pelo conteúdo embutido no jogo. Os jogos educativos computadorizados são preferidos pelos alunos por despertar neles o interesse em “brincar”, ou seja, o aluno “aprende brincando” (SILVEIRA, 2002).

A evolução da informática, trouxe ao alcance das pessoas uma nova tecnologia: o recurso multimídia. Classicamente falando, multimídia computacional é a união de várias mídias, com o intuito de fixar ou transmitir mais facilmente uma informação (FERREIRA, 1995). O Software Multimídia pode ser considerado também como um "livro eletrônico", pois, contém textos, imagens, sons, animações e filmes interligados.

A utilização da multimídia no desenvolvimento de software educacional trouxe inúmeras vantagens. Um software educacional com recursos multimídia pode tornar o aprendizado mais agradável e interessante, devido à possibilidade da inclusão de sons, fotos, imagens e animações, entre outras mídias. Este tipo de software pode contribuir no processo de ensino e aprendizagem, tornando as aulas menos monótonas e despertando no aluno o interesse à investigação e à descoberta. Um software com recursos multimídia também pode ser elaborado para que o aluno “aprenda brincando”. Isto se torna possível através da utilização de jogos educativos. Estes jogos educativos podem utilizar-se de recursos multimídia e tornam o aprendizado muito mais interessante e divertido. Estes jogos são elaborados para divertir os alunos e aumentar a chance de aprendizagem de conceitos, conteúdo e habilidades embutidas no jogo (FERREIRA, 1995).

2.3 TIPOS DE SOFTWARES EDUCACIONAIS

Os tipos mais comuns de softwares educacionais ou, também chamados de ambientes de aprendizagem interativa são: tutorial, exercício e prática, ambientes de autoria, simulação e jogos. Nos itens a seguir, cada um dos tipos de software são apresentados em maiores detalhes.

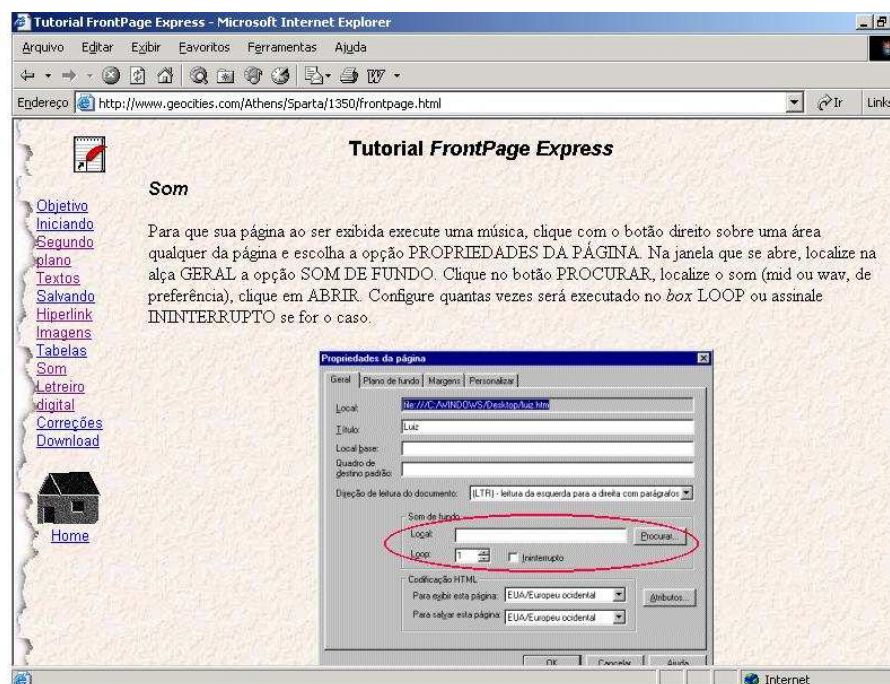
2.3.1 TUTORIAL

Os programas tutoriais, conforme ressalta Valente (1993), constituem uma versão computacional da instrução programada. A vantagem dos tutoriais é o fato de o computador poder apresentar o material com outras características que não são permitidas no papel como: animação, som e a manutenção do controle da performance do aprendiz, facilitando o processo de administração das lições.

A informação que está disponível para o aluno é definida e organizada previamente, assim o computador assume o papel de uma máquina de ensinar. A interação entre o aprendiz e o computador consiste na leitura da tela ou escuta da informação fornecida e avanço pelo material, apertando a tecla ENTER ou usando o *mouse* para escolher a informação.

Os tutoriais só permitem ao professor verificar o produto final e não os processos utilizados para alcançá-lo. A sua limitação se encontra justamente no fato de não possibilitar verificar se a informação processada passou a ser conhecimento agregado aos esquemas mentais. Além disso, em muitos casos, o tutorial não consegue manter a motivação e o interesse do aluno. A Figura 1 mostra exemplo de um tutorial.

FIGURA 1 - TUTORIAL NA INTERNET DO SOFTWARE FRONTPAGE EXPRESS



Fonte: NEITZEL(2001)

2.3.2 EXERCÍCIO E PRÁTICA

Os softwares educacionais de exercício e prática tem como característica ensinar através da repetição de exercícios propostos, fazendo com que o aluno aprenda praticando e visualizando seu desempenho no término de cada exercício. Um exemplo interessante é o software Curso HJ de Datilografia. Este software é produzido pela H&J Software, próprio para ensinar a datilografia no computador. O aluno treina a digitação com telas coloridas e imagens do teclado demonstrando a posição correta dos dedos nas teclas (Figura 2). No final de cada treino de digitação o software mostra o desempenho do aluno. Também neste software, existem jogos onde o aluno pratica a velocidade de digitação.

FIGURA 2 - TELA DO SOFTWARE - CURSO HJ DE DATILOGRAFIA



Fonte: H&J Software Com. Ltda

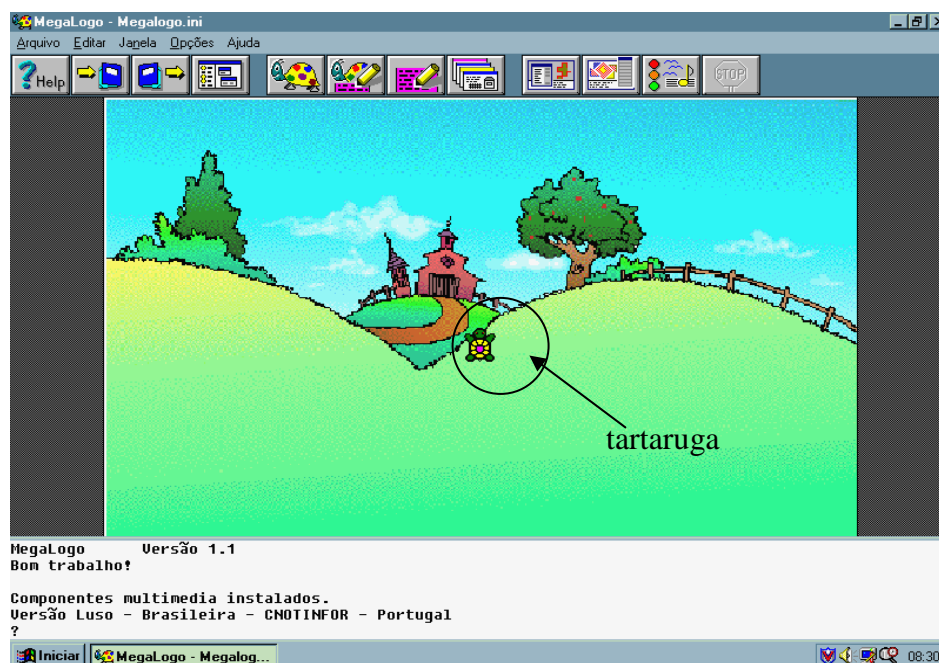
2.3.3 AMBIENTE DE AUTORIA

Um software do tipo autoria permite que o aluno crie seu próprio material, ou seja, o aluno é o autor e o construtor de seu próprio conhecimento. Um bom exemplo de software educacional do tipo autoria é o MEGALOGO (Figura 3). Logo é uma linguagem de programação proposta por Seymour Papert. A grande contribuição de Papert está na manipulação de um objeto gráfico, chamado “tartaruga”, que é capaz de andar pela tela deixando seu rastro. Ensinar a tartaruga a fazer algo (a figura de um quadrado ou uma casinha, por exemplo) é uma metáfora para a atividade de programar, no contexto da tartaruga. Dessa forma o computador é abstraído na figura da tartaruga. O resultado (rastro da

tartaruga), mostrado na tela fornece um *feedback* para a criança que pode levá-la a reformular o procedimento “ensinado” (VALENTE, 1993).

Programar a tartaruga de Logo é, portanto, um modelo “procedural” de programação, onde o procedimento que a criança cria para “ensinar” a tartaruga deve conter todos os passos que a tartaruga deve executar para conseguir o resultado desejado (VALENTE, 1993).

FIGURA 3 - TELA SOFTWARE MEGALOGO – DESENHO CRIADO COM PROCEDIMENTOS E COMANDOS DA TARTARUGA



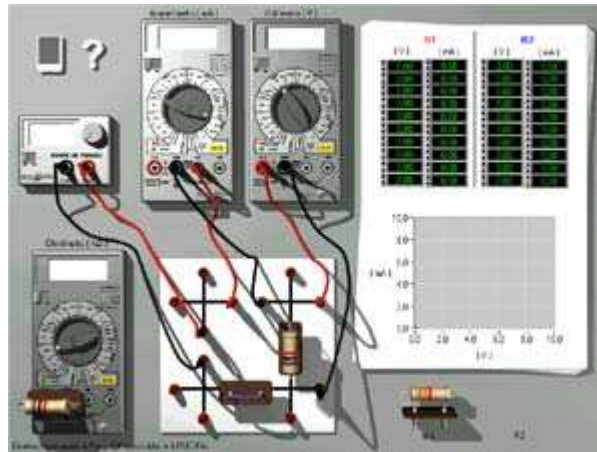
Fonte: Software Megalogo. Versão Luso-Brasileira - Portugal

2.3.4 SIMULAÇÃO

Os softwares educacionais do tipo simulação reproduzem situações reais para facilitar o aprendizado do aluno. Através de demonstrações que imitam a realidade, o aluno poderá aprender virtualmente. Um exemplo de software educacional de demonstração é o Laboratório Interativo de Eletromagnetismo (Figura 4). Esta multimídia tem por objetivo reproduzir, com total fidelidade, uma coletânea das principais experiências dos cursos de eletricidade, eletrodinâmica e magnetismo. Cada experiência vem acompanhada de um breve resumo teórico, de animações e de vídeos ilustrativos, bem como da descrição dos materiais e

das etapas do experimento. O usuário terá ao seu dispor os manuais de operação dos instrumentos utilizados nas experiências, assim como uma descrição dos materiais que as compõem (LABORCIÊNCIA, 1998).

FIGURA 4 - TELA DO SIMULADOR: LABORATÓRIO INTERATIVO DE ELETROMAGNETISMO



Fonte: LABORCIÊNCIA(1998)

2.3.5 JOGOS

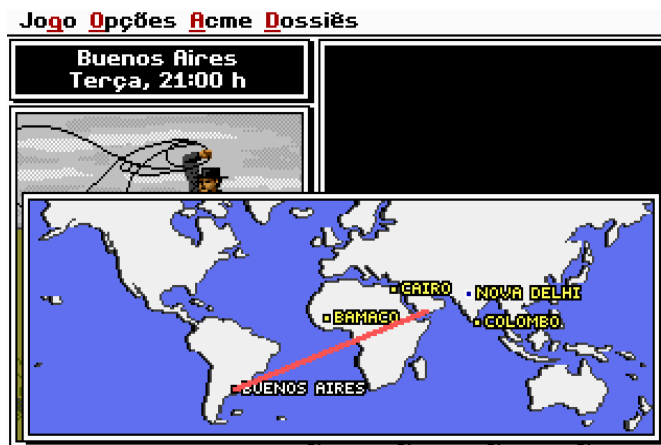
Os softwares educacionais tipo jogos educativos utilizam recursos multimídia e tornam o aprendizado muito mais interessante e divertido. Estes jogos são elaborados para divertir os alunos enquanto eles aprendem. Conforme Ferreira (1995), um dos primeiros a conseguir popularidade foi a série *Carmem Sandiego* (desenvolvido pela Broderbund), que pretendia ensinar conceitos de história e geografia a partir de uma “caçada” a uma ladra internacional e sua quadrilha ao redor do mundo. Nele as pistas eram resolvidas com aplicação de conhecimentos nessa área, despertando o interesse de crianças por essas matérias. Apesar de ser voltado para crianças, o jogo obteve grande popularidade entre os adultos e adolescentes. Exemplos de telas do jogo *Carmem Sandiego* podem ser vistos na Figura 5 e na Figura 6.

FIGURA 5 - TELA JOGO CARMEM SANDIEGO



Fonte: Broderbund Software. Jogo Carmem Sandiego 1992

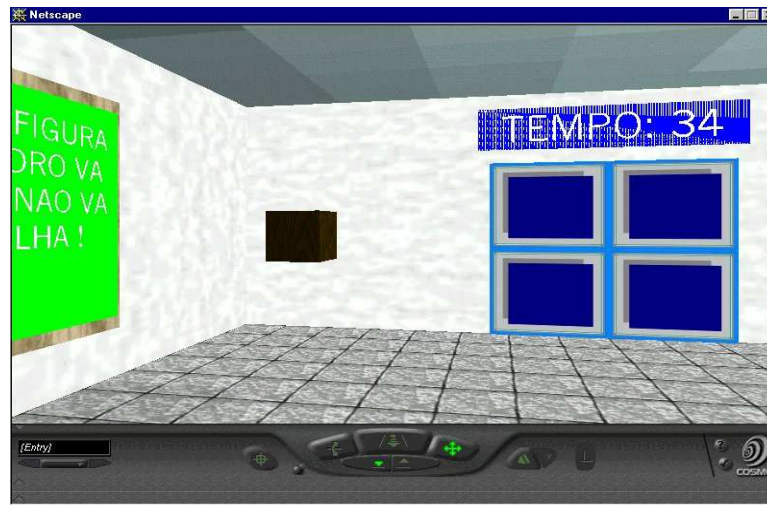
FIGURA 6 - TELA JOGO CARMEM SANDIEGO



Fonte: Broderbund Software. Jogo Carmem Sandiego 1992

Outro exemplo de software tipo Jogo Educacional é o GEO-3D (Figura 7). Construído em VRML, linguagem de Realidade Virtual, o jogo é um protótipo onde o aluno entra numa sala de aula virtual com um quadro e duas portas, sendo um vermelha e outra azul, o aluno deve ler a questão sobre geometria espacial que está no quadro e dependendo de sua resposta deverá seguir por uma das portas, sendo que só uma delas leva pelo caminho certo, seguindo-se assim até o final do jogo (WEBFOLIO, 2001)..

FIGURA 7 - TELA DO JOGO GEO-3D



Fonte: WEBFOLIO(2001)

2.4 DESENVOLVIMENTO DE SOFTWARE EDUCACIONAL

Segundo Rocha (1996), o método tradicional de desenvolvimento de software, no qual os programadores sozinhos completavam o projeto, já está ultrapassado. Hoje, deve-se lidar com métodos, procedimentos e ferramentas para aumentar a produtividade e qualidade dos produtos. Projetos de desenvolvimento de software educacional, além de envolver em seu desenvolvimento uma equipe multidisciplinar, devem refletir os objetivos educacionais propostos e o ambiente de aprendizagem almejado, criando situações que estimulem o desenvolvimento das habilidades desejadas.

Com base na proposta de Rocha (1996), a seguir apresenta-se recomendações, em forma de etapas, para o desenvolvimento de um software educacional.

