

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

## **TUTORIAL PARA O APRENDIZADO NO AMBIENTE LOGO**

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

**FÁTIMA BAUMGÄRTNER NICOLETTI**

BLUMENAU, NOVEMBRO/2002

2002/2-26

# **TUTORIAL PARA O APRENDIZADO DO AMBIENTE LOGO**

**FÁTIMA BAUMGÄRTNER NICOLETTI**

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. Luiz Bianchi – Orientador na FURB

---

Prof. José Roque Voltolini da Silva — Coordenador do TCC

## **BANCA EXAMINADORA**

---

Prof. Luiz Bianchi

---

Prof. Paulo Roberto Dias

---

Prof. Joyce Martins

# DEDICATÓRIA

Dedico este trabalho de conclusão de curso à minha família, em especial aos meus pais Bernardo e Iracema e meu esposo José Carlos, pela dedicação, amor, carinho e apoio que fizeram que eu chegasse até aqui.

# AGRADECIMENTOS

Ao meu orientador, Professor Luiz Bianchi, pela paciência, orientação e apoio. Seu auxílio foi muito importante para a realização deste trabalho.

Aos professores do Departamento de Sistemas e Computação da FURB, pelos conhecimentos passados ao longo do curso.

Aos meus amigos que entenderam minha ausência por estar dedicando o tempo à conclusão deste trabalho.

A Deus e a todos que, direta ou indiretamente, ajudaram para tornar o meu sonho uma realidade.

# SUMÁRIO

LISTA DE ABREVIATURAS.....	VII
LISTA DE FIGURAS .....	VIII
LISTA DE QUADROS .....	IX
RESUMO .....	X
1 INTRODUÇÃO.....	1
1.1 OBJETIVOS.....	2
1.2 ESTRUTURA .....	2
2 INFORMÁTICA NA EDUCAÇÃO.....	3
2.1 TIPOS DE <i>SOFTWARES</i> UTILIZADOS NO ENSINO .....	4
2.1.1 TUTORIAIS.....	5
2.1.2 MODELAGEM E SIMULAÇÃO.....	7
2.1.3 JOGOS .....	7
2.1.4 EXERCÍCIO-E-PRÁTICA .....	8
2.2 <i>SOFTWARE</i> EDUCACIONAL.....	8
2.2.1 CARACTERÍSTICAS DO <i>SOFTWARE</i> EDUCACIONAL .....	9
2.2.2 VANTAGENS DO <i>SOFTWARE</i> EDUCACIONAL .....	9
2.2.3 TAXONOMIA DE <i>SOFTWARE</i> EDUCACIONAL.....	10
2.3 EVENTOS INSTRUCCIONAIS .....	14
2.3.1 OS NOVES EVENTOS INSTRUCCIONAIS .....	15
2.4 MULTIMÍDIA .....	17
2.4.1 CARACTERÍSTICAS DO <i>SOFTWARE</i> MULTIMÍDIA.....	18
3 DESENVOLVIMENTO DO TUTORIAL .....	19
3.1 ESPECIFICAÇÃO .....	20

3.1.1 PROJETO CONCEITUAL .....	21
3.1.2 PROJETO NAVEGACIONAL.....	23
3.1.3 PROJETO DA INTEFACE .....	25
3.2 IMPLEMENTAÇÃO .....	26
4 CONCLUSÃO .....	35
4.1 SUGESTÕES .....	35
REFERÊNCIAS BIBLIOGRÁFICAS .....	36

