

**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.**

ITATIANA BÁRBARA NOVAK WENDT

Roteiro de Apresentação

- Introdução
- Objetivos
- Justificativa
- História da informática na educação
- Tipos de Softwares educacionais
- Etapas para o desenvolvimento do Software educacional
- A informática na terceira idade
- Porque ensinar informática através de um jogo as pessoas da terceira idade ?
- Conceitos de Orientação a objetos: UML - Diferentes visões do sistema
- Ambientes de Desenvolvimento: Rational Rose e Director 8
- Software proposto
- Conclusão
- Extensões
- Referências Bibliográficas

Introdução

A década de 90 do século passado foi testemunha de uma expressiva ampliação do uso de computadores.

Dentro dessa realidade, o domínio das novas tecnologias da informação e da comunicação é requisito fundamental.

O que se observa, no entanto, é que a grande maioria da população ainda não tem acesso a essas possibilidades.

Evidente que uma das causas é econômica. Mas, 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.

Apesar da grande quantidade de softwares disponível para o aprendizado da informática, o público-alvo principal são crianças e jovens. 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.

Objetivos

O Objetivo principal é 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. Além desse objetivo, o trabalho possui também os seguintes objetivos específicos:

- possibilitar o aprendizado do uso do *mouse*, o que inclui o clique, clique duplo e arrastar-soltar;
- 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*;
- monitorar o desempenho do aluno em cada um dos módulos e jogos incluídos no software;
- análise/teste da efetividade do software produzido, junto ao público alvo.

Justificativa

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 para elaborar um software que atendesse suas necessidades. Esta pesquisa 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 encontradas para a utilização do computador é o manuseio do *mouse*.

História da informática na educação

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 (PEIXOTO, 1995).

O Projeto LOGO - operacionalizado a partir de 1978, ano em que, de fato, a informática chegou a escola pública brasileira, em Campinas.

A UNICAMP criou em 1983, o Núcleo Interdisciplinar de Informática Aplicada a Educação – NIED (PEIXOTO, 1995).

Projeto EDUCOM - criado em 1983 pela Secretaria Especial de Informática da Presidência da República e pelo Ministério da Educação. Este projeto originou o Programa Nacional de Informática Educativa – PRONINFE, lançado em 1989, para apoiar o desenvolvimento e a utilização de novas tecnologias de informática nos ensinos fundamental, médio e superior e na educação especial.

Softwares Educacionais

Software Educacional é geralmente classificado como subconjunto de sistemas de treinamento.

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.

Tipos de Softwares educacionais

- Tutorial
- Exercício e prática
- Ambiente de Autoria
- Simulação
- Jogos

Tipos de Softwares educacionais

- Tutorial

Tutorial FrontPage Express

Som

Para que sua página ao ser exibida execute uma música, clique com o botão direito sobre uma área qualquer da página e escolha a opção PROPRIEDADES DA PÁGINA. Na janela que se abre, localize na alça GERAL a opção SOM DE FUNDO. Clique no botão PROCURAR, localize o som (mid ou wav, de preferência), clique em ABRIR. Configure quantas vezes será executado no *box* LOOP ou assinale ININTERRUPTO se for o caso.

Propriedades da página

Local:

Loop: Ininterrupto

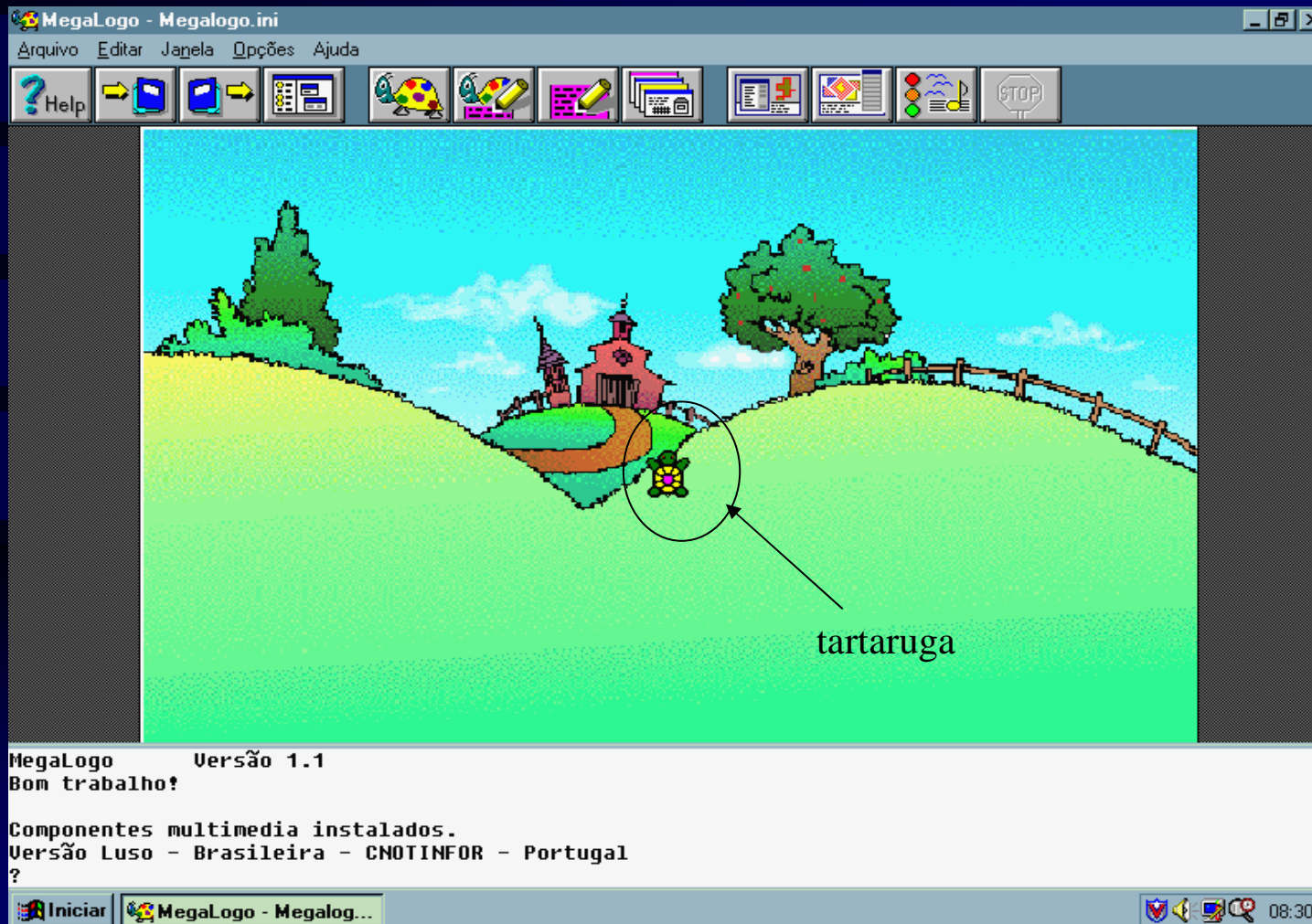
Tipos de Softwares educacionais

- Exercício e prática



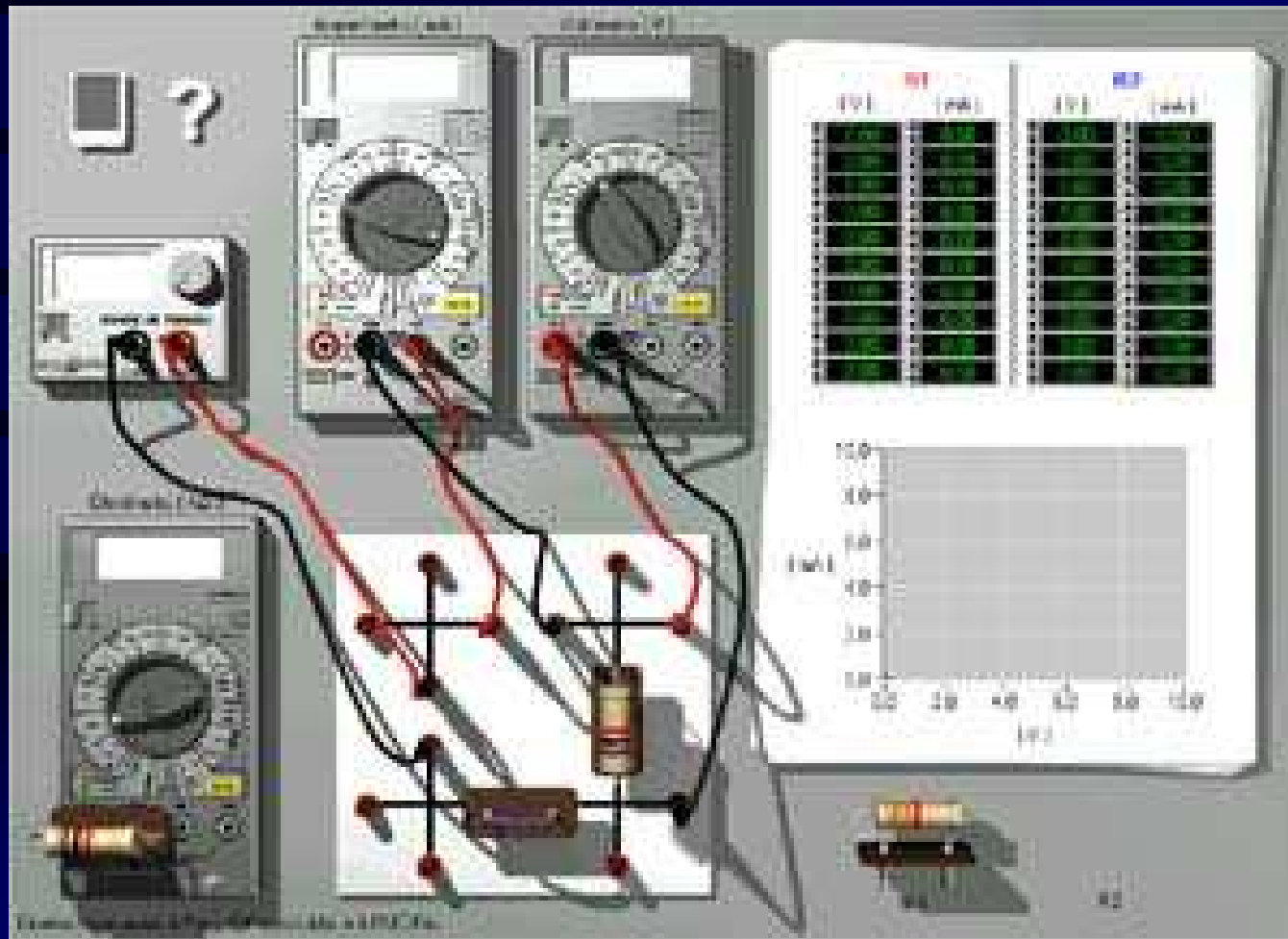
Tipos de Softwares educacionais

- Ambiente de autoria



Tipos de Softwares educacionais

- Simulação



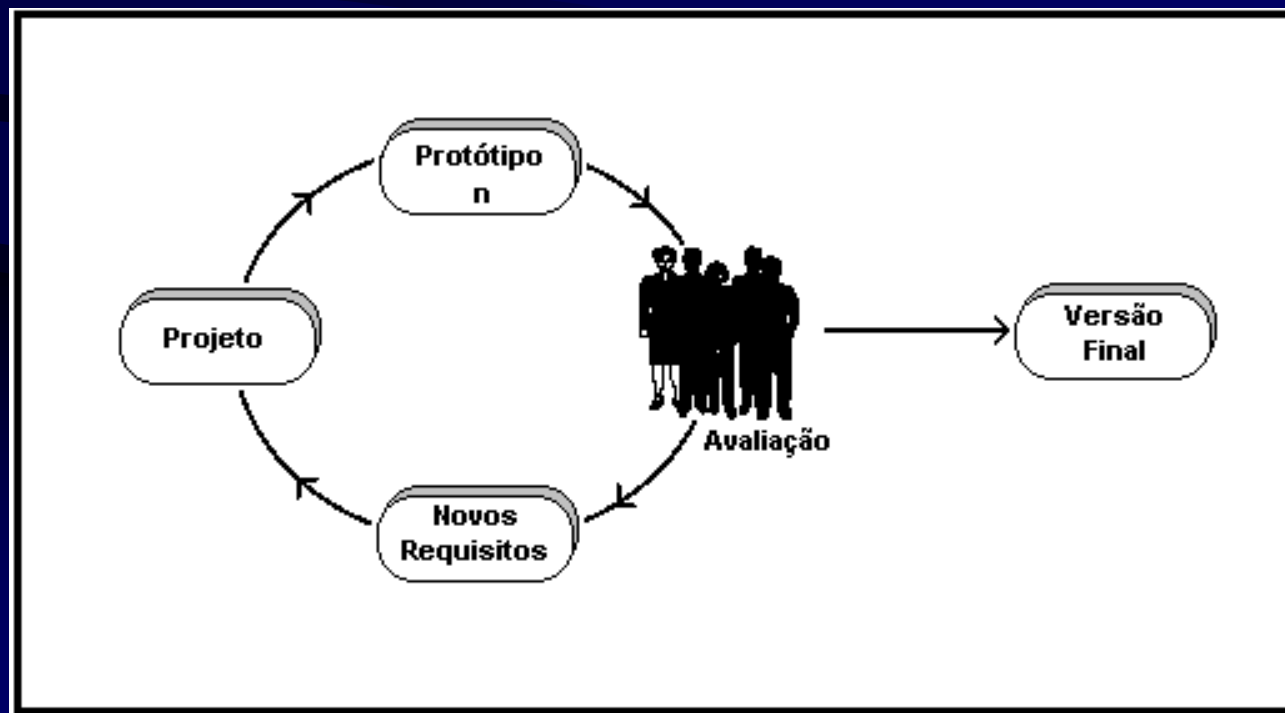
Tipos de Softwares educacionais

- Jogos



Etapas para o desenvolvimento do software educacional

- DEFINIÇÃO DO AMBIENTE DE APRENDIZAGEM;
- ANÁLISE DA VIABILIDADE;
- SELEÇÃO DO TIPO DE DOCUMENTO;
- PLANEJAMENTO DA INTERFACE;



Etapas para o desenvolvimento do software educacional

- PLANEJAMENTO DO DOCUMENTO:

- Seleção do Sistema de autoria e das ferramentas;

- Características pedagógicas;

- Facilidade de Uso;

- Interface;

- Adaptabilidade e

- Documentação.

A informática na terceira idade

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 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. (ABDALA, 2001).

Porque 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.

Orientação a Objetos: UML

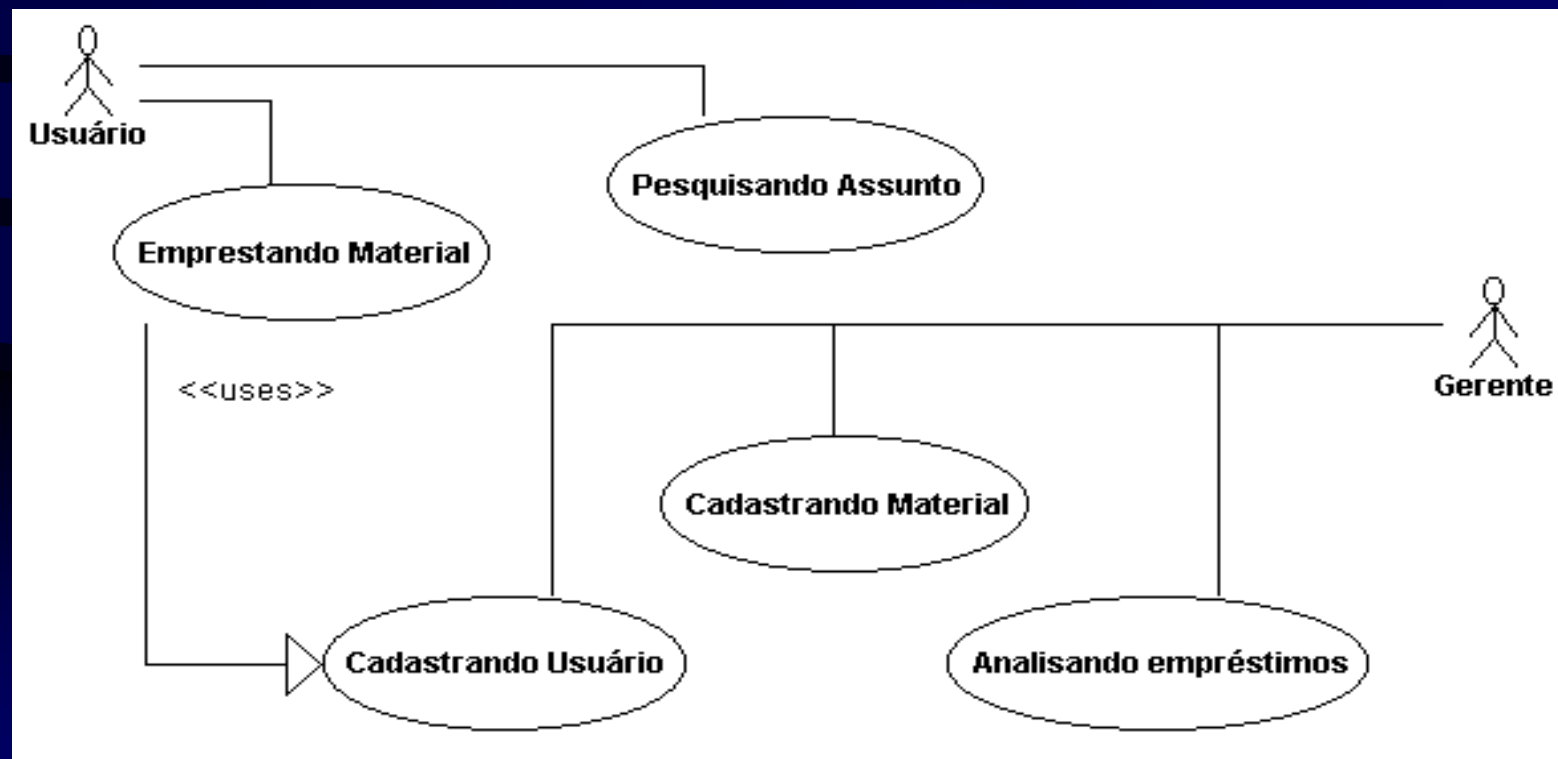
A UML é uma linguagem de modelagem e não um método. A maioria dos métodos consiste, pelo menos em princípio, de uma linguagem de modelagem e de um processo.