2.4.1 DEFINIÇÃO DO AMBIENTE DE APRENDIZAGEM

Na literatura encontramos diversos autores que têm sugerido metodologias diferenciadas para o desenvolvimento de software educacional, em geral incorporando estratégias que contemplem o ambiente de aprendizagem almejado.

O desenvolvimento do software educacional possui características específicas e a especificação dos requisitos de qualidade inclui o modelo de ensino/aprendizagem selecionado, isto é, a filosofia de aprendizagem subjacente ao software. Este é o único padrão a ser especificado “a priori” no desenvolvimento do software educacional e que vai determinar seu desenvolvimento. A experiência tem mostrado que o processo de desenvolvimento de software adequado à multimídia educacional deve ser composto do modelo de ciclo de vida de prototipagem evolutiva, acrescido da etapa inicial da escolha do ambiente educacional e avaliação por parte de professores e alunos, para que novos requisitos sejam incorporados ao software.

2.4.2 ANÁLISE DE VIABILIDADE

Os projetos podem variar em função do objetivo pelo qual o sistema é constituído, do hardware sobre o qual pode ser implantado e também em função da filosofia de desenvolvimento. Para que o projeto do software educacional seja realizado é necessária a definição de algumas estimativas dentre as quais destacam-se os recursos, custos e cronogramas. Deve-se fornecer dados sobre os usuários, restrições externas, limitações do produto e outros fatores relevantes. A estimativa dos recursos necessários para o esforço de desenvolvimento inclui: recursos de hardware, software e recursos humanos (CAMPOS, 1996). É necessário avaliar a possibilidade do reuso de componentes e identificar, acompanhar e eliminar itens de risco antes que eles possam comprometer o sucesso do projeto ou que se tornem a principal fonte de trabalhos refeitos.

2.4.3 SELEÇÃO DO TIPO DE DOCUMENTO

Na prática das escolas o que se tem verificado é a utilização dos sistemas de multimídia para o desenvolvimento de hiperdocumentos por dois grupos distintos de usuários autores :professores e alunos.

Os hiperdocumentos desenvolvidos por estes dois grupos citados acima também podem ser analisados sob outros dois prismas (CAMPOS,1996):

- a) de um lado, temos os hiperdocumentos para serem utilizados por diversos usuários, que trazem em si uma base de conhecimentos sólida e consistente e que

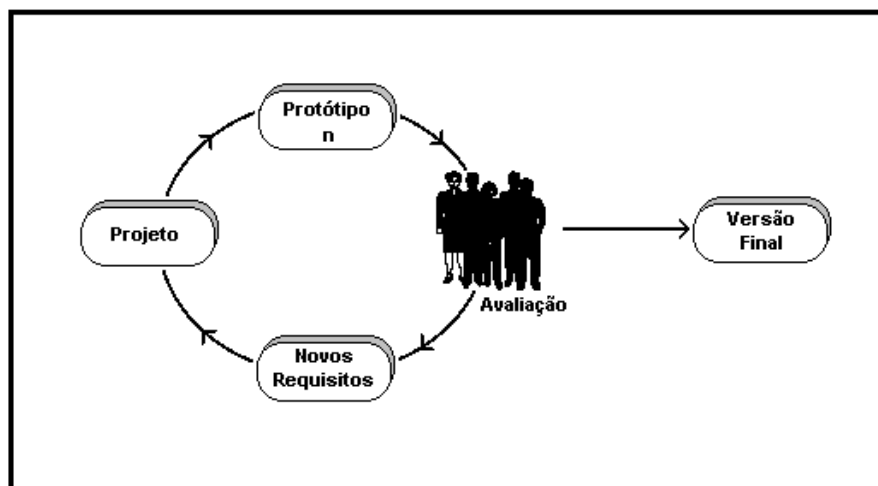
deverão ter uma vida útil, duradoura e incremental, devendo refletir um ambiente educacional rico e coeso com a prática pedagógica;

- b) de outro lado, existem produtos que não têm nenhum compromisso didático pedagógico, apenas exploratório.

2.4.4 PLANEJAMENTO DA INTERFACE

A interface do usuário é o mecanismo através do qual o diálogo entre o software e o ser humano é estabelecido. Os fatores humanos devem ser levados em consideração para que o diálogo seja ameno. Como o homem percebe o mundo através do sistema sensório, o planejamento de uma interface deve considerar os sentidos visuais, tácteis e auditivos. É importante notar os níveis de habilidades pessoais e as diferenças individuais entre os usuários(CAMPOS,1996). A Figura 8, apresenta um modelo de ciclo de vida de prototipagem evolutiva para a interface.

FIGURA 8 - MODELO DE CICLO DE VIDA PARA A INTERFACE.



Fonte: CAMPOS (1996)

2.4.5 PLANEJAMENTO DO DOCUMENTO

Segundo Makedon², apud Campos (1996), o material que irá compor a multimídia deve ser pesquisado, organizado, assimilado, escrito e produzido um script que, como uma peça de teatro orquestra a aparência e a ativação dos diversos componentes e mídias no momento desejado.

2.4.6 SELEÇÃO DO SISTEMA DE AUTORIA E DAS FERRAMENTAS

Para desenvolver o trabalho de autoria de um programa de multimídia são necessários ao menos um sistema de autoria, destinado ao desenvolvimento do programa propriamente dito e sistemas de apoio à autoria: pintura, desenho, ilustração, animação, titulação, diagramação, tratamento de figuras, etc.

O desenvolvimento de um software de qualidade requer a verificação da presença ou ausência de critérios de qualidade. Selecionar um sistema de autoria é uma etapa importante porque é neste momento que o usuário contemplará os requisitos e expectativas escolhendo a ferramenta correta para a aplicação.

Uma característica importante nos sistemas de autoria é a interatividade. É ela que coloca o usuário no controle do sistema, manipulando as diversas mídias nos diferentes modos de interação. É, ainda, a interatividade que permite o trabalho cooperativo de múltiplos autores.

2.4.7 CARACTERÍSTICAS PEDAGÓGICAS

As características pedagógicas formam um conjunto de atributos que evidenciam a conveniência e a viabilidade da utilização do software em situações educacionais (ROCHA, 1996):

² Makedon, Fillia et all. Issues and Obstacles with Multimedia Authoring. Proceedings of the EDMEDIA 94. Educational Multimedia and Hypermedia, 1994. 38-45.

- a) O software educacional deve permitir a identificação do ambiente educacional e do modelo de aprendizagem que ele privilegia;
- b) o software deve ser adequado e pertinente em relação ao contexto educacional ou a uma disciplina específica;
- c) o software deve contribuir para que o aluno alcance o objetivo educacional e para isso deve ser amigável e de fácil utilização, deve possuir aspectos motivacionais e respeitar individualidades. É importante que inclua atributos como: clareza e correção dos conteúdos, recursos motivacionais, carga informacional e tratamento de erros.

2.4.8 FACILIDADE DE USO

A facilidade de uso é o conjunto de atributos que evidenciam a facilidade de uso do software. Inclui as subcaracterísticas (ROCHA, 1996):

- a) Facilidade de aprendizado: avalia a facilidade dos usuários em aprender a usar o software;
- b) Facilidade de memorização: avalia a facilidade dos usuários em memorizar informações importantes para o uso do software;
- c) Robustez: avalia se o software mantém o processamento corretamente a despeito de ações inesperadas.

2.4.9 INTERFACE

As características da interface configuram atributos que evidenciam a existência de um conjunto de meios e recursos que facilitam a interação do usuário com o software. Inclui as subcaracterísticas (ROCHA, 1996):

- a) Condução: avalia os meios disponíveis para aconselhar, informar e conduzir o usuário na interação com o computador. Inclui atributos como: presteza, localização, *feedback* imediato e legibilidade;

- b) Afetividade: avalia se o software proporciona uma relação agradável com o aluno ao longo do processo de aprendizado;
- c) Consistência: avalia se a concepção da interface é conservada igual em contextos idênticos e se ela se altera em contextos diferentes;
- d) Significado de códigos e denominações: avalia a adequação entre objeto ou informação apresentado ou pedido e sua referência;
- e) Gestão de erros: avalia os mecanismos que permitem evitar ou reduzir a ocorrência de erros, e que favoreçam a correção quando eles ocorrem. Inclui os atributos: proteção contra erros, qualidade das mensagens de erro e correção dos erros e reversão fácil das ações.

2.4.10 ADAPTABILIDADE

A adaptabilidade é o conjunto de atributos que evidenciam a capacidade do software de se adaptar a necessidades e preferências do usuário e ao ambiente educacional selecionado. Inclui atributos como (ROCHA, 1996):

- a) Customização: avalia a facilidade da adaptação da interface para o uso de diferentes usuários;
- b) Adequação ao ambiente: avalia a facilidade de adequação do software ao modelo e aos objetivos educacionais adotados.

2.4.11 DOCUMENTAÇÃO

A documentação é o conjunto de atributos que evidenciam que a documentação para instalação e uso do software deve ser completa, consistente, legível e organizada. Segundo Rocha (1996), para ambientes e *sites* apoiados na Web é importante avaliar também a característica qualidade da informação, que inclui as subcaracterísticas: conteúdos corretos, fontes fidedignas, carga informacional compatível, pertinência, temas transversais, entre outros.

3 A INFORMÁTICA NA TERCEIRA IDADE

O ensino de informática na terceira idade iniciou-se devido à grande procura e interesse dessas pessoas a utilizar o computador como mais um meio de comunicação e pela curiosidade em aprender uma tecnologia que se tornou acessível a eles.

As pessoas da terceira idade, há alguns anos, não demonstravam interesse em aprender esta nova tecnologia, pois a informática era de maior acesso aos jovens, que estavam entrando no mercado de trabalho. Mas agora, com a informática totalmente presente, houve uma mudança no modo de pensar das pessoas, criando assim, uma grande procura das pessoas da terceira idade em adquirir conhecimento em informática. Segundo Campos (2001), a maioria sequer mexeu num computador, quanto mais acessou internet, mas se empenha em aprender e usufruir, ao máximo, da novidade, “quando eles se sentam em frente ao computador têm vontade de aprender tudo e ficamos impressionadas com a persistência deles”, conta a monitora Roberta Polardi (Monitora do Programa Internet Sênior, criado através de uma parceria entre o Governo do Estado de Minas Gerais e a Prefeitura de Juiz de Fora). Segundo Nascimento (2000), não é fácil acostumar-se com a presença desse pequeno universo virtual, em que símbolo e figura são uma constante, em que há a predominância de algo que não se pega.

A informática para pessoas da terceira idade abre um novo horizonte de pesquisa para o ensino-aprendizado. O primeiro passo para isso é conhecer a percepção das pessoas de terceira idade sobre as condições que facilitam e as que dificultam a aprendizagem da utilização do computador (MATURIDADE, 2000).

O famoso neurologista e escritor norte-americano Robert Sapolski diz (em entrevista ao jornal Maturidade) que “as pessoas envelhecem intelectualmente quando fecham ciclos de aprendizado em sua vida, ou se contentam só com as coisas aprendidas até aquele momento”. Para quem não deseja esse tipo de envelhecimento, a Internet é uma grande oportunidade. Às vezes, a inibição em freqüentar um curso de informática ao lado de alunos muito mais jovens, pode também barrar a entrada para esse mundo de entretenimento, cultura e utilidades. Futuramente, "navegar" na Internet já não exigirá mais o computador. O telefone celular e a televisão tornar-se-ão também outros meios de acesso (MATURIDADE, 2000).

A seguir, algumas considerações segundo Abdala (2001):

Usar o computador é uma questão de sobrevivência. Mas se as crianças de hoje já nascem sabendo usar o *mouse*, são ágeis no teclado e entendem a “linguagem de Bill Gates”, o grupo que sempre trabalhou com máquina Olivetti não tem tanta desenvoltura. Como não podem parar no tempo, o jeito é voltar às salas de aula, e aprender como se usa o computador.

O Museu da República, do Rio de Janeiro, por exemplo, criou, há seis anos, o curso de informática para a terceira idade. A procura aumenta a cada ano. O Museu aposta em uma abordagem diferenciada: "Nós usamos um método de ensino especial para a terceira idade. Tentamos aproximar a linguagem do curso com o jargão específico da profissão exercida pelo aluno. Se a pessoa foi dona-de-casa, usamos metáforas como receitas culinárias para ensinar os comandos do computador. Se o aluno foi advogado, tentamos conciliar a linguagem do Direito com o que ele vai aprender. Isso facilita o aprendizado", conta Luis Cláudio Pereira, coordenador do curso (Depoimento do Coordenador do curso de Informática para a terceira idade criado pelo Museu da República da cidade do Rio de Janeiro, em 1995). Além disso, o Laboratório de Informática do museu, responsável pelo curso, ainda faz uso de um outro recurso para facilitar o aprendizado. "Sempre abrimos vagas para a monitoria voluntária de turmas para os que já terminaram o curso. Assim, eles ajudam os outros alunos e acabam servindo de exemplo para eles". Mas, será mesmo necessário um método de ensino diferenciado para jovens e idosos? "A gente não gosta de comparar, mas existe, no aprendizado, uma diferença entre as faixas etárias. Pessoas mais velhas têm medo de tudo o que é novo, e o computador ainda é uma coisa nova. No entanto, esse medo vai desaparecendo conforme vão conhecendo outras pessoas que já passaram pelo curso". Maria Helena Silveira, de 66 anos, já fez cursos básicos como Windows e Word, e, agora, voltou para fazer um curso de editoração gráfica. "É um curso muito legal; os professores sabem ensinar. Eu até comprei um notebook, depois que eu terminei os cursos básicos", elogia.

Pelo menos uma hora por dia. Esse é o tempo que as amigas Maria Carmelita Hoover, Nair Arruda Bento e Yedda Maria Fernandes dedicam diariamente ao computador. Atividades preferidas? Trocar e-mail e visitar *sites* de entretenimento e notícias internacionais. Seriam histórias triviais de gente que não foge em nada ao comportamento padrão do internauta brasileiro, não fosse por um detalhe: a idade. As amigas têm mais de 60 anos e fazem parte de um grupo cada vez maior de pessoas que já entraram na chamada boa

idade, mas não se renderam à inércia nem às convenções sociais e decidiram fazer parte da era digital (COMMERCIO, 2001).

O depoimento de um dos alunos do curso é emblemático:

“Renunciar à máquina elétrica não me foi fácil. Sentar-se diante do monitor e, mediante o digitar de uma tecla ou o clique no mouse, dar ordens ao microcomputador, foi tarefa um tanto árdua. Mas alguns meses depois, tornando-me amigo íntimo de barras de ferramenta, teclas, mouse, ícones, barras de menu, barras de rolagem, mergulhei no mundo do computador, considerando inicialmente inatingível” (NASCIMENTO, 2000).

3.1 POR QUE ENSINAR INFORMÁTICA ATRAVÉS DE UM JOGO ÀS PESSOAS DA TERCEIRA IDADE?

Os jogos devem estimular a imaginação das pessoas da terceira idade, auxiliar no processo de ensino-aprendizagem, facilitar a construção do conhecimento e auxiliar na aquisição da auto-estima. Devem também promover a criatividade, desenvolver a autonomia. Um jogo educativo computadorizado pode propiciar ao aluno da terceira idade um ambiente de aprendizagem rico. Este jogo deve ser ao mesmo tempo atrativo e eficaz no ensino-aprendizagem da informática a pessoas da terceira idade.

Segundo Nascimento (2000), os jogos de computador podem ser ferramentas auxiliares de aprendizagem para pessoas idosas. McCloskey (1983) sugere a utilização do computador como uma ferramenta lúdica, afirmando que algumas experiências com jogos de computador podem ser extremamente úteis na aprendizagem. Com efeito, algumas das melhores simulações têm o caráter de jogo, o que aumenta o seu potencial de utilização pedagógica. Os jogos permitem uma grande variedade de situações e uma exploração flexível dessas situações pelo jogador (num computador, essa resposta é rápida e individualizada).