## LISTA DE ABREVIATURAS

AEAC - Aprendizagem de habilidades cognitivas amplas

CAI - *Computer Assisted Instruction*

IA - Inteligência Artificial

ICAI - *Intelligence Computer Assisted Instruction*

NIED – Núcleo Interdisciplinar de Informática Aplicada à Educação

OOHDM - *Object Oriented Hypermedia Design Method*

STI - Sistemas Tutores Inteligente

UNICAMP – Universidade de Campinas

## LISTA DE FIGURAS

Figura 1 - Taxonomia para programas educacionais.....	11
Figura 2 – Ciclo de aprendizagem utilizado pelo tutorial desenvolvido .....	19
Figura 3 – Módulos do tutorial.....	20
Figura 4 – Diagrama de Caso de Uso.....	21
Figura 5 – Diagrama Projeto Conceitual.....	23
Figura 6 – Modelo Navegacional .....	24
Figura 7 – Apresentação dos módulos de aprendizagem .....	26
Figura 8 – Barra de navegação do tutorial .....	27
Figura 9 – Animação dos itens principais, parte I.....	27
Figura 10 – Animação dos itens principais, parte II.....	28
Figura 11 – Animação para editar figura, parte I .....	29
Figura 12 – Animação para editar figura, parte II.....	29
Figura 13 – Animação para editar figura, parte III.....	30
Figura 14 – Animação para editar figura, parte IV .....	30
Figura 15 – Animação para editar figura, parte V.....	31
Figura 16 – Animação para editar figura, parte VI .....	31



## LISTA DE QUADROS

Quadro1 - Eventos instrucionais conforme as modalidades de <i>softwares</i> educacionais.....	15
Quadro 2 - Funcionalidades das operações do projeto de interface .....	25
Quadro 3 - Parte do código referente às propriedades Tipo, Seção e Arquivo.....	32
Quadro 4 - Parte do código referente a função abrir janela.....	33
Quadro 5 - Parte do código referente a função fechar janela.....	34

## **RESUMO**

O presente trabalho tem por objetivo o desenvolvimento de um tutorial para auxiliar no aprendizado do ambiente LOGO, abordando, para chegar a este fim, um breve estudo sobre a teoria de eventos instrucionais de Robert M. Gagné e sua influência na informática na educação. Apresenta também uma breve descrição de cada um dos diferentes tipos de softwares, assim como suas vantagens e desvantagens.

# **ABSTRACT**

The present work has for objective the development of a tutorial one to assist SOON in the learning of the environment, approaching, to arrive at this end, a briefing study on the theory of instruccionalis events of Robert M. Gagné and its influence in computer science in the education. It also presents one brief description of each one of the different types of softwares, as well as its advantages and disadvantages.

# 1 INTRODUÇÃO

Hoje em dia o computador é uma ferramenta que está sendo utilizada nos diversos setores da sociedade. Na educação a realidade não é diferente, tendo mostrado o quanto pode ser útil. Existem muitos títulos de *softwares* para os mais diversos fins, desde *softwares* para auxiliar na alfabetização de crianças até tutoriais multimídia.

Entretanto, a utilização destes *softwares* no processo de ensino-aprendizagem das instituições de ensino é muito restrita, visto que os professores possuem os mais variados sentimentos com relação ao assunto, incluindo a satisfação de estar participando de uma realidade tecnológica, até pouco tempo futurística; a ansiedade por descobrir “tudo o que esta máquina pode fazer”; a sensação de não “levar jeito” com essas coisas ou, ainda, o medo de enfrentar as mudanças que chegam com a Informática Educativa (Weiss, 1998).

Desta forma, os professores optam por uma das formas mais procuradas para a aplicação da multimídia e hipermídia que são os tutoriais.

Um tutorial é um *software* no qual a informação é organizada de acordo com uma seqüência pedagógica particular e apresentada ao estudante seguindo essa seqüência, ou então o aluno pode escolher a informação desejada através do *mouse* ou teclado. No primeiro caso, o *software* tem o controle da situação de ensino e do que pode ser apresentado ao aprendiz, isto é, o aprendiz pode mudar de tópicos simplesmente apertando a tecla *ENTER* ou o *software* altera a seqüência de acordo com as respostas dadas pelo aprendiz. Na outra situação, o aprendiz tem o controle e pode escolher o que deseja ver. Em geral, *softwares* que permitem ao aprendiz controlar a seqüência da informação são organizados em forma de hipertextos e o aprendiz pode “navegar” entre itens de informação.