A 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).

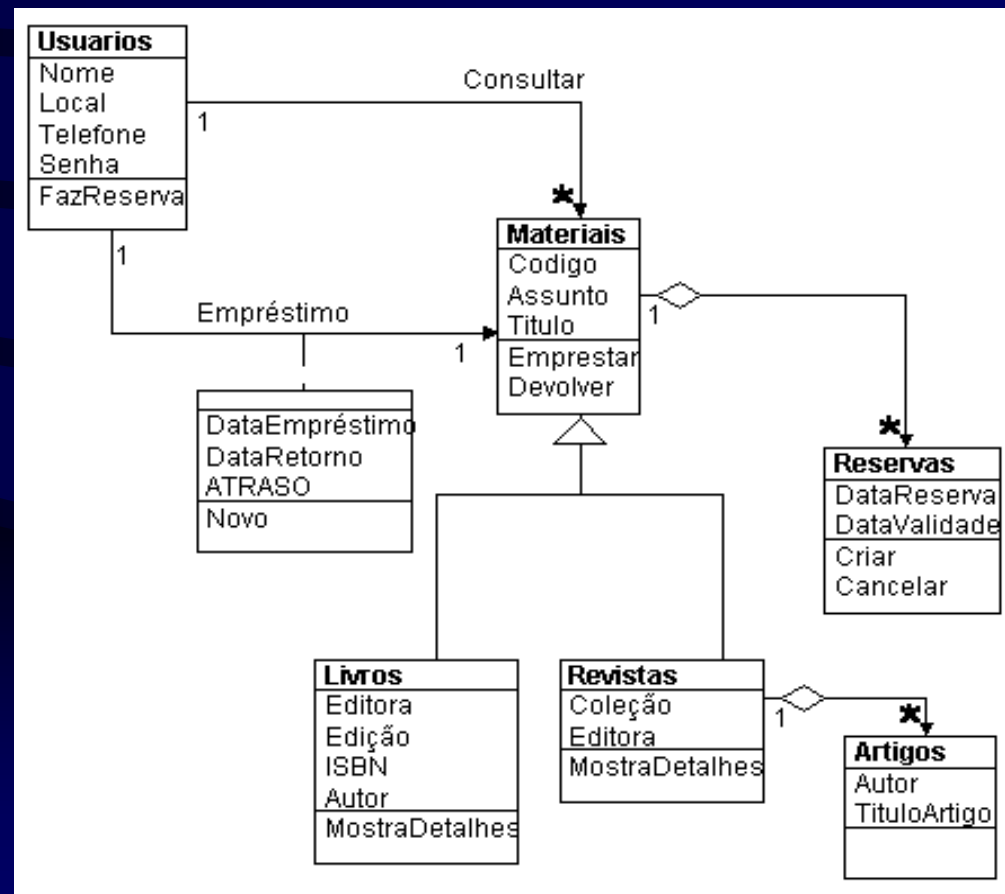
UML - Diferentes visões do sistema

Diagrama de Casos de uso: dão uma visão externa do sistema e dos objetivos que os atores externos tem do sistema.



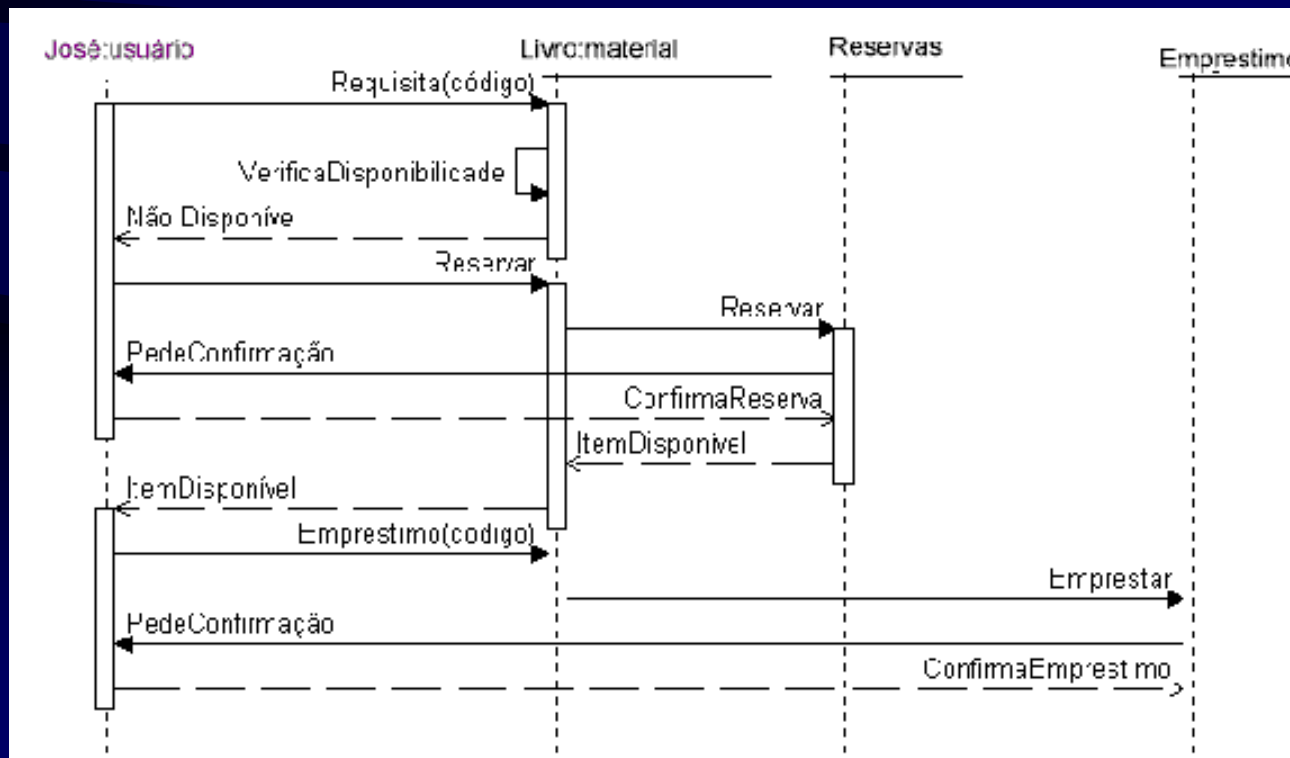
UML - Diferentes visões do sistema

Diagrama de Classes: tratam dos processos que ocorrem entre as estruturas que compõem o sistema e dão uma visão da dinâmica interna do sistema.



UML - Diferentes visões do sistema

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.