Os jogos educativos computadorizados tornaram-se intrinsecamente motivadores, devido ao desafio, a fantasia e a curiosidade que despertam. Os jogos podem fornecer resultados não previstos, mas tão importantes quanto os previamente determinados, oferecendo oportunidades para que o aluno utilize a lógica, raciocínio e habilidades de

organização para resolver problemas de maneira mais interessante do que seriam expostos em um exercício comum. Os jogos são ambientes de aprendizagem que não fornecem o conteúdo diretamente ao aluno, mas incentivam a descoberta, a busca e o raciocínio (SILVEIRA, 1998).

Um jogo educativo não ensina explicitamente; nele aprende-se partindo da vivência lúdica e da reflexão sobre a mesma. O software proposto neste trabalho aproveita-se das vantagens da utilização da multimídia para auxiliar na criação de um jogo educativo que será utilizado no processo de ensino e aprendizagem da informática para pessoas da terceira idade.

4 ORIENTAÇÃO A OBJETOS

4.1 CONCEITOS BÁSICOS

Unified Modeling Language (UML) é um método de análise e projeto orientado a objetos (OOA & D) que surgiu no início dos anos noventa. A UML unifica os métodos de Booch, Rumbaugh (OMT) e Jacobson, mas o seu alcance é bem maior. Ela passou por um processo de padronização pela Object Management Group (OMG) e é agora um padrão OMG (FOWLER, 2000).

A UML é, na verdade, uma linguagem de modelagem e não um método. A maioria dos métodos consiste, pelo menos em princípio, de uma linguagem de modelagem e de um processo. A **linguagem de modelagem** é a notação (principalmente gráfica) utilizada por métodos para expressar projetos. O **processo** é a sugestão de quais passos a serem seguidos na elaboração de um projeto (FOWLER, 2000).

Segundo Deboni (2000), não se encontra na UML a descrição de passos que se deve seguir para se desenvolver um sistema, nem mesmo quais são as etapas para se modelar um sistema. A UML se limita, exclusivamente, a representar um sistema através de um conjunto de diagramas, onde cada diagrama se refere a uma visão parcial do sistema, que em conjunto forma um todo integrado e coerente.

4.2 USO DA UML

A UML é usada no desenvolvimento dos mais diversos tipos de sistemas. Ela abrange sempre qualquer característica de um sistema em um de seus diagramas e é também aplicada em diferentes fases do desenvolvimento de um sistema, desde a especificação da análise de requisitos até a finalização com a fase de testes (BARROS, 1998).

O objetivo da UML é descrever qualquer tipo de sistema, em termos de diagramas orientado a objetos. Naturalmente, o uso mais comum é para criar modelos de sistemas de software, mas a UML também é usada para representar sistemas mecânicos sem nenhum software. Aqui estão alguns tipos diferentes de sistemas com suas características mais comuns (BARROS, 1998):

- a) Sistemas de Informação: Armazenar, pesquisar, editar e mostrar informações para os usuários. Manter grandes quantidades de dados com relacionamentos complexos, que são guardados em bancos de dados relacionais ou orientados a objetos;
- b) Sistemas Técnicos: Manter e controlar equipamentos técnicos como de telecomunicações, equipamentos militares ou processos industriais. Eles devem possuir interfaces especiais do equipamento e menos programação de software de que os sistemas de informação. Sistemas Técnicos são geralmente sistemas real-time;
- c) Sistemas Real-time Integrados: Executados em simples peças de hardware integrados a telefones celulares, carros, alarmes etc. Estes sistemas implementam programação de baixo nível e requerem suporte real-time;
- d) Sistemas Distribuídos: Distribuídos em máquinas onde os dados são transferidos facilmente de uma máquina para outra. Eles requerem mecanismos de comunicação sincronizados para garantir a integridade dos dados e geralmente são construídos em mecanismos de objetos como CORBA, COM/DCOM ou Java Beans/RMI;
- e) Sistemas de Software: Definem uma infra-estrutura técnica que outros softwares utilizam. Sistemas Operacionais, bancos de dados, e ações de usuários que executam ações de baixo nível no hardware, ao mesmo tempo em que disponibilizam interfaces genéricas de uso de outros softwares;
- f) Sistemas de Negócios: descreve os objetivos, especificações (pessoas, computadores etc.), as regras (leis, estratégias de negócios etc.), e o atual trabalho desempenhado nos processos do negócio.

É importante perceber que a maioria dos sistemas não possuem apenas uma destas características acima relacionadas, mas várias delas ao mesmo tempo. Sistemas de informações de hoje, por exemplo, podem ter tanto características distribuídas como real-time. E a UML suporta modelagens de todos estes tipos de sistemas (BARROS, 1998).

4.3 DIFERENTES VISÕES DO SISTEMA

A UML permite ao analista representar o sistema seguindo diferentes visões. Cada visão possui uma grande dependência das outras visões, garantindo assim a coerência e completeza do modelo, e conseqüentemente do sistema de software. A relação entre os diagramas deve ser garantida pelo projetista, mas não é assegurada pela UML. Os diagramas podem ser agrupados segundo aspectos da visão que proporcionam (Tabela 1).

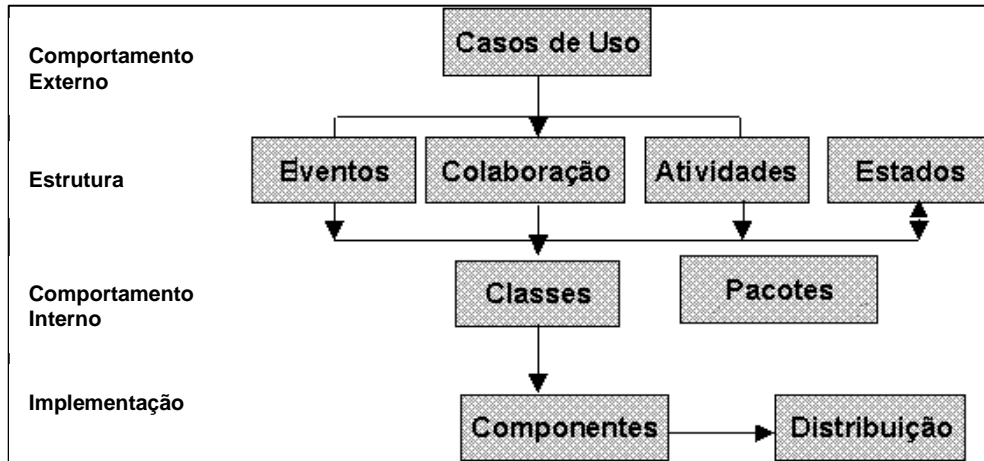
TABELA 1.: DIAGRAMAS UML

Diagrama de Comportamento Externo	Diagrama de Casos de Uso : dão uma visão externa do sistema e dos objetivos que os atores externos tem do sistema.
Diagramas de Comportamento Interno	Diagrama de Pacotes, Diagrama de Classes: que tratam dos processos que ocorrem entre as estruturas que compõem o sistema e dão uma visão da dinâmica interna do sistema.
Diagramas estruturais	Diagrama de Colaboração, Diagrama de Estados, Diagrama de Seqüência de Evento: dão uma visão estática da estrutura de suporte do sistema, sobre a qual ele será construído.
Diagramas de Implementação	Diagrama de Distribuição, Diagrama de Componentes: descrevem como estas estruturas são implementadas em software e hardware.

Fonte: DEBONI (2000)

A Figura 9 mostra como os diagramas podem se integrar para descrever um sistema. Apesar de não propor um método, existe na UML uma interação forte entre os diagramas como mostra as setas da Figura 9. Essa figura não faz parte da UML, mas pode ajudar a navegação entre os diagramas (DEBONI, 2000).

FIGURA 9 - RELAÇÕES ENTRE OS DIAGRAMAS DA UML



Fonte: DEBONI (2000)

Para este trabalho de conclusão de curso, a análise do software foi feita com: Diagramas de Caso de Uso, Diagramas de Classes e Diagrama de Sequência de Eventos (Análise detalhada no capítulo 6). Portanto a seguir é feita uma explicação sobre o uso desses Diagramas da linguagem de modelagem UML.

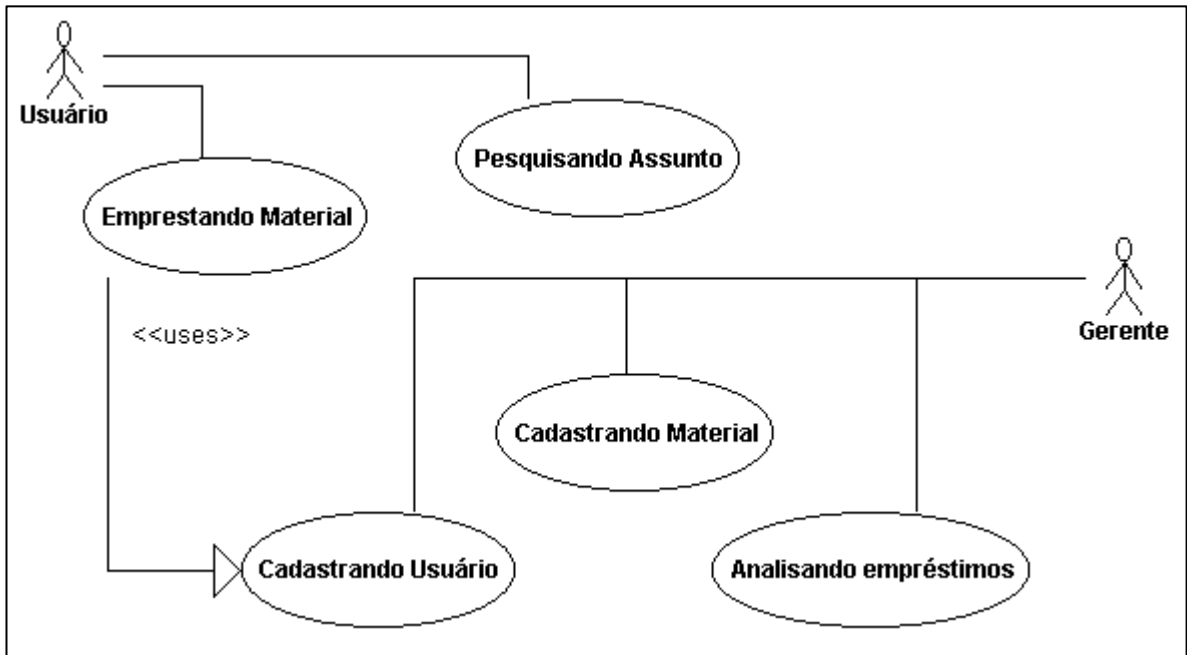
4.4 DIAGRAMA DE CASOS DE USO

Um caso de uso descreve um objetivo que um ator externo ao sistema tem com o sistema. Um ator pode ser um elemento humano ou não que interage com o sistema. O ator se encontra fora do escopo de atuação do sistema, enquanto o conjunto de casos de uso formam o escopo do sistema. A linha que separa os atores dos casos de uso é a fronteira do sistema (DEBONI, 2000).

O diagrama de casos de uso (Figura 10) representa graficamente esta interação, e define o contexto do sistema. Os atores são representados por representações simplificadas de uma figura humana, enquanto os casos de uso são elipses contendo cada uma o nome de um caso de uso. Os atores se comunicam com os casos de uso, que é representado por uma linha

unindo os dois elementos. Uma seta pode, opcionalmente, representar o fluxo principal de informação nesta interação e ajudar a leitura do caso de uso (DEBONI, 2000).

FIGURA 10 - EXEMPLO DE CASOS DE USO



Fonte: DEBONI (2000)

Como os casos de uso representam um objetivo do ator é comum dar como nome aos casos de uso, frases verbais curtas no infinitivo (EmprestarMaterial) ou no gerúndio (EmprestandoMaterial) onde o sujeito é normalmente o ator (DEBONI, 2000). Ex.:

- a) o usuário empresta material;
- b) o usuário pesquisa assunto.

Cada caso de uso deve receber uma descrição textual que permita o entendimento do objetivo. Esta descrição pode ser detalhada em cenários. Um cenário é uma instância de um caso de uso, isto é, é uma situação onde o ator utilizou o sistema para conseguir atingir o objetivo do caso de uso. Um cenário pode ser considerado otimista se o ator obteve sucesso no seu objetivo, pode ser pessimista se o ator não conseguiu e ocorreu uma situação de exceção, ou o cenário pode ser alternativo, quando frente a uma situação de exceção o ator

optou por caminhos alternativos. Assim para cada caso de uso pode se descrever um texto com (DEBONI, 2000):

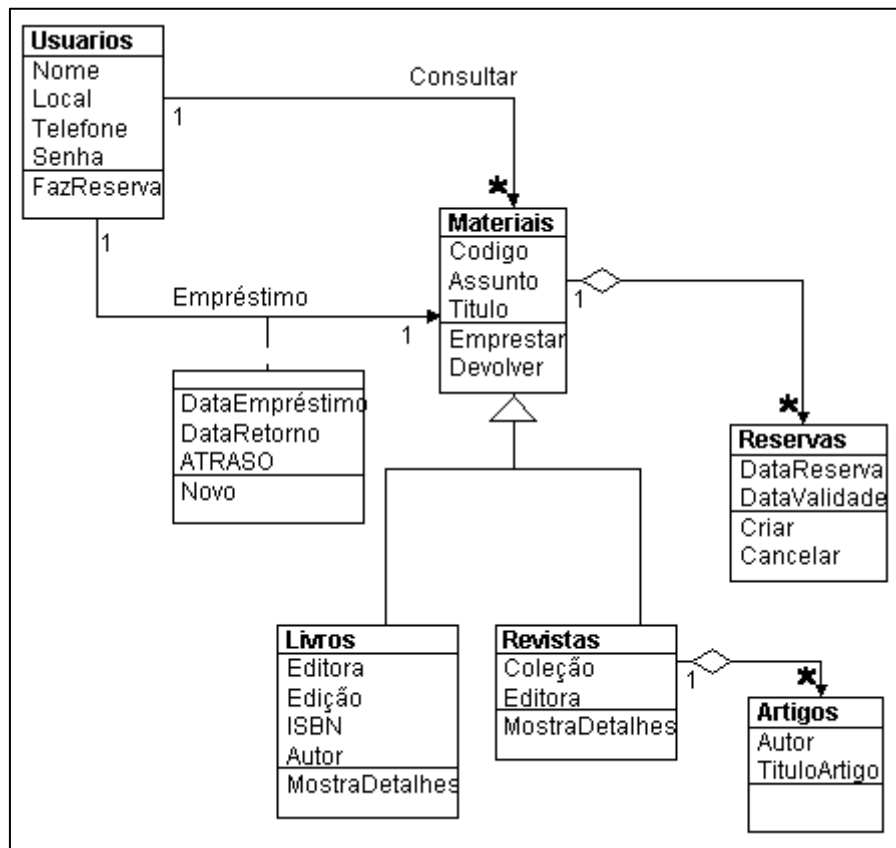
- a) cenários otimistas;
- b) cenários pessimistas;
- c) cenários alternativos.

4.5 DIAGRAMA DE CLASSES

Segundo Deboni (2000), os diagramas de classe descrevem as classes que formam a estrutura do sistema e suas relações. As relações entre as classes podem ser associações, agregações ou heranças. As classes possuem além de um nome, os atributos e as operações que desempenham para o sistema. Uma relação indica um tipo de dependência entre as classes, essa dependência pode ser forte como no caso da herança ou da agregação ou mais fraca como no caso da associação, mas indicam que as classes relacionadas cooperam de alguma forma para cumprir um objetivo para o sistema.