Em ambos os casos, o conteúdo que está disponível ao aluno, foi definido e organizado previamente. Ele está restrito a esse conteúdo e o computador assume o papel de uma máquina de ensinar. A interação do aluno consiste na leitura da tela ou na escuta da informação fornecida, no avanço do material, na escolha da informação e nas respostas das possíveis perguntas que o tutorial pode conter (Brignoli, 1995).

Diante do exposto, o presente trabalho visa o desenvolvimento de um tutorial multimídia que permita que qualquer pessoa possa aprender o ambiente LOGO. O tutorial desenvolvido tem por objetivo fazer com que a pessoa aprenda o conteúdo previsto para o ambiente LOGO de forma interativa, recebendo orientação constante fornecida pelo próprio

tutorial ao longo de seu processo de aprendizagem. O tutorial desenvolvido tem por finalidade contribuir para o aprendizado, mostrando que qualquer pessoa é capaz de aprender de forma diferente dos métodos tradicionais e de modo mais produtivo e eficaz, uma vez que proporciona maior estímulo para o aprendizado.

Através do presente trabalho, pretende-se contribuir para melhoria dos métodos de ensino e abrir caminho para novos projetos de conciliação da informática com a educação.

## 1.1 OBJETIVOS

O presente trabalho tem como objetivo desenvolver um tutorial multimídia que permita o aprendizado no ambiente LOGO, disponibilizando-o em CD-ROM.

Os objetivos específicos do trabalho são:

- a) permitir que o usuário repita os procedimentos ensinados;
- b) incluir exercícios que permitam à pessoa ampliar o conhecimento sobre um dado tópico do conteúdo;
- c) permitir a interatividade com o tutorial como se o aluno estivesse utilizando o próprio ambiente LOGO;
- d) incluir animações explicativas de cada lição.

## 1.2 ESTRUTURA

Este trabalho está dividido em quatro capítulos. O primeiro capítulo aborda os objetivos do trabalho, suas justificativas, limitações e sua organização.

O segundo capítulo fala sobre informática na educação, aborda conceitos e características sobre multimídia e também trata do *software* educacional e suas taxonomias.

O desenvolvimento do tutorial, incluindo a especificação, a implementação e o funcionamento, estão descritos no terceiro capítulo.

Finalmente, no quarto capítulo, tem-se as conclusões, considerações finais e sugestões para trabalhos futuros.

## 2 INFORMÁTICA NA EDUCAÇÃO

Formalmente, a informática na educação foi introduzida no Brasil pela Universidade de Campinas (UNICAMP), em 1973. Naquele ano, a UNICAMP criou um grupo de pesquisa interdisciplinar nas áreas de computação, lingüística e psicologia educacional do qual resultou a elaboração do Projeto LOGO (MEC, 2001).

O Projeto LOGO foi operacionalizado a partir de 1978, ano em que, de fato, a informática chegou à 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 à Educação (NIED).

O grande avanço tecnológico atual e as redes de computadores, em especial a Internet, aumentam a influência da informática na educação trazendo novas alternativas além do eficiente ambiente LOGO, permitindo conectar pessoas espalhadas pelo mundo todo. A tecnologia de computadores tem sido o novo impulso e a nova promessa para um entendimento mais amplo da Educação, viabilizando funções em que não só os alunos, mas os próprios professores possam desenvolver suas atividades de modo colaborativo. Por intermédio da Internet, os alunos têm a chance de acessar e explorar novas bases de dados, conhecendo novas e diferentes realidades, acumulando conhecimentos e informações que, mais tarde, serão refletidas e estudadas ajudando a aquisição do conhecimento (Baranauskas, 1999).