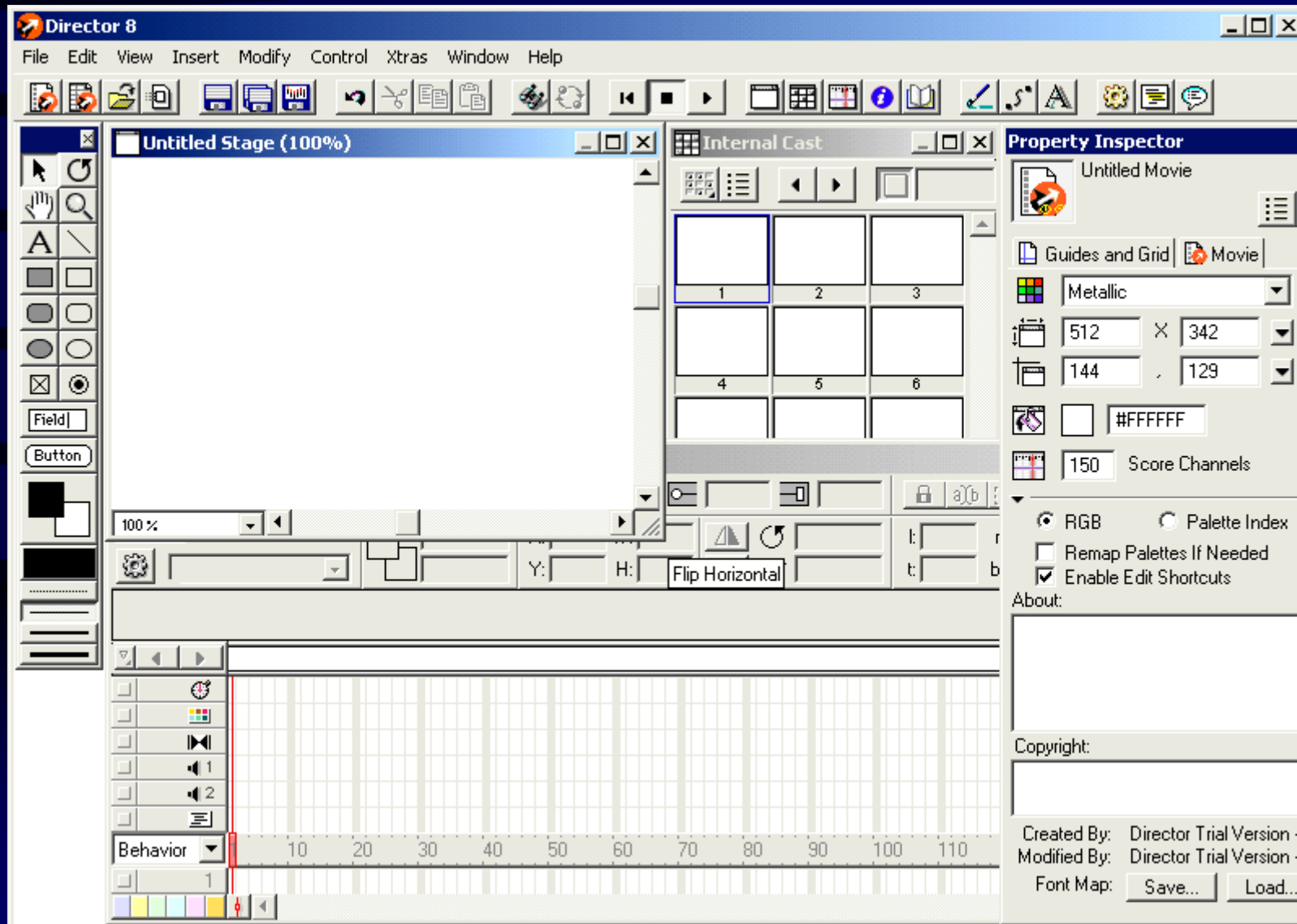
Ambientes de Desenvolvimento

Ambiente para Especificação do Sistema: RATIONAL ROSE



Ambientes de Desenvolvimento

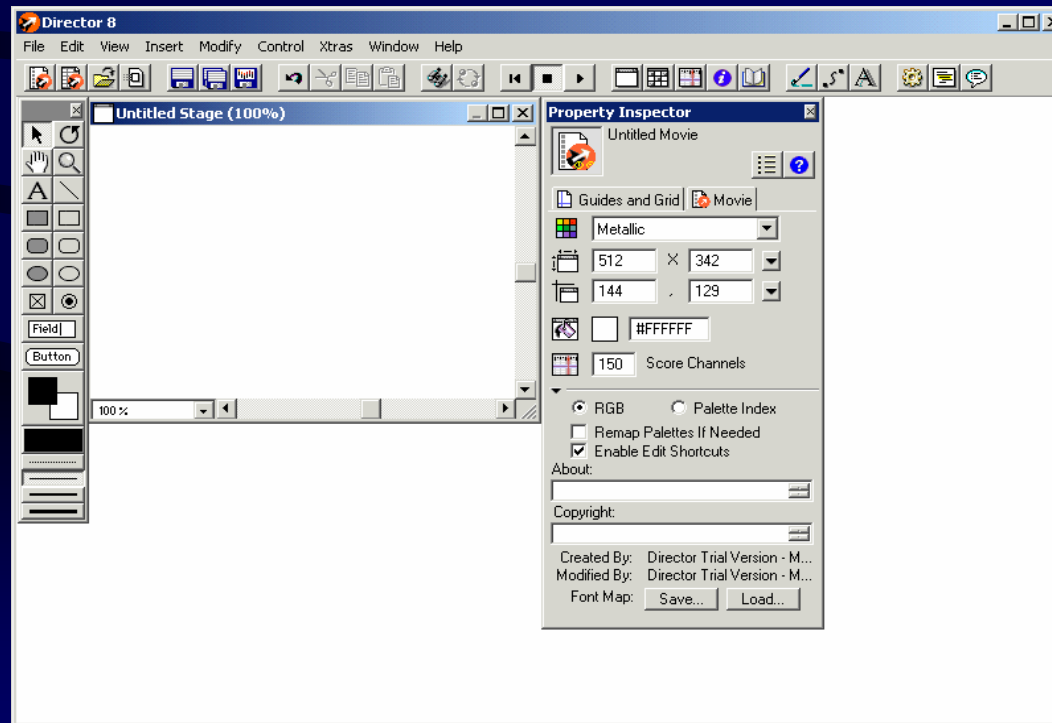
Ambiente de desenvolvimento: Director 8



Ambiente de Desenvolvimento: Director 8

A JANELA *STAGE*:

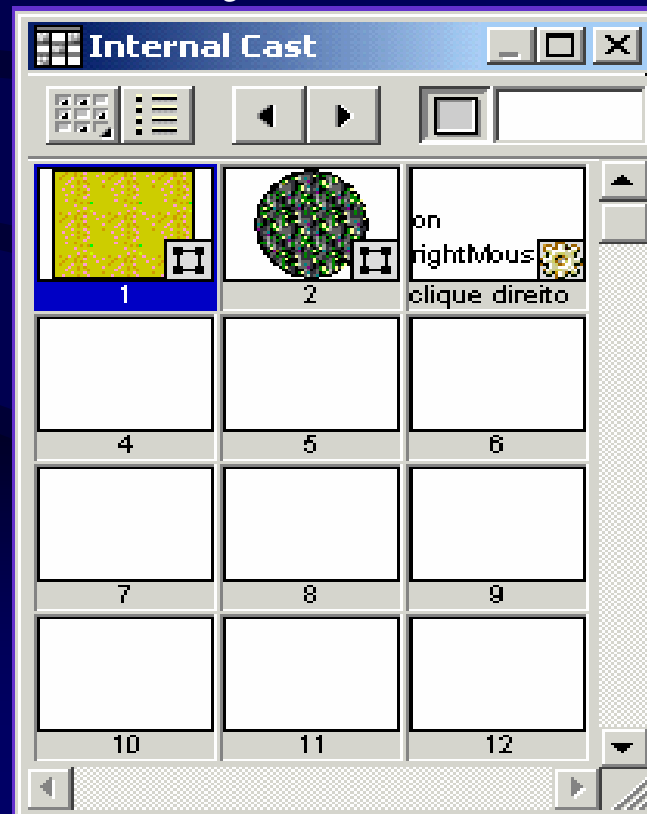
No *stage*, são colocados todos os itens (cenários, atores etc) que irão compor a cena a ser desenvolvida. No *stage* podem ser definidas as propriedades do filme atual (*Modify – Movies Properties*), tais como, tamanho do *stage* de apresentação, a localização, a paleta *Default*, entre outros (Bizzotto, 2000).