Sendo uma linguagem de descrição, a UML permite diferentes níveis de abstração aos diagramas, dependendo da etapa do desenvolvimento do sistema em que se encontram. Assim, os diagramas de classe podem exibir nas fases iniciais da análise apenas o nome das classes, e em uma fase seguinte os atributos e operações (Figura 11). Finalmente, em uma fase avançada do projeto pode exibir os tipos dos atributos, a visibilidade, a multiplicidade das relações e diversas restrições. Existem elementos na UML para todas estas representações.

FIGURA 11 - EXEMPLO DE DIAGRAMA DE CLASSES



Fonte: DEBONI (2000)

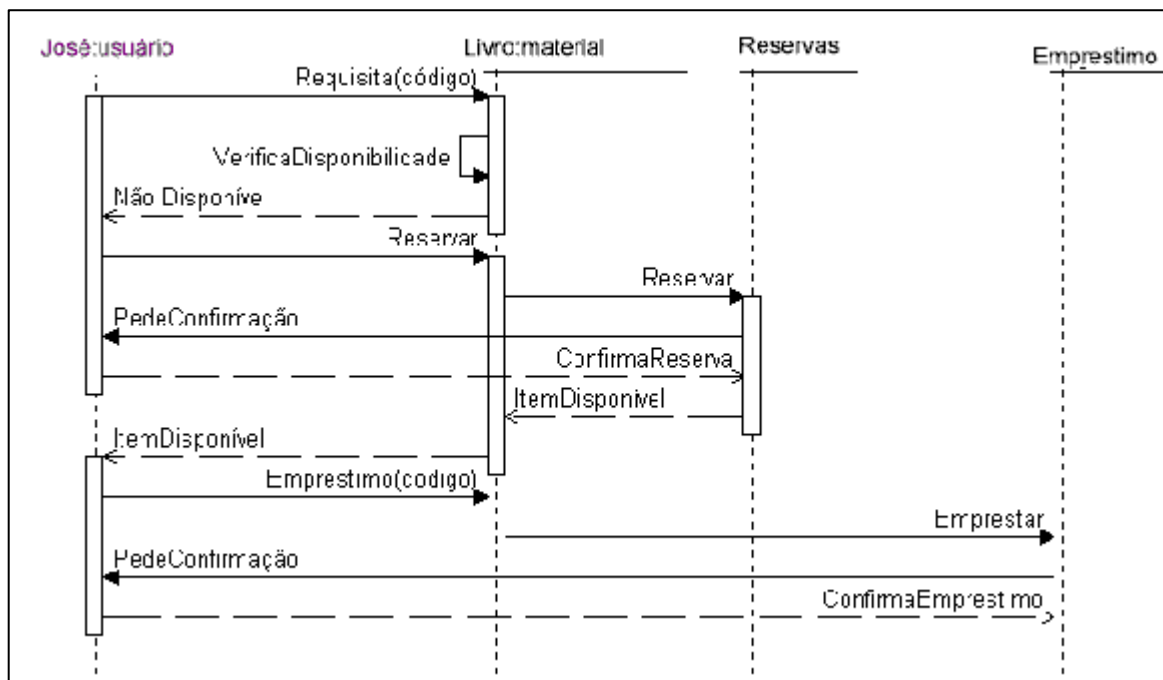
O diagrama de classes, ao final do processo de modelagem, pode ser traduzido em uma estrutura de código que servirá de base para a implementação dos sistemas. Observa-se, no entanto, que não existe no diagrama de classes uma informação sobre os algoritmos que serão utilizados nas operações, e também não se pode precisar a dinâmica do sistema porque não há elementos sobre o processo ou a seqüência de processamento neste modelo. Estas informações são representadas em outros diagramas, como os diagramas de seqüência de eventos ou diagramas de estado (DEBONI, 2000).

4.6 DIAGRAMA DE SEQÜÊNCIA DE EVENTOS

Os casos de uso representam conjunto de cenários que descrevem os diferentes processos que ocorrem no sistema. O diagrama de seqüência de eventos permite modelar estes processos através da troca de mensagens (eventos) entre os objetos do sistema. Os objetos são representados por linhas verticais e as mensagens como setas que partem do objeto que invoca

um outro objeto. As setas podem ser cheias para indicar uma mensagem de chamada ou tracejadas para indicar uma mensagem de retorno. A Figura 12 mostra um exemplo deste diagrama, onde se pode observar que o tempo segue o eixo vertical de cima para baixo. Devem ser desenhados tantos diagramas de seqüência quanto cenários foram levantados no diagrama de casos de uso (DEBONI, 2000).

FIGURA 12 - EXEMPLO DE UM DIAGRAMA DE SEQÜÊNCIA DE EVENTOS



Fonte: DEBONI (2000)

Cada mensagem no diagrama de seqüência de eventos corresponde a uma operação no diagrama de classes. Como as mensagens são operações invocadas, elas devem estar presentes nos objetos de destino, que são ativadas pelas mensagens no objeto de origem (DEBONI, 2000).

5 AMBIENTES DE DESENVOLVIMENTO

5.1 RATIONAL ROSE

O Rational Rose é uma ferramenta CASE que utiliza a linguagem de modelagem de sistemas UML. Quando um objeto é criado no Rational Rose, este é chamado de modelo. Abaixo, uma breve descrição do ambiente:

A Figura 13 mostra a aplicação Rational Rose como aparece quando é aberta pela primeira vez. Há três subjanelas (MORO, 2000) :

- ao topo à esquerda: a Janela de Visão permite que o usuário determine que perspectiva do projeto será usada. Selecionando *Logical View* nessa janela indica que o usuário deseja olhar as classes com seus atributos e operações;
- abaixo à esquerda: Janela de Documentação. É usada para adicionar textos descrevendo as classes;
- à direita: Janela de Diagrama de Classes. É usada para criar a descrição de classes em UML.

FIGURA 13 - TELA INICIAL DO RATIONAL ROSE



No menu *Browse* escolhe-se o tipo de janela que o usuário irá trabalhar. Esta é uma parte importante para a especificação no *Rational Rose*, é neste menu que o usuário define o

tipo de análise que será feito, com as opções: *Class Diagram* (Diagrama de Classes), *Use Case Diagram* (Diagramas de Casos de Uso), *Interaction Diagram* (Diagrama de Interação ou de Seqüência de Eventos), *Component Diagram* (Diagrama de Componentes), *State Diagram* (Diagrama de Estados) e *Deployment Diagram* (Diagrama de Organização).

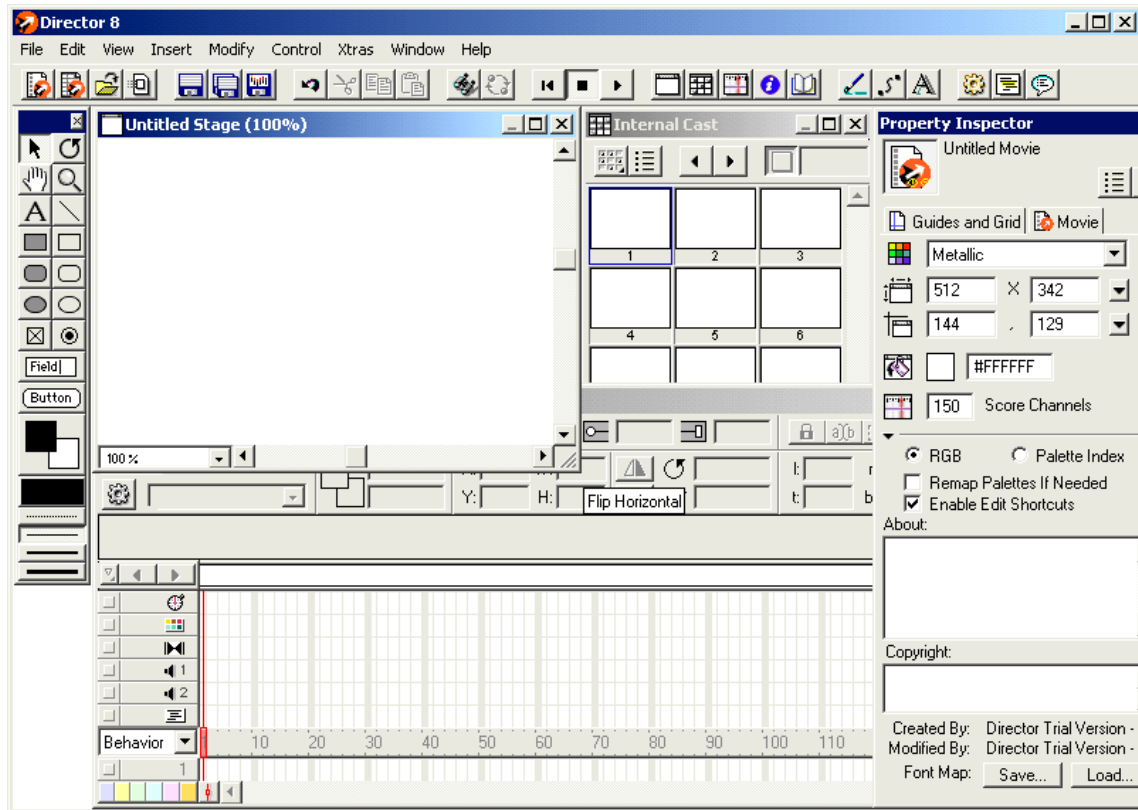
Para este trabalho foram utilizados os Diagramas de Classes, Casos de Uso e de Interação. No Capítulo 6 é descrita a especificação do software proposto sendo ilustrado os diagramas modelados em *Rational Rose* com a Linguagem UML

5.2 AMBIENTE DE DESENVOLVIMENTO: DIRECTOR 8

O Macromedia Director 8 é uma das ferramentas mais utilizadas para o desenvolvimento de softwares hipermídia. Além disso, esta utilização vem aumentando sensivelmente em função da grande evolução do Director nos últimos anos, pois vem incorporando novas e importantes características, principalmente com relação à integração com a Internet (BIZZOTTO, 2000).

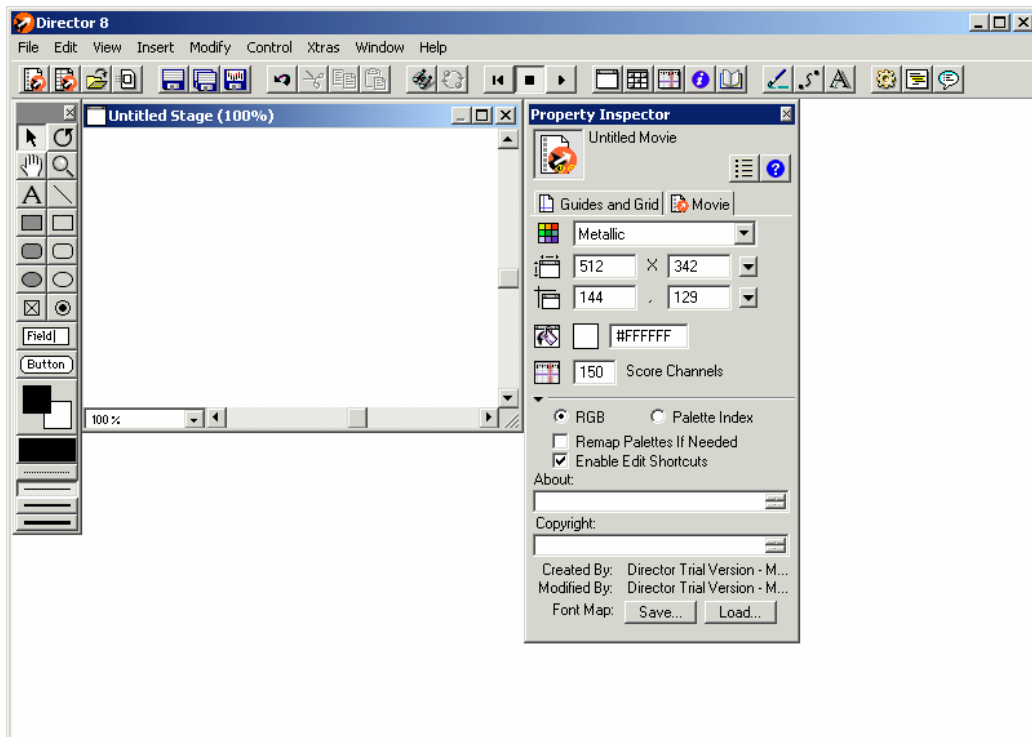
O desenvolvimento no Director é análogo ao desenvolvimento de um filme de cinema ou de uma peça de teatro. Assim, deve-se definir o tipo de filme (software) a ser desenvolvido; os atores (o elenco) que irão participar; o papel de cada ator; a seqüência de cenas, etc. A Figura 14 apresenta alguns dos elementos citados como *stage* (cenário), *CastMembers* (atores), *scripts* (papel de cada ator), *Score* (seqüência de cenas), que serão posteriormente detalhados.

FIGURA 14 – TELA DO DIRECTOR 8



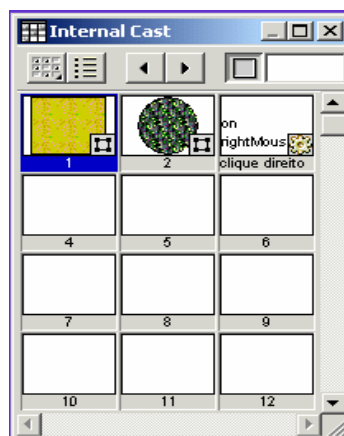
5.2.1 STAGE

No Director, tudo ocorre no *stage*, ou seja, no palco (fazendo uma analogia com o teatro). No *stage*, como mostra a Figura 15, são colocados todos os itens (cenários, atores etc) que irão compor a cena a ser desenvolvida. No *stage* podem ser definidas as propriedades do filme atual (*Modify – Movies Properties*), tais como, tamanho do *stage* de apresentação, a localização, a paleta *Default*, entre outros (BIZZOTTO, 2000).

FIGURA 15. A JANELA *STAGE*

5.2.2 JANELA *CAST*

A Janela *Cast*, possui os *CastMembers*, que são os atores utilizados por um dado projeto, que atuarão sobre o *Stage*. Desta forma, qualquer imagem, som, vídeo etc, incluído no projeto, irá aparecer na janela *Cast*. A Figura 16 mostra os membros no *CastMember*, com suas informações indicadas.

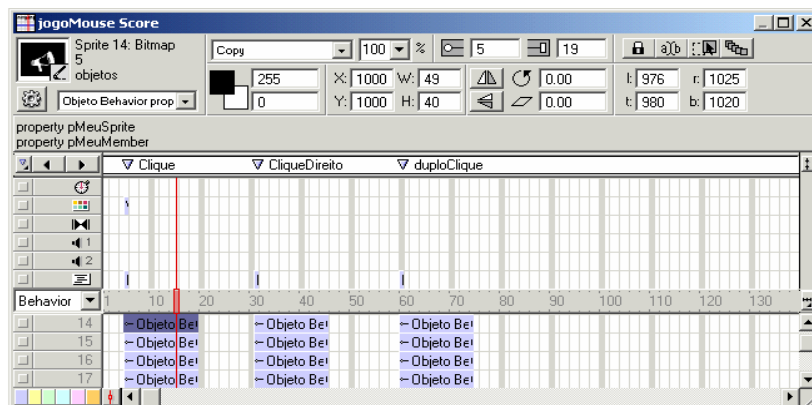
FIGURA 16. A JANELA *CAST*

Um mesmo membro do *Cast Member* pode ser posicionado em locais diferentes do *stage*, sendo chamado de “Instância de *Cast Member*”, ou seja, uma cópia do original. Cada *Cast Member* pode ser utilizado de uma vez em uma mesma cena, sendo que cada um deles pode ter comportamento e aparências distintas.

5.2.3 SCORE

O *score* pode ser entendido como o roteiro (ou seqüência) do filme onde os *Cast Members* são organizados de acordo com o planejado. O *score* é dividido em linhas (canais) e colunas (*frames*). A intersecção de uma coluna com uma linha denominamos célula. Além dos canais usuais para o gerenciamento dos *Cast Members*, ele possui seis canais especiais para efeitos, conforme mostra a Figura 17.

FIGURA 17. A JANELA SCORE



5.2.4 CONTROL PANEL

A janela *Control Panel* é uma janela especial que funciona como se fosse o controle de um videocassete. Além destes controles, esta janela oferece importantes informações sobre o filme. Através desta janela é possível “rodar” um filme (ou parte dele) para avaliar se ele se comporta conforme planejado (BIZZOTTO, 2000).

FIGURA 18. A JANELA *CONTROL PANEL*

Os componentes que compõem o *Control Panel*, conforme mostra a Figura 18 são:

- a) botão *step backward*: retrocede um *frame*;
- b) botão *sound*: controla a intensidade de som no filme;
- c) botão *step forward*: avança um *frame*;
- d) botão *rewind*: retorna para o *frame* 1;
- e) botão *stop*: interrompe a exibição do filme;
- f) *frame counter*: indica onde se encontra a cabeça de leitura e gravação;
- g) botão *play*: inicia a exibição do filme;
- h) botão de modo de tempo real: permite o andamento do filme em *frames* por segundo;
- i) botão modo de tempo real: velocidade real de apresentação do filme;
- j) tempo *display*: indica o andamento do filme;
- k) *loop playback*: permite que o filme fique em *loop*, executando.

5.2.5 LINGO

Lingo é o nome da linguagem utilizada pelo Director, para a produção mais efetiva junto ao usuário.

Através do uso do Lingo, tem-se um maior controle do que acontece no filme, podendo, inclusive, alterar o que foi feito a partir do *score*. Assim, o Lingo é o que torna o Director mais robusto e flexível (BIZZOTTO, 2000).

Segundo Small (1999), o Lingo permite a programação orientada a objetos, embora não seja considerado uma linguagem orientada a objetos. Através das variáveis globais privadas, cada objeto possui características próprias. A terminologia utilizada difere no Lingo, sendo comparados na Tabela 2.

TABELA 2 - TERMINOLOGIA UTILIZADA

Termo no Lingo	Equivalente em Orientação Objeto
<i>Parent Script</i>	Classe (<i>Class</i>)
<i>Child Object</i>	Instância de classe (<i>Class instance</i>)
<i>Property variable</i>	Variável de instância (<i>instance variable</i>)
<i>Handler</i>	Método
<i>Ancestor script</i>	Super classe

FONTE: ADAPTADO DE Dalfovo (1997)

5.2.5.1 SCRIPTS

Os *scripts* do Lingo podem variar quanto a sua localização e tipo (BIZZOTTO, 2000):

- 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*.

5.2.5.2 EVENTOS

No momento em que o usuário vai utilizar o software, ocorrem os seguintes eventos no instante da inicialização (BIZZOTTO, 2000):

- a) *Prepare movie*: este é o primeiro evento que ocorre quando um filme é executado, ou seja, o filme está sendo preparado para ser rodado;
- b) *BeginSprite*: indica que a cabeça de leitura e gravação entrou em um segmento de *sprite* (*sprite span*), ou seja, indica que um *sprite* foi iniciado;
- c) *PrepareFrame*: ocorre antes do Director apresentar um *sprite* no *stage*. Assim, neste evento pode-se incluir ações que alterem as propriedades do *sprite*, uma vez que ele ainda não foi “desenhado”;
- d) *Start Movie*: este evento ocorre no primeiro *frame* do filme.

A partir do momento que o evento *StartMovie* ocorreu, o Director passa a executar os *frames* do filme, ocorrem os seguintes eventos:

- a) *beginSprite*: indica que a cabeça de leitura e gravação entrou em um segmento de *sprite* (*sprite span*). Este evento ocorre sempre que existe um novo segmento de *sprite*;
- b) *prepareFrame*: o *frame* está sendo preparado para ser “desenhado” no *stage*;
- c) *enterFrame*: ocorre depois que o *sprite* é “desenhado” no *stage* e antes que a cabeça de leitura e gravação deixe o *frame* atual;
- d) *exitFrame*: ocorre quando a cabeça de leitura e gravação deixa um dado *frame*;
- e) *endSprite*: ocorre quando a cabeça de leitura e gravação deixa a última célula de um segmento de *sprite*.

Estes eventos são independentes da ação do usuário, ou seja, não é necessária a intervenção do usuário para que estes eventos ocorram. Mas, existem eventos que dependem da ação direta do usuário. Estes eventos são:

- a) *mouseEnter*: ocorre quando o cursor entra no espaço de um *sprite*;
- b) *mouseWithin*: ocorre enquanto o ponteiro do *mouse* estiver sobre um dado *sprite*;
- c) *mouseLeave*: ocorre quando o *mouse* deixa o espaço de um *sprite*;
- d) *mouseDown*: ocorre quando o usuário pressiona o botão do *mouse*;
- e) *mouseUp*: ocorre quando o usuário solta o botão do *mouse*;
- f) *mouseUpOutside*: ocorre quando o usuário pressiona o botão do *mouse* sobre um *sprite*, mas solta o botão fora deste *sprite*;
- g) *rightMouseDown*: ocorre quando o botão direito do mouse é pressionado;
- h) *rightMouseUp*: ocorre quando o botão direito do mouse é liberado;
- i) *keyDown*: ocorre quando o usuário pressiona uma tecla;
- j) *keyUp*: ocorre quando o usuário solta uma tecla.

5.2.5.3 LISTAS

As listas em Director, ou *arrays* em outras linguagens, são “variáveis” que podem conter diversos elementos. A alocação dos elementos na memória é feita segundo a necessidade desta, não denso, portanto, necessárias às informações quanto ao tamanho dos elementos da lista. O primeiro elemento da lista no Director é contado como “1”,

diferentemente das demais linguagens que consideram o primeiro elemento de lista “0” (SMALL, 1999). A Tabela 3 mostra sintaxes para criação de listas.

TABELA 3 – SINTAXE PARA A CRIAÇÃO DE LISTAS

Sintaxe	Criando uma lista
<i>Set the list to []</i>	Para criar uma lista linear vazia
<i>Set the list to [:]</i>	Para criar uma propriedade de lista vazia, pode-se usar um operador de lista para especificar valores na lista.
Especifique a lista de elementos com os parâmetros da função <i>list()</i> . <i>List(value1, value2, value3...)</i>	Para criar uma lista linear usando a função <i>list()</i> . Exemplo: <i>Set designers = list(“Gee”, “kayne”, “Ohashi”)</i>

FONTE: Small (1999).

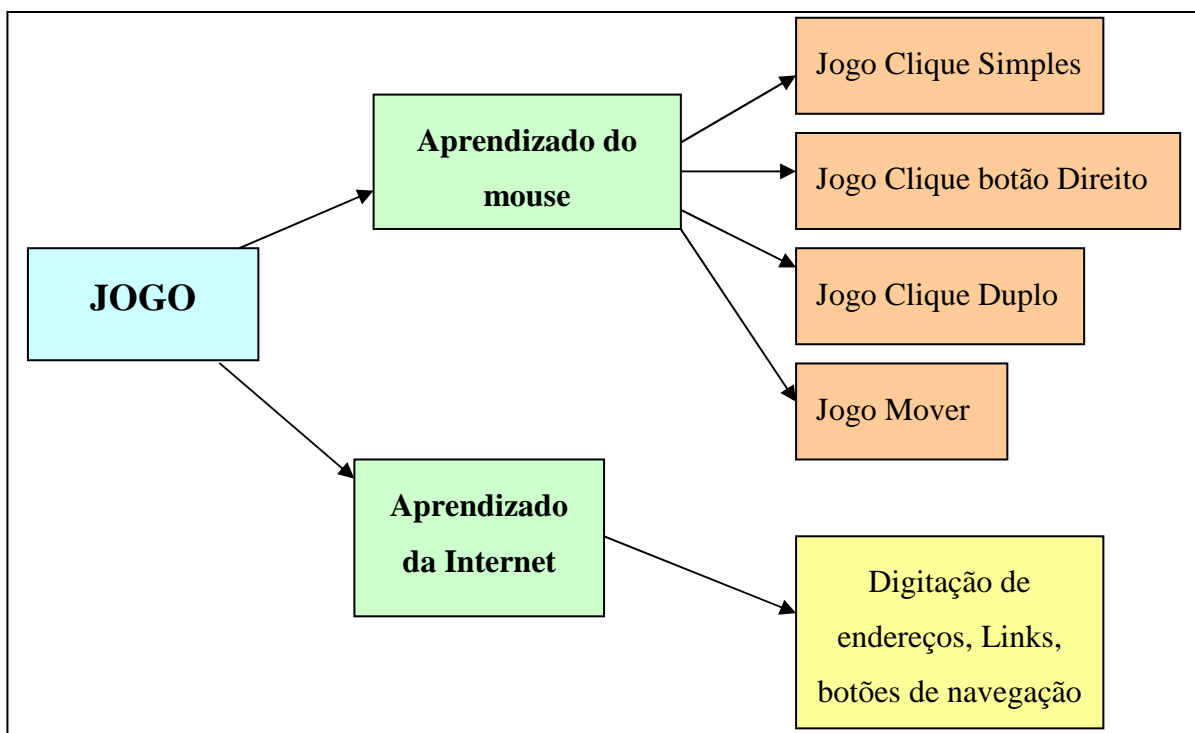
6 SOFTWARE PROPOSTO

O software proposto procura desenvolver as habilidades cognitivas necessárias para a utilização do mouse e a navegação na internet por pessoas da terceira idade. O tipo de software escolhido foi o jogo educacional, porque, conforme ressaltado no Capítulo 2, esse tipo de software possui a vantagem de desenvolver um conceito ou uma habilidade sem que o aluno se dê conta disso.

Para organizar o conteúdo a ser desenvolvido e, com isso, facilitar o aprendizado, o software foi dividido em dois módulos (Figura 19):

- a) **Aprendizado do Mouse:** nesse módulo, o aluno irá treinar os movimentos de clique, clique-duplo, clique com o botão direito e arrastar e soltar;
- b) **Aprendizado da Internet:** o aluno irá exercitar a utilização de um navegador da Internet através de um jogo cujo objetivo é erradicar a Dengue em algumas casas de determinadas cidades. Durante o jogo, o aluno aprende, intuitivamente, a digitar um endereço Web, a utilizar o botão voltar e clicar em *hyperlinks*.

FIGURA 19 - ESTRUTURA DO JOGO



Há uma diferença básica entre o primeiro e o segundo módulo: o andamento do jogo. No primeiro módulo, cada jogo (clique, clique-duplo etc.) é constituído de fases. Assim, quando o usuário executa a tarefa com um índice de acerto superior a 80% , ele “passa de fase”, ou seja, a velocidade de queda dos objetos aumenta, dificultando o jogo.

No segundo módulo, no entanto, não existem fases, mas sim uma tarefa: um conjunto de casas a serem “limpas”. Dessa forma, não há uma variação nas propriedades do jogo (tempo, velocidade etc.).

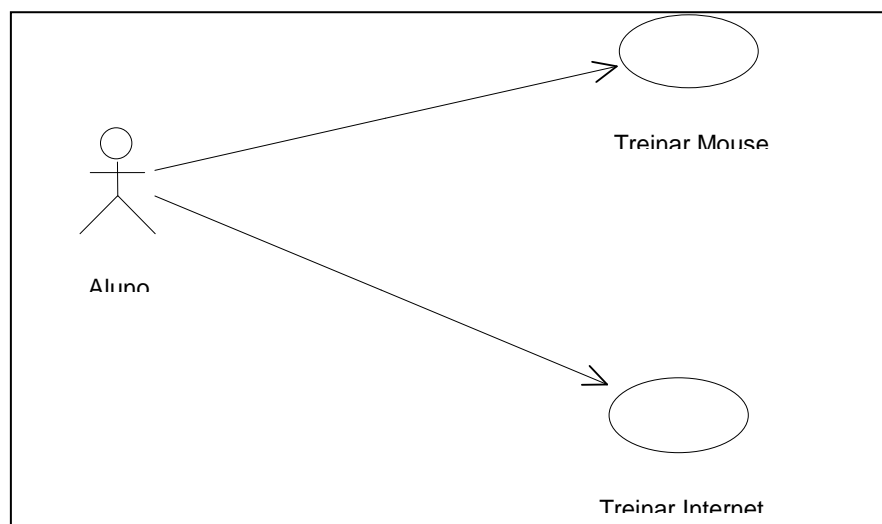
Torna-se importante ressaltar que os dois módulos possuem uma característica em comum: o monitoramento do desempenho do aluno. O software acumula, enquanto o aluno joga, os erros, os acertos, o número de vezes que ficou em uma mesma fase e o tempo gasto. O professor pode, a qualquer momento, acessar o relatório do desempenho do aluno, uma vez que o software armazena essa informação em um arquivo texto externo.

6.1 ESPECIFICAÇÃO DO SOFTWARE

6.1.1 DIAGRAMAS DE CASO DE USO

Utilizando a Linguagem de Modelagem UML (descrita no Capítulo 4), para iniciar a especificação do software foram desenvolvidos dois diagramas de Casos de Uso: Um Caso de Uso para a Visão do Aluno, e outro Caso de Uso para a visão do Professor. Na Figura 20 é demonstrado o Diagrama de Caso de Uso na visão do Aluno. No Quadro 1, explica-se as funções dos casos de uso deste diagrama.

FIGURA 20 - DIAGRAMA DE CASO DE USO: ALUNO

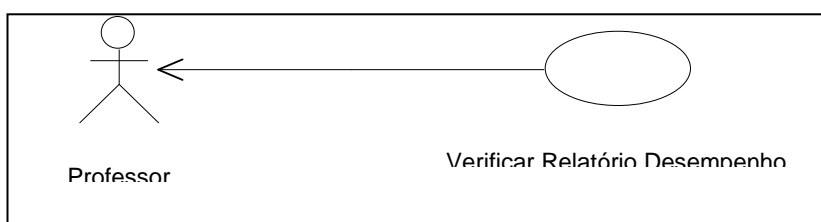


QUADRO 1 - DETALHANDO CASOS DE USO DO DIAGRAMA DE CASOS DE USO: ALUNO

Nº	Caso de Uso	Função
1	Treinar Mouse	O aluno irá jogar exercitando os cliques e movimentos do mouse.
2	Treinar Internet	O aluno irá jogar exercitando digitação de endereços, cliques em botões de navegação e links

Na Figura 21, é demonstrado o Diagrama de Caso de Uso: Professor. Este diagrama mostra o professor verificando o desempenho do aluno no jogo através de um Relatório.