Ambiente de Desenvolvimento: Director 8

A JANELA *CAST*:

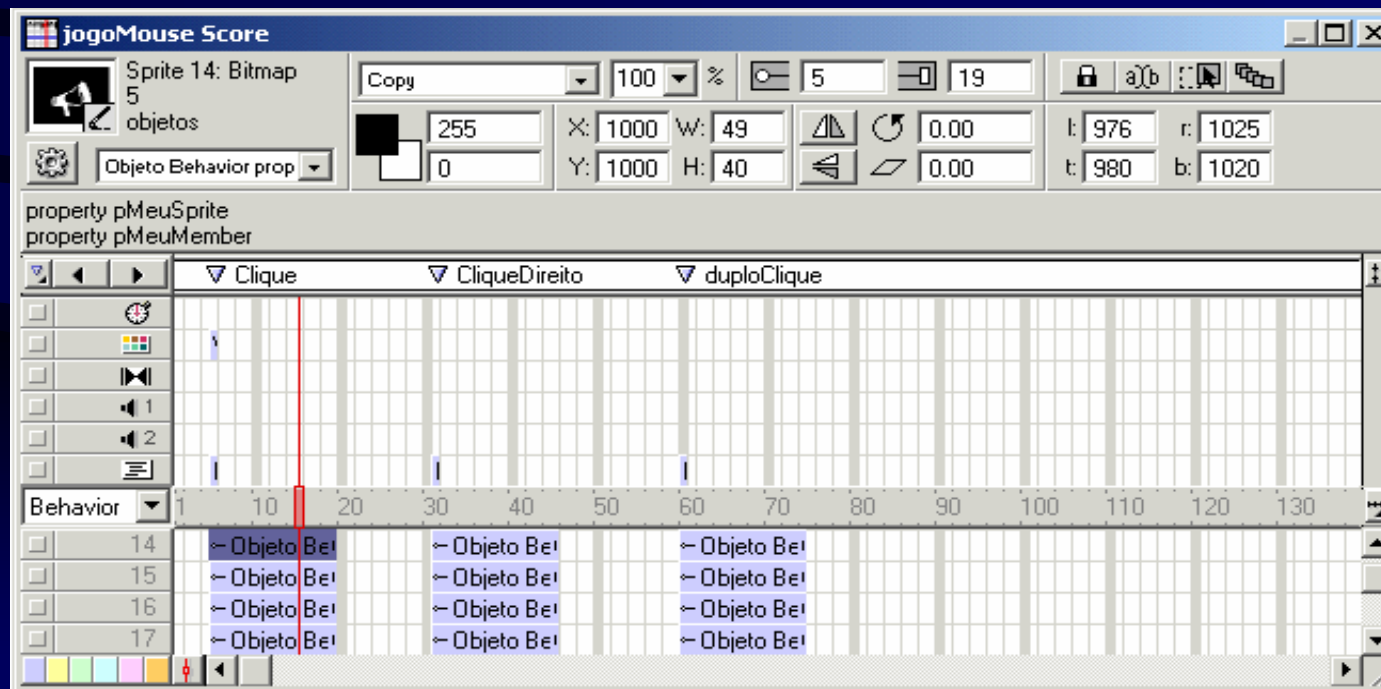
A Janela *Cast*, possui os *CastMembers*, que são os atores utilizados por um dado projeto, que atuarão sobre o *Stage*. Desta forma, qualquer imagem, som, vídeo etc, incluído no projeto, irá aparecer na janela *Cast*.



Ambiente de Desenvolvimento: Director 8

A JANELA *SCORE*:

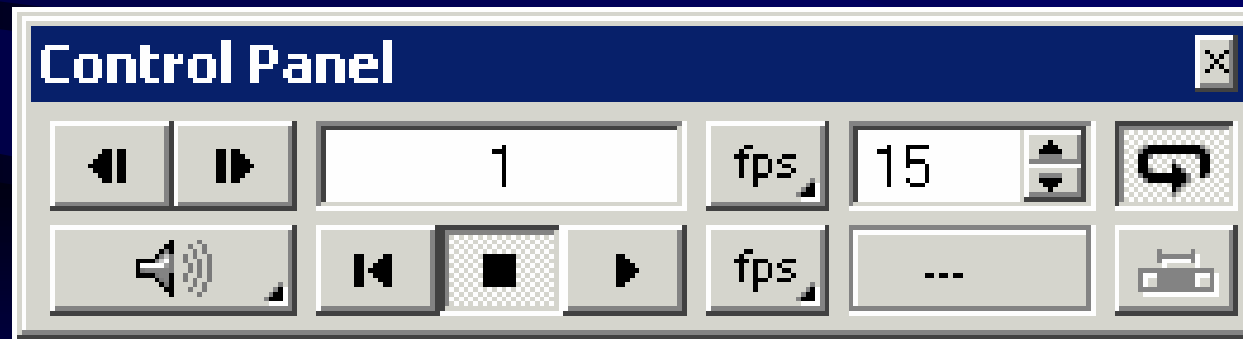
O *score* pode ser entendido como o roteiro (ou seqüência) do filme onde os *Cast Members* são organizados de acordo com o planejado. O *score* é dividido em linhas (canais) e colunas (*frames*).



Ambiente de Desenvolvimento: Director 8

A JANELA *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).



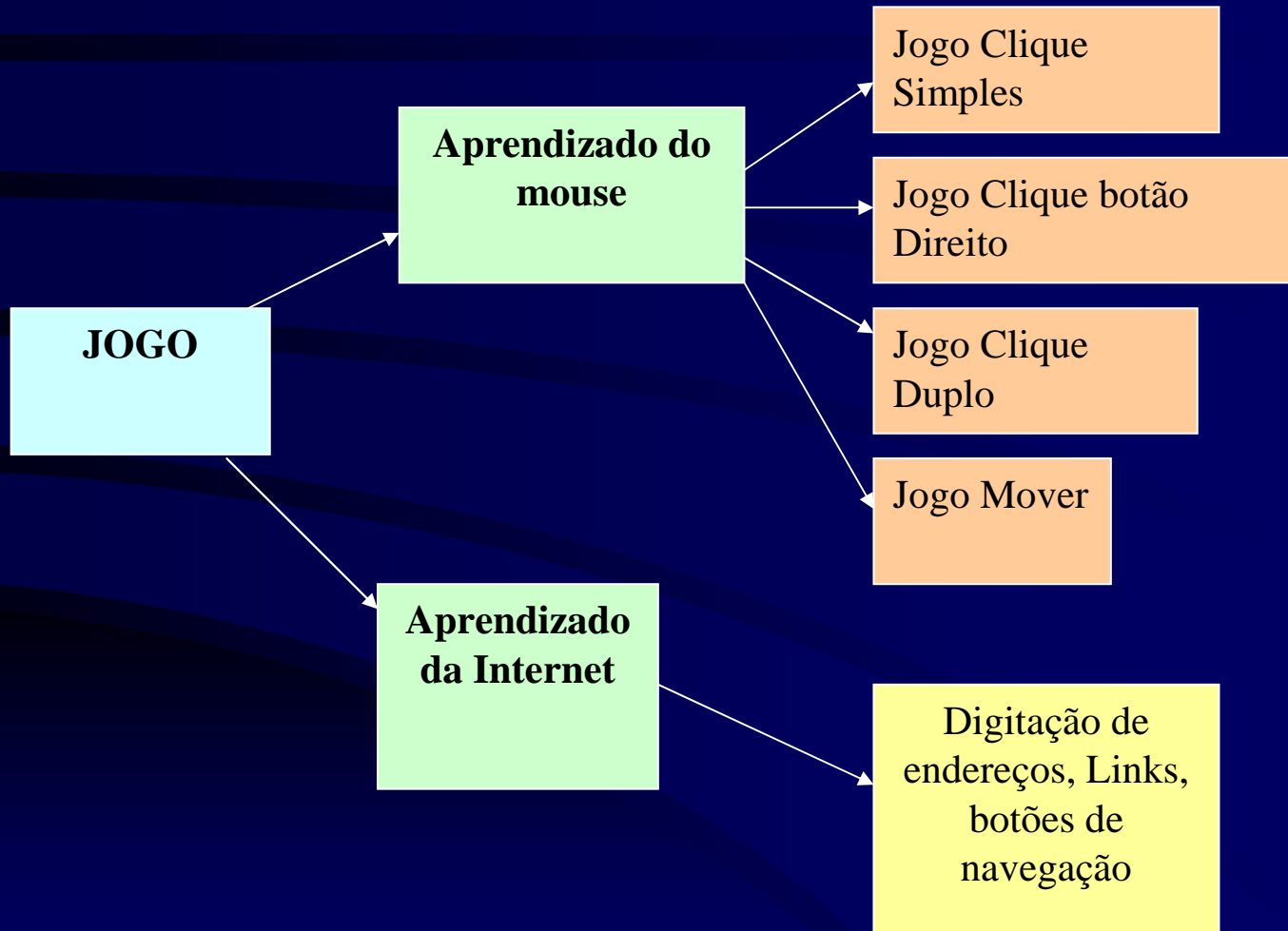
Ambiente de Desenvolvimento: Director 8

LINGUAGEM LINGO:

Lingo é o nome da linguagem utilizada pelo Director, para a produção mais efetiva junto ao usuário. O Lingo é uma linguagem de “*script*”, cuja sintaxe e construção se aproxima da língua inglesa em sua forma usual, ou seja, falada. Os *scripts* podem ser compostos de um único comando, assim como de uma seqüência deles, dispostos em conjuntos, similares a parágrafos. A complexidade varia de acordo com a aplicação (KEINER, 2001).