FIGURA 21 - DIAGRAMA DE CASO DE USO: PROFESSOR



6.1.2 DIAGRAMAS DE CLASSES

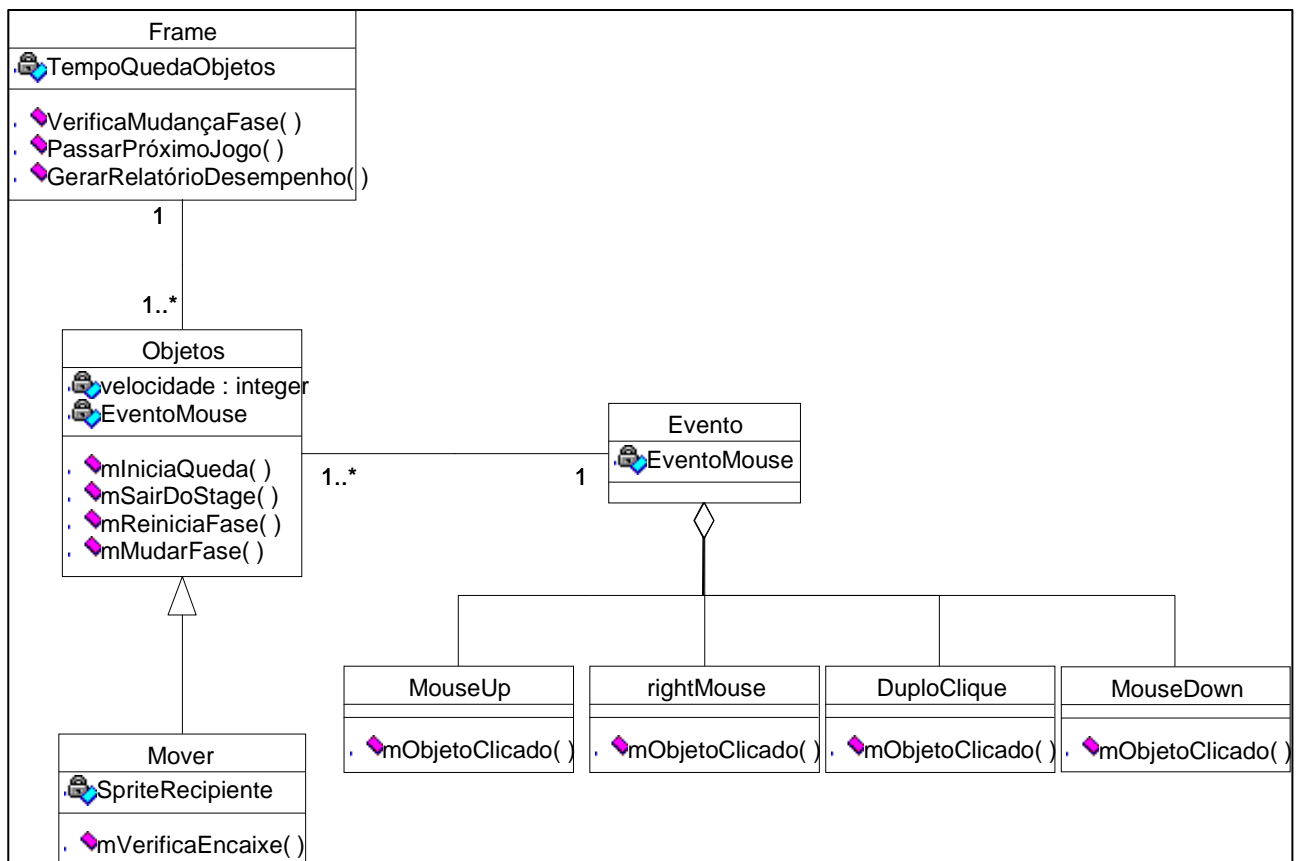
Conforme a Linguagem de modelagem UML (descrita no capítulo 4), foram criados dois diagramas de Classes para cada módulo do Jogo: Um Diagrama de Classes para o Módulo dos Jogos de treinamento com o mouse e outro Diagrama de Classes para o Módulo do Jogo de treinamento da Internet. Na Figura 22 é demonstrado o diagrama de classes do 1º módulo do Jogo: o Jogo do treinamento com o mouse. As classes são:

- a) Classe Frame: possui o controle de mudança de fases e mudança para o próximo jogo para todos os objetos;
- b) Classe Objetos: são os objetos que “caem” na tela, seus atributos são a sua velocidade de queda onde a classe Frame controla, e os eventos do mouse que eles respondem (da classe evento). Suas operações são: IniciaQueda (iniciar a queda dos objetos na tela), SairDoStage (saída dos objetos da tela), ReiniciaFase (quando os objetos reiniciam com a mesma velocidade anterior ao repetir a fase),

MudarFase (quando os objetos mudam de fase aumentando a velocidade de queda destes objetos);

- c) Classe Evento: É uma classe generalizada que possui os controles de eventos de mouse exemplificadas nas classes: MouseUp (clique simples – botão esquerdo do mouse), RightMouse (clique botão direito do mouse), DuploClique (clique duplo sobre o botão esquerdo do mouse) e MouseDown (usado para o jogo de clicar e arrastar objetos na tela);
- d) Classe Mover: São os objetos que “caem” na tela, seus atributos são a sua velocidade de queda onde a classe Frame controla, e o evento do mouse que ele responde (da classe evento) é o especificado na classe MouseDown (para clicar e mover os objetos na tela). Suas operações são: IniciaQueda (iniciar a queda dos objetos na tela), SairDoStage (saída dos objetos da tela), ReiniciaFase (quando os objetos reiniciam com a mesma velocidade anterior ao repetir a fase), MudarFase (quando os objetos mudam de fase aumentando a velocidade de queda destes objetos).

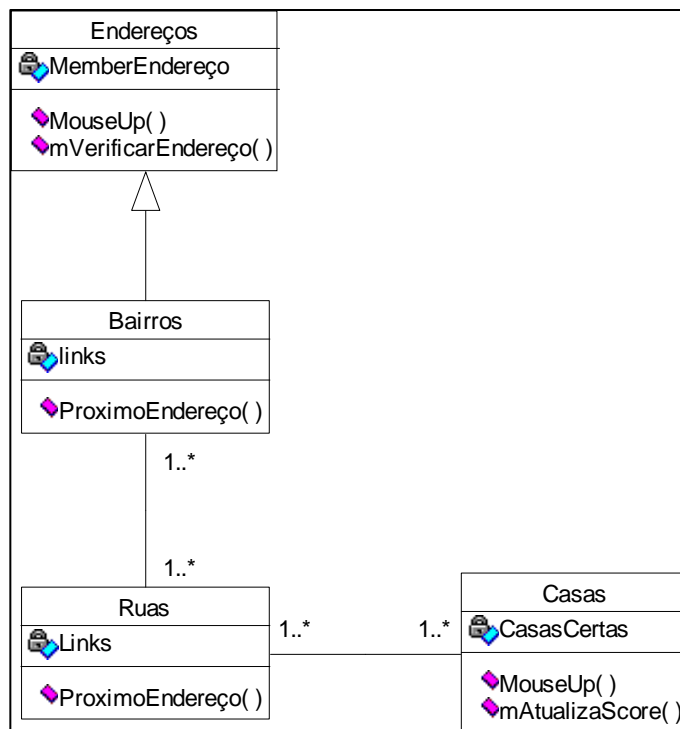
FIGURA 22 - DIAGRAMA DE CLASSES JOGO MOUSE



Na Figura 23 é demonstrado o diagrama de classes do 2º módulo do Jogo: o Jogo do treinamento com a internet. As classes são:

- Classe Endereço: Possui controles para os próximos links. Existe um atributo chamado MemberEndereço que é verificado na operação mVerificarEndereço para comparar se o endereço do link foi digitado corretamente. Esta Classe é Agregada com a Classe Bairros (tipo Todo-parte, onde Bairros são partes do todo que são os endereços);
- Classe Bairros e Classe Ruas: Possuem o endereço do próximo link e uma operação PróximoEndereço que controla esta mudança de página;
- Classe Casas: Possui um atributo CasasCertas utilizado para a contagem e atualização do número de acertos do aluno ao clicar nas casas. Possui 2 operações: mMouseUp (para controle do evento clique simples do mouse) e mAtualizaScore (Para contar quantas casas foram clicadas corretamente).

FIGURA 23 - DIAGRAMA DE CLASSES: JOGO INTERNET



6.1.3 DIAGRAMAS DE SEQÜÊNCIA DE EVENTOS

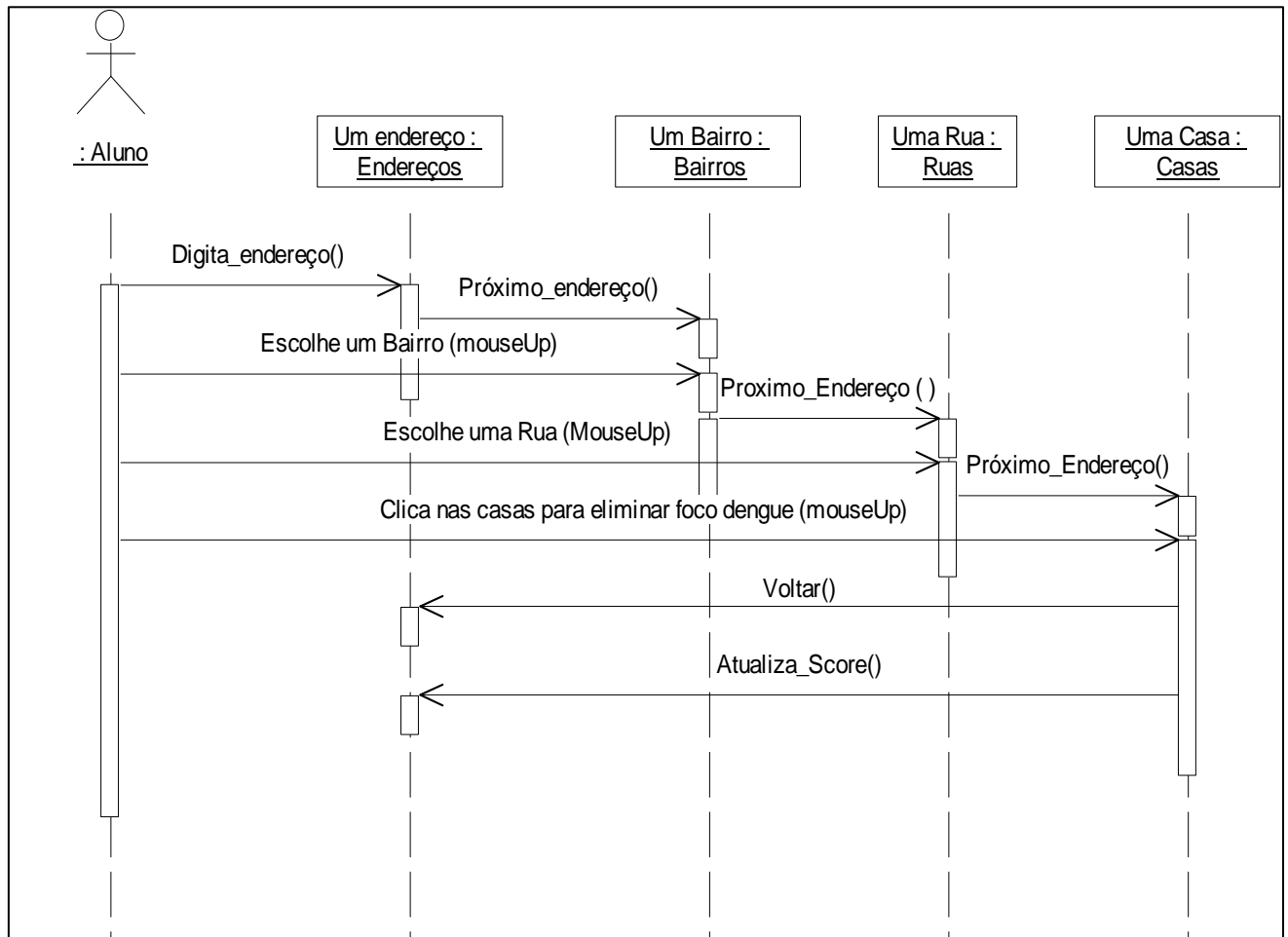
Conforme a Linguagem de modelagem UML (descrita no capítulo 4), foram criados dois diagramas de Interação para a realização da especificação em cada módulo do Jogo: Um Diagrama de Interação para o Módulo dos Jogos de treinamento com o mouse e outro Diagrama de Interação para o Módulo do Jogo de treinamento da Internet. Na Figura 24, é demonstrado o Diagrama de Interação: Jogo Internet. Este Diagrama exemplifica passo-a-passo a interação do aluno com o jogo simulando Internet:

a) O Aluno digita um endereço (dentre os que estão disponíveis em tela), este endereço passa para a página de Bairros (através da operação Próximo_endereço);

b) O Aluno escolhe um dos bairros disponíveis em tela, clicando com o botão esquerdo do mouse. Ao clicar em um bairro, a operação Próximo_endereço abrirá outra página. A página de Ruas. Esta página de Ruas também dispõe de links com nomes de Ruas, onde o aluno também escolhe uma delas com o clique simples do mouse (botão esquerdo do mouse);

c) Ao abrir a última página, o aluno escolhe as casas que quer clicar, e a cada clique correto, uma operação chamada Atualizar_Score, adiciona a quantidade de casas certas clicadas. O aluno também tem a opção de voltar para o início para visitar outro endereço.

FIGURA 24 - DIAGRAMA DE INTERAÇÃO: JOGO INTERNET



Na Figura 25, é demonstrado o Diagrama de Interação: Jogo Mouse. Este Diagrama exemplifica passo-a-passo a interação do aluno com o jogo de treinamento com o mouse:

- a) Ao iniciar o jogo o aluno joga clicando nos objetos com botão simples (MouseUp);
- b) Se o aluno passar de fase, o jogo aumenta a velocidade de queda dos objetos;
- c) Se o aluno não conseguir passar de fase, o jogo repete a fase mantendo a velocidade anterior;
- d) E assim, sucessivamente irá passando de fases aumentando a velocidade de queda dos objetos e passando para os próximos jogos: Clique direito, Duplo clique e Clicar e Mover os objetos. No último jogo do Mouse, o aluno passa para o Jogo Internet.

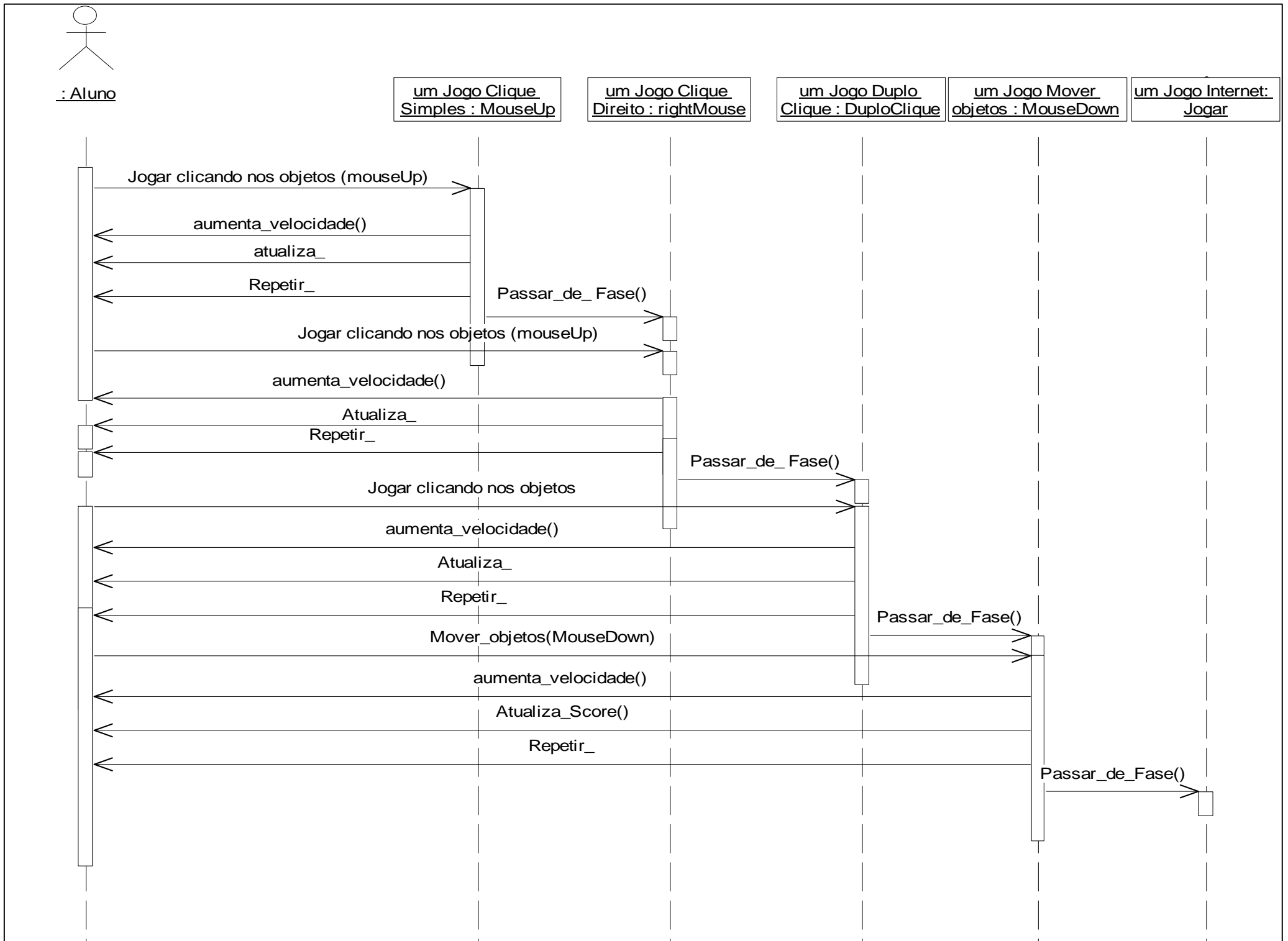


FIGURA 25 - DIAGRAMA DE INTERAÇÃO: JOGO MOUSE

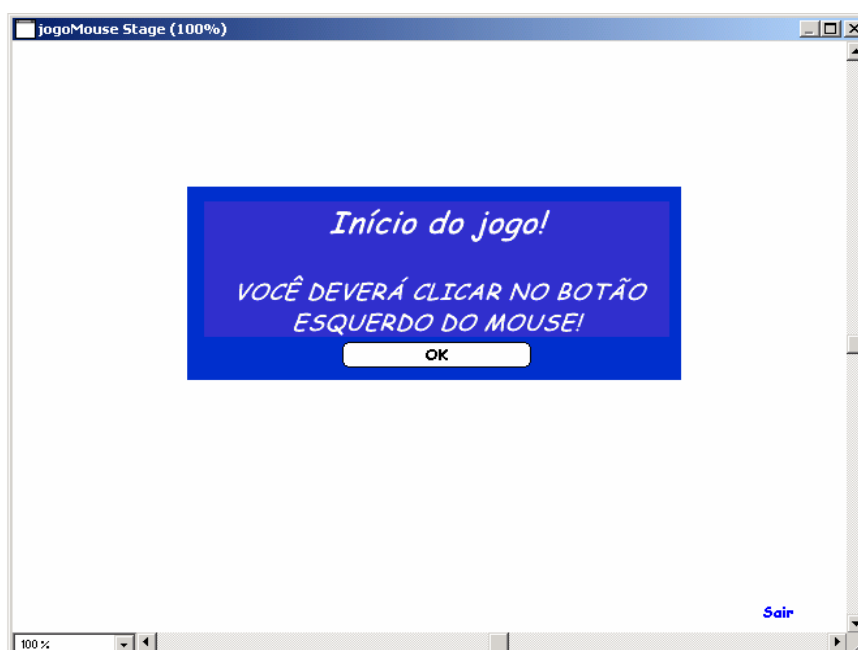
6.2 IMPLEMENTAÇÃO E TUTORIAL DO SOFTWARE

Conforme descrito anteriormente, o Software está dividido em dois módulos principais: Jogo do Mouse e do Jogo Internet. No Jogo do Mouse existem 4 jogos para praticar o uso do mouse: Clique simples, clique botão direito, duplo-clique e clique mover objetos. O objetivo do 1º módulo é clicar nos objetos para passar de fases de cada jogo até chegar ao 2º módulo, onde treina-se a navegação na internet. Nesse módulo, o objetivo é “limpar” as casas de uma cidade do mosquito da dengue. Para isso, o usuário deve navegar pelas cidades, bairros e ruas, para encontrar as casas infectadas.

Nessa primeira versão, o aluno não pode abrir o segundo jogo (duplo-clique) sem passar pelo primeiro (clique simples). A seguir, serão detalhados os passos necessários para a utilização do software desenvolvido.

Primeiramente, quando inicia o Jogo, o aluno deverá digitar o seu nome para que o software possa fazer o monitoramento do seu desempenho. Logo em seguida aparece a tela inicial do 1º módulo. Nesta tela existe um quadro que instrui o aluno de como ele deverá clicar com o mouse nos objetos que irão cair na tela (Figura 26). Quando o usuário clicar no botão OK deste quadro de instrução, iniciará o primeiro jogo do primeiro módulo: o jogo do clique simples do mouse.

FIGURA 26 - TELA INICIO DO JOGO



Nesse jogo, conforme a Figura 27, os objetos “caem” na tela. Para evitar que estes objetos alcancem o “chão” (parte inferior da tela), o aluno deverá clicar sobre cada um dos objetos.

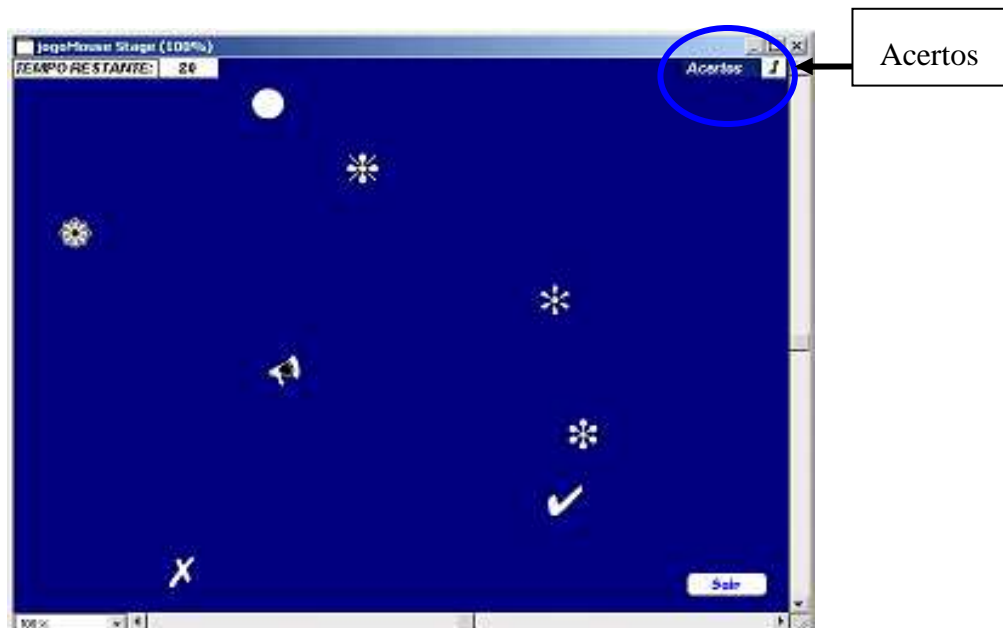
Para acomodar os objetos que estão na tela foi criado uma lista, conforme *script* (código de implementação) no Quadro 2, esta lista acomoda os números dos *sprites* (objetos que estão “caindo” na tela):

QUADRO 2 - CÓDIGO DE IMPLEMENTAÇÃO PARA ADICIONAR OBJETOS A LISTA

```
on mInicializaListaSprites(me)
  -- Gerando uma lista vazia para acomodar os números dos
  sprites
  pListaSprites = []
end mInicializaListaSprites
pListaSprites.add(tSprite) -Para adicionar os sprites a lista
```

Na Figura 27, pode-se observar também que o jogo possui um controle de acertos do aluno. A cada clique correto sobre os objetos, estes objetos “somem” da tela e é acrescido um valor ao quadro *Acertos* no canto superior direito da tela. O *script* do Quadro 3 foi desenvolvido para este fim.

FIGURA 27 - OBJETOS “CAINDO” NA TELA DO JOGO



QUADRO 3 - SOMA DOS OBJETOS CLICADOS

```

-- conta quantos objetos foram clicados
pObjetosClicados = pObjetosClicados + tValor

-- mostra na tela no campo de acertos a quantidade de
objetos que foi clicado

member("score").text = String(pObjetosClicados)

```

Para que os objetos pudessem desaparecer da tela, foi desenvolvido o *script* demonstrado no Quadro 4. Pode-se verificar que os objetos que saem da tela, retornam para a lista novamente para reiniciarem a queda.

QUADRO 4 - RETIRANDO OBJETOS DA TELA E ADICIONANDO-OS NA LISTA DE SPRITES

```

-- posicionando o sprite fora do stage (fora da tela)
sprite(pMeuSprite).loc = point(1000,1000)
updateStage

-- reposicionando o sprite na lista de sprites
mIncluiSprite(script "quadroNegro",pMeuSprite)

-- adiciona novamente o sprite a lista
pListaSprites.add(tSprite)

```

Se o aluno conseguir clicar 80% dos objetos que estão “caindo” na tela, passará de fase onde o jogo aumentará a velocidade de queda dos objetos. Se e o aluno não conseguir passar de fase, o jogo irá repetir a fase mantendo a velocidade anterior. Para isso foi desenvolvido o *script* do Quadro 5.

QUADRO 5 - CÓDIGO PARA VERIFICAR MUDANÇA DE FASE E JOGO

```

-- Verificando a mudança de fase

-- calculando o percentual de objetos clicados
  tPercentual = float(tObjetosClicados) / pObjetosFase*100

-- verificando se este percentual é maior ou igual a
80%(desempenho mínimo)
  if tPercentual >= pDesempenhoMinimo then

-- se for muda de fase obtendo o número da próxima fase
-- vai verificar se existe a próxima fase, isto é, se é um
número inteiro.

if integerP(pFase) then

-- se for passa de fase
  sendAllSprites(#mMudarFase)

  alert "Parabéns! Você passou de fase!"

  sendSprite(pSpriteTempo,#mIniciaTempo)

  else
-- se acabar as fases, muda para o próximo jogo (outro
evento do mouse)

  mPassarProximoJogo(me)

end if

```

Cada Jogo de clique do mouse possui 4 fases (Quadro 5). Quando o aluno passa de fase no jogo, a próxima fase continua com o treino do clique simples do mouse, porém com um acréscimo na velocidade e na quantidade de objetos que o aluno deverá clicar para impedir a queda. Quando o aluno passar pelas 4 fases deste 1º jogo do módulo Jogo do mouse, irá passar para o próximo jogo (2º jogo do módulo Jogo do mouse), o qual treinará outro evento do mouse: o clique com o botão direito sobre os objetos para também impedi-los de “cair” (Figura 28). O trecho do *script* detalhado no Quadro 6, também demonstra o controle da mudança de jogo.

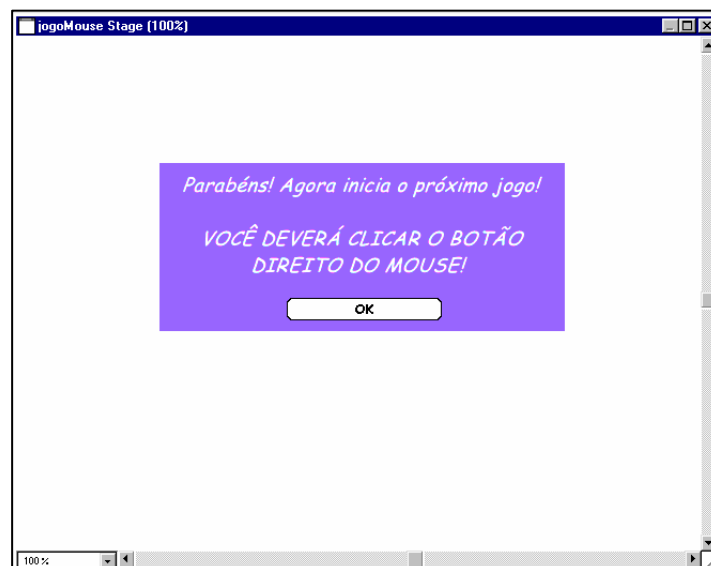
QUADRO 6 - LISTA COM AS FASES E QUANTIDADE DE OBJETOS

```
-- controla a lista de fases do jogo e suas quantidades
respectivas de objetos para cada fase:

on mInicializarJogo(me)

  -- Inicializando lista do jogo
  pListaJogo = [#fase1:20,#fase2:25,#fase3:35,#fase4:40]
```

FIGURA 28 - PASSANDO PARA O 2º JOGO DO 1º MÓDULO:
CLIQUE BOTÃO DIREITO DO MOUSE



Para os Jogos de Clique Botão Direito do mouse (2º Jogo do 1º módulo) e Duplo Clique (3º Jogo do 1º módulo), as telas são iguais como mostrada na Figura 27, e a implementação é a mesma já explicada para a 1ª fase do 1º jogo. Na passagem para os próximos jogos, as telas são semelhantes a mostrada na Figura 28, com um aviso detalhando como será o próximo jogo.

Para que os objetos obedecessem determinados cliques do mouse em cada um dos Jogos do 1º módulo, criou-se uma lista de eventos do mouse. Cada evento do mouse foi

atribuído aos objetos durante o início da implementação de cada jogo. O Quadro 7 detalha a criação da lista de eventos do mouse.

QUADRO 7 - CRIAÇÃO DA LISTA DE EVENTOS DO MOUSE

```
-- criação da lista de eventos do mouse

pPropertyList[#pEvento] = [\
    #comment:"Objeto responde a qual evento?",\
    #format:#symbol,\
    #range:[#mouseUp,#rightMouse,#duploClique,#mouseDown],\
    #default:#mouseUp\
]
```

Para o Jogo de Clicar e Mover (4º e último jogo do 1º módulo), foram feitas algumas alterações na tela e nos objetos. Os objetos agora devem ser clicados e movidos para as caixas que estão na tela. Para isso, foi criado um cenário com duas caixas: Animais e Flores. O aluno deverá clicar e mover os objetos (figuras de animais e flores) para suas respectivas caixas (Figura 29).

FIGURA 29 - TELA JOGO CLICAR E MOVER OBJETOS



Para que os objetos “sumissem” da tela ao serem colocados nas suas caixas certas, desenvolveu-se o *script* do Quadro 8 onde verifica-se o encaixe dos objetos.

QUADRO 8 - VERIFICANDO ENCAIXE DE OBJETOS NAS CAIXAS

```

-- para saber qual objeto se encaixa em determinada
caixa, foi acrescentada uma lista onde a caixa que é o
retângulo recebe este sprite da lista.

tRetangulo = sprite(pSpriteRecipiente).rect

-- verificando se o local do meu objeto na tela ao
clicar com o mouse e parando o movimento de queda do
objeto.
tLoc = sprite(pMeuSprite).loc
pMovendo = FALSE

-- se o objeto está dentro da caixa correta (isto é
definido para cada objeto quando inserido na tela no
momento da implementação) o objeto sai da tela, senão
o objeto continua caindo.
  if tLoc.inside(tRetangulo) then

    mSairDoStage(me)

  else

    pCaindo = TRUE

  end if

```

No segundo módulo do Jogo (Jogo Internet) o aluno deverá digitar o endereço de uma das cidades que estão disponíveis na tela (escolhendo uma delas e digitando no campo endereço), e clicar o Botão Ir para abrir a próxima janela onde contém uma lista de Bairros. O aluno deverá digitar corretamente este endereço para que possa acessar a próxima parte (Figura 30). Para o Botão abrir a próxima janela onde estão os bairros (Figura 31), foi desenvolvido o *script* detalhado no Quadro 9.