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).

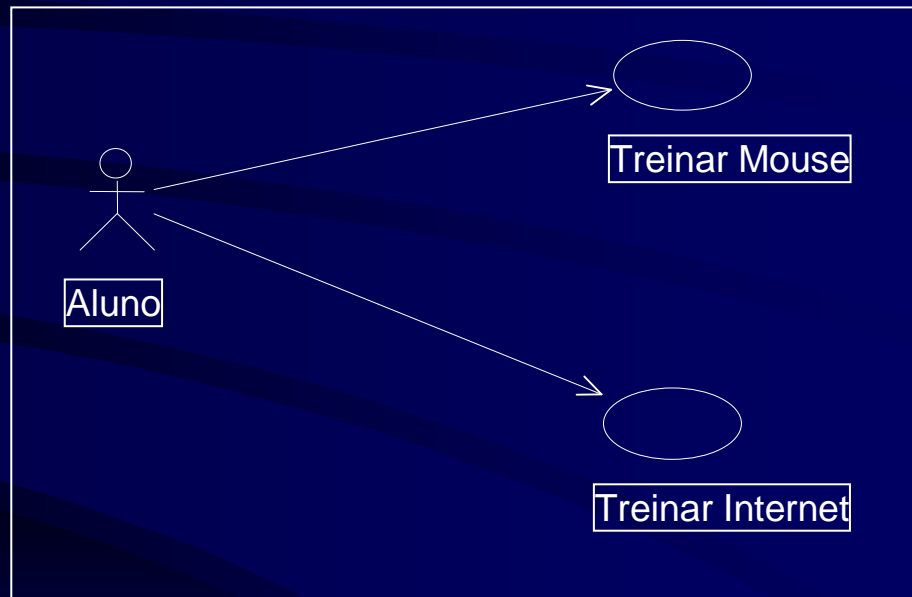
Software Proposto



Especificação do Software

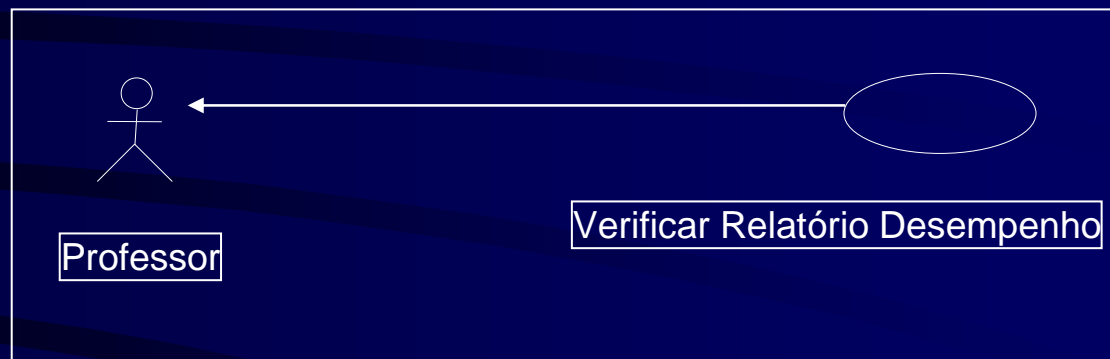
Diagramas de Casos de Uso

Diagrama de Caso de Uso: Aluno



Especificação do Software

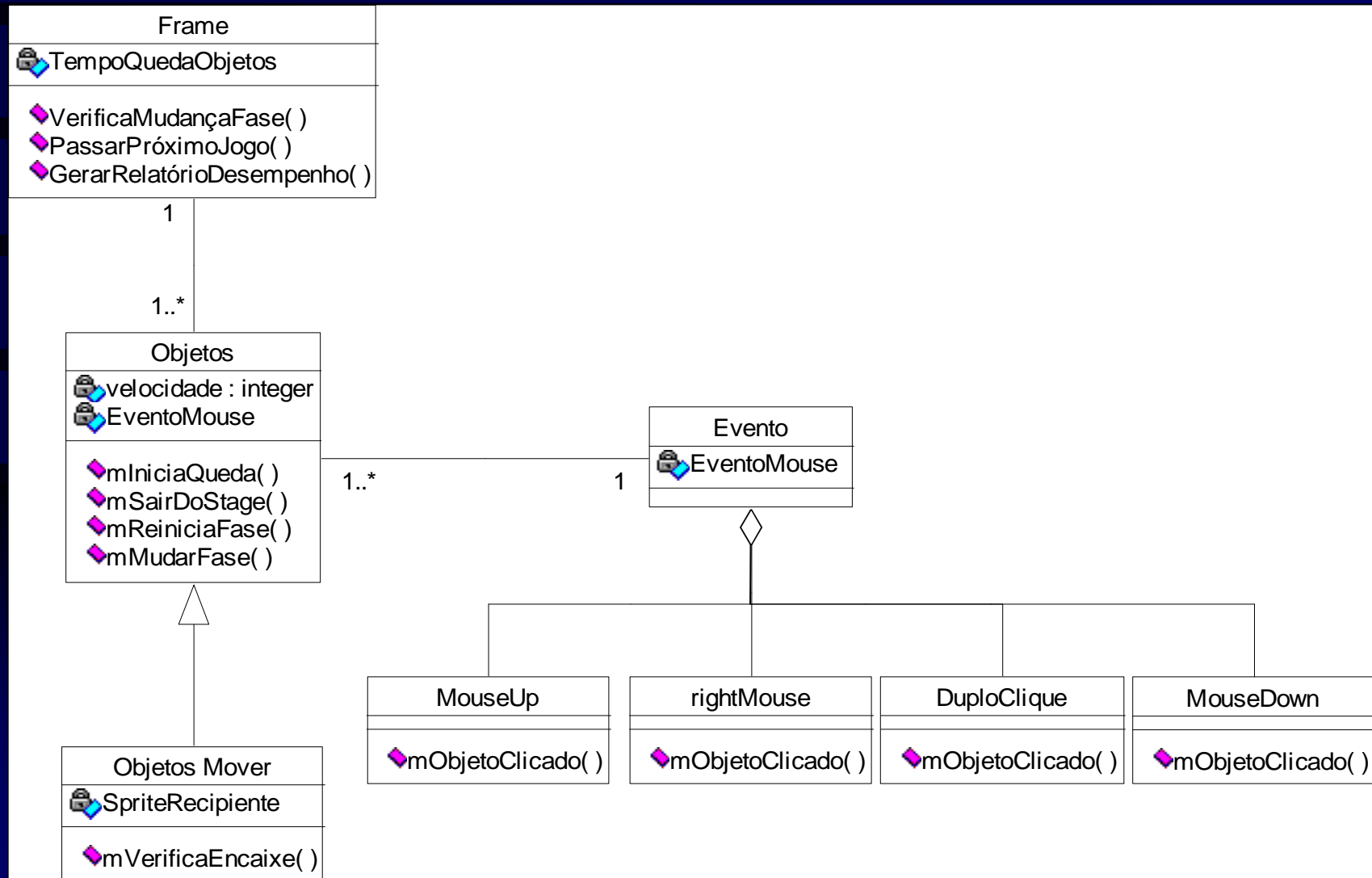
Diagrama de Caso de Uso: Professor



Especificação do Software

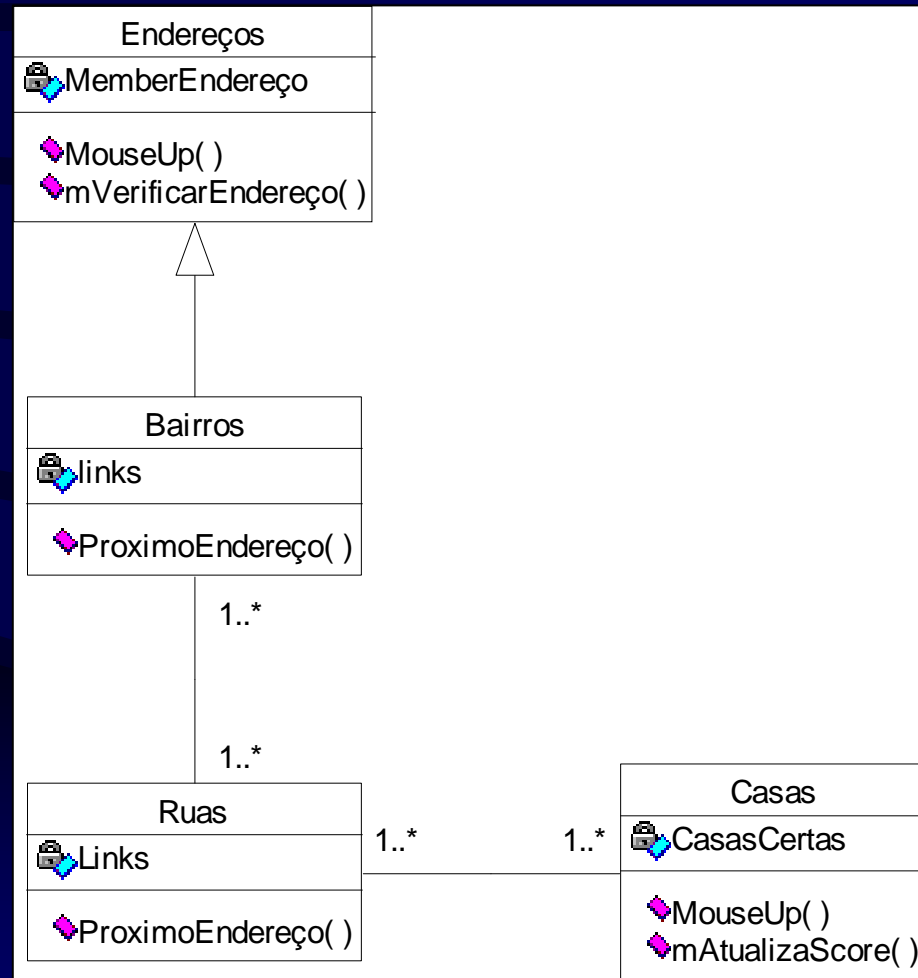
Diagramas de Classes

Diagrama de Classes: Jogo Mouse



Especificação do Software

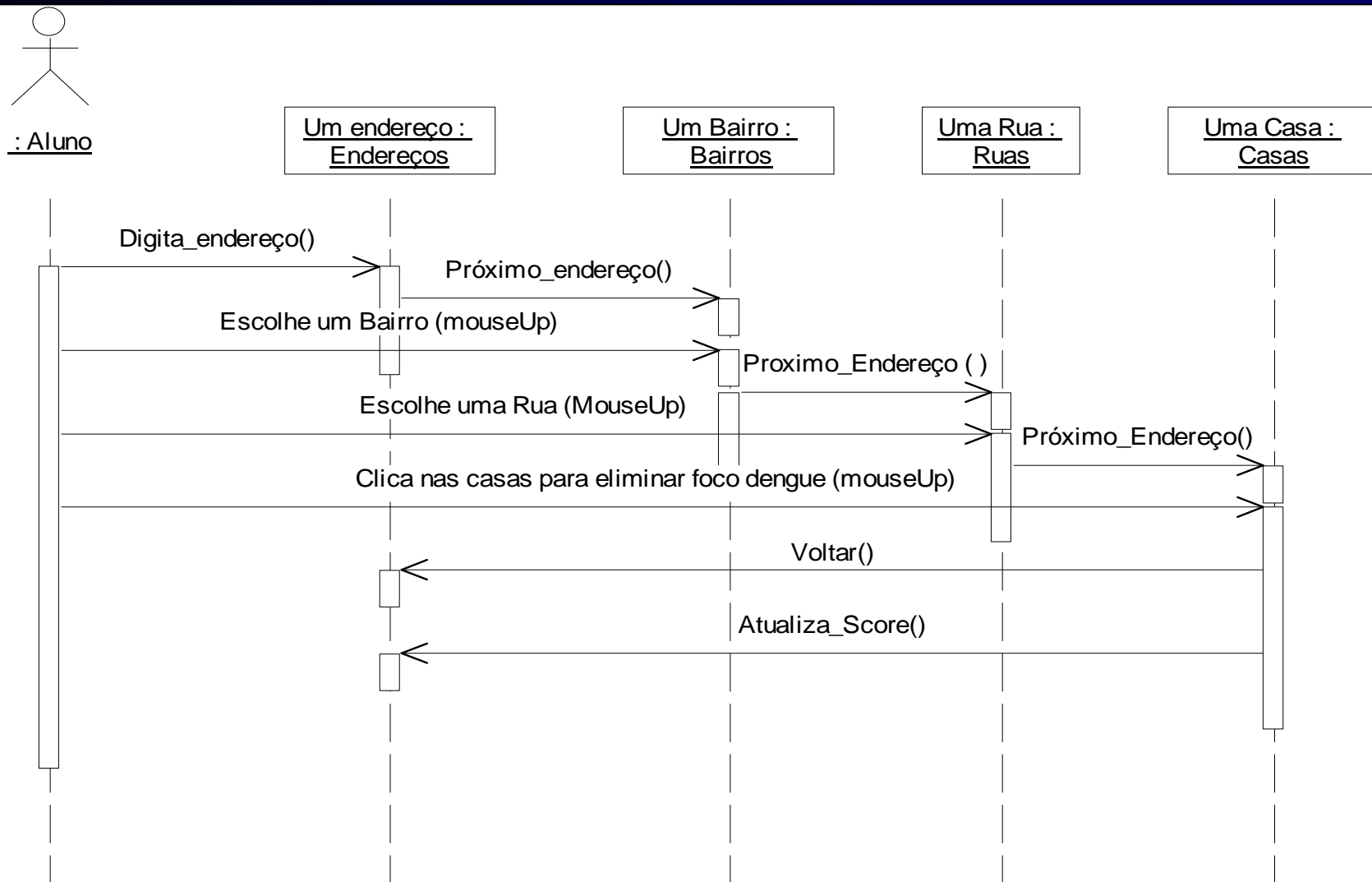
Diagrama de Classes: Jogo Internet



Especificação do Software

Diagramas de Seqüência de Eventos

Diagrama de Seqüência de Eventos: Jogo Internet



Especificação do Software

Diagrama de Seqüência de Eventos: Jogo Mouse

: Aluno

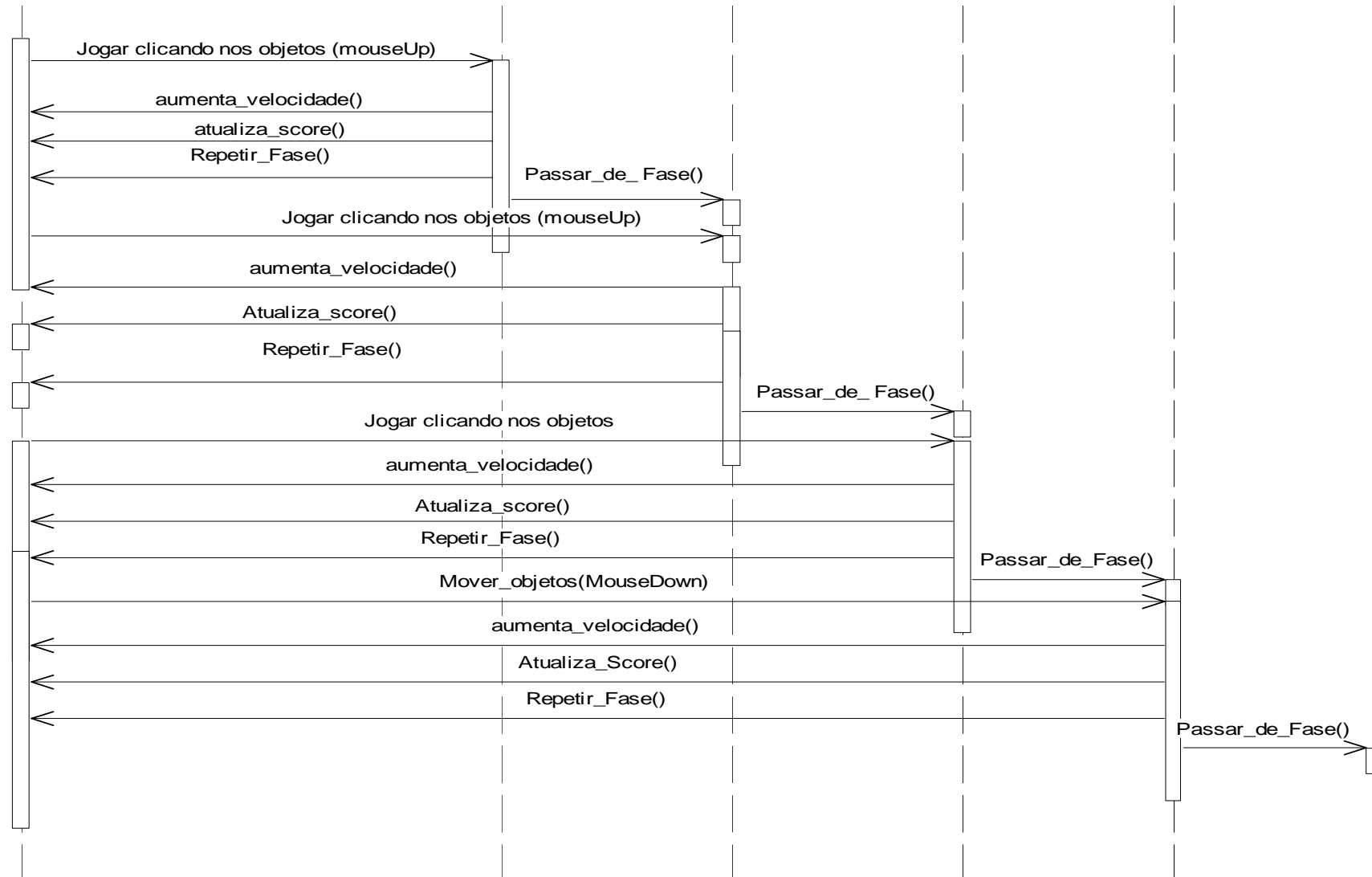
um Jogo Clique
Simples : MouseUp

um Jogo Clique
Direito : rightMouse

um Jogo Duplo
Clique : DuploClique

um Jogo Mover
objetos : MouseDown

um Jogo Internet:
Jogar



Implementação do Software

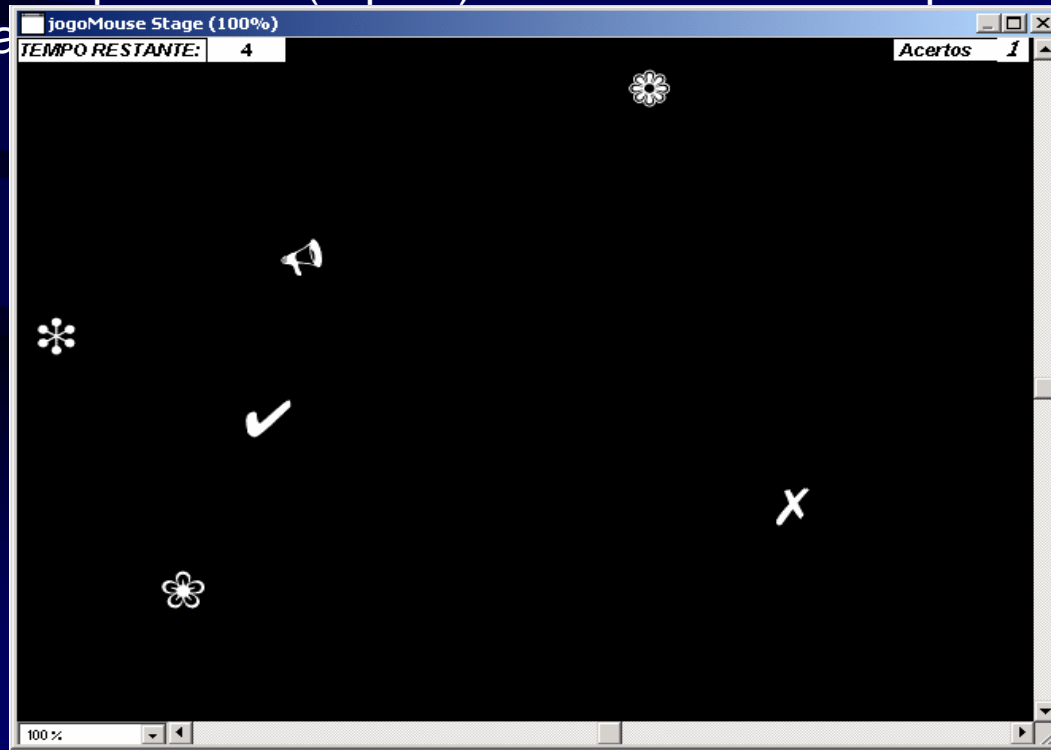
```
on mInicializaListaSprites(me)
```

```
-- Gerando uma lista vazia para acomodar os números  
dos sprites
```

```
pListaSprites = []
```

```
end mInicializaListaSprites
```

```
pListaSprites.add(tSprite) –Para adicionar os sprites a  
lista
```



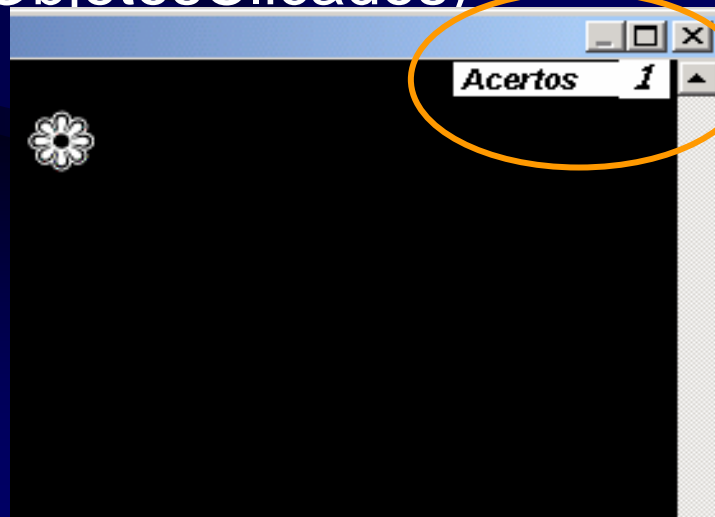
Implementação do Software

-- 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)
```



Implementação do Software

-- posicionando o *sprite* fora do *stage* (fora da tela)

```
sprite(pMeuSprite).loc = point(1000,1000)  
updateStage
```

-- reposicionando o *sprite* na lista de *sprites*
adiciona novamente o *sprite* a lista
pListaSprites.add(tSprite)

Implementação do Software

```
-- 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
```

Implementação do Software

-- 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]
```


Implementação do Software

-- criação da lista de eventos do mouse

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

Implementação do Software

-- 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
```



Implementação do Software

- 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.criciúma.com.br":  
  pEndereco = "criciúma"
```

- 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!"
```



Implementação do Software

-- 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.



```
on gravarDesempenhoAluno(me,tPropriedade,tValor)
```

```
  pNomeAluno = mObtemNomeAluno(script "quadroNegro")
```

```
  pNomeArquivo = "arquivoDesempenhoJogoMouseAluno" & pNomeAluno & ".txt"
```

```
  oFileXtra = xtra("FileXtra3").new()
```

```
  oFileIO = xtra("fileIO").new()
```

```
  tExiste = oFileXtra.fx_FileExists(pNomeArquivo)
```

```
  if tExiste then
```

```
    oFileXtra.fx_FileDelete(the moviePath & pNomeArquivo)
```

```
  end if
```

```
  oFileIO.createFile(the moviePath & pNomeArquivo)
```

```
  tLista = mObterListaDesempenhoAluno(script"quadroNegro")
```

```
  oFileIO.openFile(the moviePath & pNomeArquivo)
```

```
  oFileIO.writeString(string(tLista))
```

```
  oFileIO.closeFile()
```

```
end gravarDesempenhoAluno
```

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.

Extensões

Como sugestões para trabalhos futuros, pode-se citar:

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

Referências Bibliográficas

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

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.

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.

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

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.

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