FIGURA 30 - TELA JOGO INTERNET: DIGITAÇÃO DE ENDEREÇOS PARA ACESSAR OUTRAS PÁGINAS



QUADRO 9 - VERIFICAÇÃO DA DIGITAÇÃO CORRETA DO ENDEREÇO

```
-- existe uma lista que armazena os endereços corretos
-- ao clicar o botão Ir é verificado se o texto digitado
é igual ao endereço correto
```

```
tTexto = pMemberEndereco.text
```

```
case tTexto of:
```

```
-- verificando a digitação dos endereços para abertura
das páginas de cada cidade
```

```
"www.blumenau.com.br":
  pEndereco = "blumenau"
```

```
"www.florianopolis.com.br":
  pEndereco = "florianopolis"
```

```
"www.joinville.com.br":
  pEndereco = "joinville"
```

```
"www.criciuma.com.br":
  pEndereco = "criciuma"
```

```
-- se for igual, o botão abrirá a próxima página
```

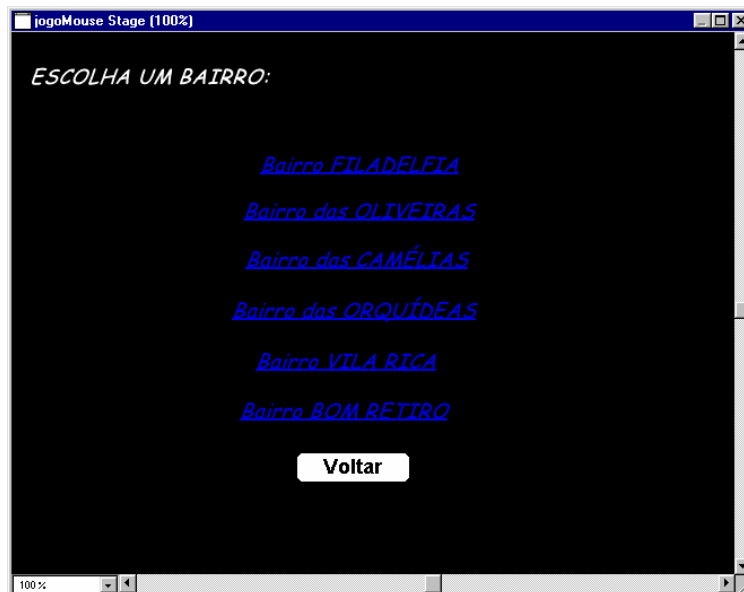
```
go to frame pEndereco
```

```
-- senão uma mensagem avisando o erro aparece na tela
```

```
alert "Endereço Digitado Incorretamente!"
```

```
"www.itajai.com.br":
```

FIGURA 31 - TELA JOGO INTERNET: BAIRROS



Quando o aluno escolher e clicar um dos bairros disponíveis na tela, abrirá uma próxima janela semelhante a mostrada na Figura 31 com uma lista de Ruas que ao serem clicadas levarão a última tela do jogo: tela onde estão as casas com “foco da dengue”. Quando o aluno clica nas casas “infectadas”, ele elimina o foco da dengue (Figura 32). Para criar uma idéia de casa limpa, quando o aluno clica nas casas com céu azul (casas infectadas), um aviso aparece na tela parabenizando o aluno pela limpeza da casa, e a casa é substituída pela casa com o céu branco. Para desenvolver este procedimento, foi implementado o *script* detalhado no Quadro 10.

FIGURA 32 - TELA JOGO INTERNET: CASAS



QUADRO 10 - CÓDIGO DE IMPLEMENTAÇÃO JOGO INTERNET: CASAS

```
-- Ao clicar nas casas, será feita uma verificação se esta
casa era infectada (já definida antes na implementação o
nome casainfectada para este objeto). Se este objeto clicado
era um casaInfectada, na tela aparecerá um aviso dizendo que
a casa está livre da dengue, e substituirá esta casa por
outro objeto casa que é o objeto casaLimpa.

on mouseUp(me)

if pMeuMember.name = "casaInfectada" then

    sprite(pMeuSprite).member = member("casaLimpa")

    alert "Ótimo! Esta casa agora está livre da dengue!"

else

    alert "Essa casa já está livre da dengue!"

-- se clicar nas casas que já são objetos casaLimpa, um
aviso aparece na tela dizendo que essa casa já está livre da
dengue, não realizando nenhuma substituição.
```

6.3 RELATÓRIO DA UTILIZAÇÃO DO SOFTWARE

O software foi testado com cinco pessoas em potencial. Todas estas pessoas possuíam dificuldades com a manipulação do mouse para efetivar seu aprendizado da informática. Quando o software foi mostrado a estas pessoas, despertou-lhes grande interesse, pois sabiam que era um jogo (o que lhes pareceu muito divertido).

Ao iniciarem o aprendizado através do jogo, sentiram um pouco de dificuldade para manipular o mouse, mas com a repetição de movimentos e cliques do mouse foram adquirindo habilidade motivados com o intuito de vencer o jogo, e o trabalho de ensinar a manipular o mouse (acostumar a coordenação motora), que antes era desgastante e levava mais tempo até que o aluno conseguisse ter o domínio, foi substituído por um aprendizado motivante, divertido e eficaz.

A seguir, alguns depoimentos referentes a utilização do jogo:

A Sra. Cecília Krieger, 63 anos, diz que:

“Antes era muito complicado para encontrar a seta no monitor e conseguir associá-la com o movimento do mouse, as vezes eu movimentava muito rápido ou clicava errado. Com o jogo sinto que estou conseguindo movimentar mais rápido o mouse e conseguindo clicar nos botões corretos do mouse!”

A Sra. Terezinha Prolico, 72 anos, afirma que:

“Nunca senti interesse em aprender informática, mas conhecendo este jogo vi nele a facilidade em aprender e a conhecer como se trabalha com o mouse e a aprender a internet!”

A Sra. Regina Buchler, 56 anos, diz que:

“Sinto-me motivada a ganhar este jogo, e como aumentam as velocidades sinto que cada vez mais estou conseguindo movimentar mais rapidamente o mouse. Muito interessante!”

A Sra. Rosemary Herwig, 59 anos afirma que:

“Agora consigo movimentar o mouse mais rápido! Clicar nos objetos, é muito legal!”

7 CONCLUSÃO

O presente trabalho teve como objetivo o desenvolvimento de um software educacional, na forma de jogo, que auxiliará o aprendizado do uso do *mouse* e de um navegador da internet por pessoas da terceira idade. Em seu formato atual, o software desenvolvido permite o aprendizado do uso do *mouse*, o que inclui o clique, clique-duplo, clique com o botão direito e arrastar-soltar. O software permite também o aprendizado dos recursos básicos de um navegador da internet, o que inclui a digitação de endereços, navegação através de *links*, voltar e avançar.

Com isso, observa-se que, com relação às funções previstas, o presente trabalho atendeu plenamente aos objetivos traçados. No entanto, não foi possível um monitoramento efetivo do desempenho dos alunos. Na versão atual, o software armazena apenas o desempenho geral do aluno em cada jogo (percentual de acertos).

O software foi apresentado a alguns usuários em potencial para uma avaliação preliminar e informal. Os resultados, apesar de positivos, não podem ser considerados conclusivos, pois, o número de usuários e o tempo de utilização do software, bem como os mecanismos de validação, não foram os adequados.

As ferramentas utilizadas, Rational Rose e Macromedia Director 8, mostraram-se adequados para o desenvolvimento do software proposto.

7.1 EXTENSÕES

Como sugestões para trabalhos futuros, pode-se citar:

- a) Criação de um jogo para praticar a digitação;
- b) Disponibilizar o jogo na internet;
- c) Criar uma versão multi-usuário.

REFERÊNCIAS BIBLIOGRÁFICAS

ABDALA, Vitor. **Informática sem mistério**. O computador não é um bicho de sete cabeças. Rio de Janeiro, ? 2001. Disponível em: <<http://www.maisde50.com.br/artigo.asp?id=4122>>. Acesso em: 04 abr. 2002.

BARROS, Pablo. **UML: Linguagem de Modelagem Unificada em Português**. São José, dez. 1998. Disponível em: <<http://ftp.sj.univali.br/prof/Rodrigo%20Bittencourt%20Westrupp/Westrupp/antigo/2162/UML-em-Portugues/page03.html>>. Acesso em: 07 maio 2002.

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

BRASIL. Presidência da República. **Presidente (1999-2002: F.H. Cardoso)**. 7 anos do real. Estabilidade, crescimento e desenvolvimento social. Brasília, DF, 2001. 42 p. Disponível em: <http://www.planalto.gov.br/publi_04/cap_1.pdf>. Acesso em: 09 mar. 2002.

CAMPOS, Emilene. **Terceira idade ganha espaço público para acesso à internet**. Juiz de Fora, set. 2001. Disponível em: <<http://www.jfservice.com.br/informatica/arquivo/infojf/2001/09/27-idosos>>. Acesso em: 21 mar. 2002.

CAMPOS, Fernanda; CAMPOS, Gilda; ROCHA, Ana Regina. **Dez etapas para o desenvolvimento de Software Educacional do tipo hipermídia**. Rio de Janeiro, ? 1996. Disponível em: <http://www.niee.ufrgs.br/ribie98/CONG_1996/CONGRESSO_HTML/19/ETAPAS.HTML> Acesso em: 22 mar. 2002.

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

CHAVES, Paulo Humberto Barbosa. **Informática para terceira idade faz sucesso e é aprovada pelos alunos**. Rio de Janeiro, set. 2001. Disponível em: <<http://www.puc-rio.br/jornaldapuc/nov97/informatica.html>>. Acesso em: 20 out. 2001.

COMMERCIIO, Jornal do. **Vovô viu a Web (e gostou!)**. Recife, jan. 2001. Disponível em: <<http://www.sudene.gov.br/anteriores/456/informatica.html>>. Acesso em: 11 abr. 2002.

DALFOVO, Regiani. **Protótipo de software para o ensino de introdução a microinformática**. 1997. 65 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

DEBONI, José Eduardo Zindel. **Breve Introdução aos Diagramas da UML**. Voxxel Consultoria de Sistemas, São Bernardo do Campo, ?.2000. Disponível em: <<http://www.voxxel.com.br/IntroUML/>>. Acesso em: 06 maio 2002.

FAGUNDES, Léa da Cruz. **Informática na Escola**. Tecnologia Educacional, Rio de Janeiro, v.21 (107), p. 79-84, jul./ago. 1992.

FERREIRA, Josemar Dias. **Multimídia para programadores e analistas**. Rio de Janeiro: Infobook, 1995.

FOWLER, Martin; SCOTT, Kendall. **UML Essencial: um breve guia para a linguagem-padrão de modelagem de objetos**. Porto Alegre: Bookman, 2000.

GAGNÉ, Robert M. **Princípios essenciais da aprendizagem para o ensino**. Porto Alegre: Globo, 1980.

GOLDMAN, Sara Nigri. **A Internet: Rede de inclusão ou de exclusão?** Rio de Janeiro, ?.2001. Disponível em: <<http://www.idademaiores.com.br/areas/inform.htm>>. Acesso em: 21 mar. 2002.

GONZALEZ, Julio Francisco Planella. **Director 8.5: criando aplicativos multimídia**. São Paulo: Berkeley Brasil, 2001.

GREEN, Thomas J. **Macromedia Director 8 & Lingo – guia rápido para desenvolvimento na Web**. Rio de Janeiro: Ciência Moderna Ltda.,2001.

GROSS, Phil. **Director 8 and Lingo**. Berkeley: Macromedia, 2000.

HALFOUN, Eli. **Idosos já são 629 milhões e representam 10% da população.** Tribuna da imprensa, Rio de Janeiro, Terceira Idade, p.11. mar 2002. Disponível em: <<http://www.cies.org.br/scripts1/novocliping.asp>>. Acesso em: 03 mar 2002.

LABORCIÊNCIA, Produtos Educacionais e Editora Ltda. **Softwares Educacionais: Simuladores,** São Paulo, ago 2001. Disponível em: <<http://www.laborciencia.com.br/inf.htm>>. Acesso em: 14 jun 2002.

MATURIDADE, O Jornal do poder maduro. **Internet para você:** o pessoal maduro está caindo na rede. São Paulo, nº 43, jun 2000. Disponível em: <<http://www2.uol.com.br/maturidade/m43com3.htm>>. Acesso em: mar 2002.

MEC. Ministério da Educação. Desenvolvido pelo Departamento de Informática de educação à distância, 2001. Apresenta textos sobre o Programa nacional de informática na educação – Proinfo. Disponível em: <<http://www.proinfo.gov.br>>. Acesso em: 10/04/2002.

MCCLOSKEY, M., **Intuitive Physics**, Sci. Am. 249, 1983 114.

MIELKE, Fernando Luiz. **Ensino assistido por computador:** algumas considerações teóricas da ergonomia e da inteligência artificial num ambiente hipertexto. 1991. 112 f. Dissertação de Mestrado em Engenharia de Produção – Departamento de Engenharia de Produção e Sistemas, Universidade Federal de Santa Catarina, Florianópolis.

MONTEIRO, Eduardo Bastos; GOMES, Flávia Rezende dos Santos. **Informática e educação.** São Paulo: Tecnologia Educacional, 1993.

MORO, Mirella Moura. **Engenharia de software**, Porto Alegre, maio 2000. Disponível em: <<http://www.inf.ufrgs.br/~mirella/cmp102/>>. Acesso em: 03 abr. 2002.

NASCIMENTO, Gilson. **Informática na terceira idade:** um guia bem humorado para quem quer aprender informática depois dos 60 anos. Rio de Janeiro: Brasport, 2000.

NEITZEL, Luis Carlos. **Tutorial Frontpage Express**, Barra Velha, dez. 2001. Disponível em: <<http://www.geocities.com/Athens/Sparta/1350/frontpage.html>>. Acesso em: 14 jun. 2002.

PEIXOTO, Ana Sofia Brito, OLIVEIRA, Rosa dos Anjos. **Terminologia do ensino por computador:** abordagem socioterminológica, Brasília, ?1995. Disponível em: <<http://www.ibict.br/cionline/240395/24039514.pdf>>. Acesso em: 01 abr. 2002.

ROCHA, Ana Regina, CAMPOS, Fernanda, CAMPOS, Gilda. **Dez etapas para o desenvolvimento de software educacional do tipo hipermídia**, Rio de Janeiro, ? 1996.

Disponível em:

<http://www.timaster.com.br/revista/colunistas/ler_colunas_emp.asp?cod=331>. Acesso em: 05 abr.2002.

SILVEIRA, Sidnei Renato. **Estudo de uma Ferramenta de Autoria Multimídia para a Elaboração de Jogos Educativos**, Porto Alegre, ?1998. Disponível em: <<http://www.inf.ufrgs.br/pos/SemanaAcademica/Semana98/sidnei.html> > Acesso em: 04 abr. 2002.

SMALL, Peter. **Lingo sorcery: the magic of lists, objects and intelligent agents**. Chichester: John Wiley, 1999.

TESSARI, Olga Inês. **Novas Tecnologias**, São Paulo, ?2001. Disponível em: <<http://apsicologa.tripod.com.br/oit/id11.html>> Acesso em: 22 mar. 2002.

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

WEBFOLIO, Educação a Distância. **Jogos educacionais**, Porto Alegre, jun. 2002. Disponível em: <<http://www.pgie.ufrgs.br/webfolioead/jogos.html>> Acesso em: 14 jun. 2002.