Tajra (1998) enfatiza que o uso da informática dentro de um ambiente educacional varia de acordo com a proposta pedagógica que está sendo utilizada em cada caso. É muito importante que as pessoas incorporadas nestes projetos estejam dispostas a novos desafios. As vantagens freqüentemente encontradas por Tajra (1998) são:

- a) os alunos ganham autonomia nos trabalhos, podendo desenvolver boa parte das atividades sozinhos, dentro de suas características pessoais;
- b) em função da gama de ferramentas disponíveis nos *softwares*, os alunos, além de ficarem mais motivados, também, tornam-se mais criativos;
- c) os ambientes tornam-se mais dinâmicos e ativos, os alunos que sobressaem pelo uso da tecnologia costumam ajudar aqueles que estão com dificuldades;
- d) as aulas expositivas perdem espaço para os trabalhos cooperativos;

- e) estímulo a uma forma de comunicação voltada para a realidade atual de globalização;
- f) a maioria dos programas disponíveis no mercado são em outros idiomas, sendo que esta característica em si não deve ser vista como empecilho, mas como uma motivação para o aprendizado de novos idiomas;
- g) além da escola poder direcionar as fontes de pesquisas para os recursos já existentes, tais como livros, enciclopédias, revistas, jornais e vídeos, ela pode optar por mais uma fonte de aprendizagem: o computador;
- h) uma grande contribuição da informática é o auxílio para o desenvolvimento das habilidades de comunicação e de estrutura lógica do pensamento.

Contudo, apesar dessa aparente evolução, pode-se considerar que a educação ainda está precária quanto à utilização de computadores como ferramenta para a aprendizagem. Parte das dificuldades para a utilização de recursos de informática na educação e a resistência dos professores em se adequar às novas tecnologias, está ligada às tentativas de implantação da informática nas atividades escolares (Tajra, 1998).

Cabe lembrar que o livro foi um dos primeiros instrumentos tecnológicos inclusos no processo de ensino-aprendizagem, causando na época muitas alterações educacionais. No entanto, hoje está totalmente incorporado no processo. Assim, embora a implantação da informática na área da educação ainda seja recente, e muitos questionem os efeitos de sua utilização, não existe possibilidade de não utilizá-la (Tajra, 1998).

## **2.1 TIPOS DE SOFTWARES UTILIZADOS NO ENSINO**

A aprendizagem, segundo Valente (1999), pode ocorrer basicamente de duas maneiras: a informação é memorizada ou é processada pelos esquemas mentais. Neste último caso, o conhecimento é construído. Essas diferenças em aprender são fundamentais, pois no primeiro caso significa que a informação não foi processada e, portanto, não está passível de ser aplicada em situações de resoluções de problemas e desafios.

Essa informação, quando muito, pode ser repetida de maneira mais ou menos fiel indicando uma fidelidade na retenção da mesma. Por outro lado, o conhecimento construído está incorporado aos esquemas mentais que são colocados para funcionar diante de situações

problema ou desafios. Neste caso, o aluno deve aplicar o conhecimento adquirido resolvendo o problema ou buscar novas informações que o ajudem na solução da tarefa.

Assim, o mecanismo de construção do conhecimento pressupõe a existência de estruturas mentais ou de conhecimento organizado, que podem ser observados em comportamentos (habilidades) ou declarações (linguagem). Pressupõe-se o princípio da continuidade – um novo conhecimento deve estar relacionado com o que já se conhece. Aprender significa enriquecer essas estruturas por meio de adição de novos conhecimentos ou através da reorganização das estruturas por meio do pensar e refletir.

Valente (1999) afirma que o computador pode ser um importante recurso para transmitir a informação ao usuário ou facilitar o processo de construção do conhecimento. No entanto, o aprender não está relacionado estritamente ao *software*, mas sim à interação aluno-*software*. O nível de compreensão está relacionado com o nível de interação que o aprendiz tem com o objeto e não com o objeto em si.

Alguns *softwares* apresentam características que favorecem a compreensão; outros onde certas características não estão presentes, requerem um maior envolvimento do professor, criando situações complementares ao *software* de modo a favorecer a compreensão, como no caso do tutorial. No entanto, cada um dos diferentes *softwares* usados na aprendizagem apresentam características que podem favorecer, de maneira mais ou menos explícita, o processo de construção do conhecimento (Valente, 1999).

A seguir serão apresentados alguns tipos de *softwares* educativos e recursos da informática utilizados no processo de construção de conhecimento pelo aluno.

### **2.1.1 TUTORIAIS**

Um tutorial é um *software* no qual a informação é organizada de acordo com uma seqüência pedagógica particular e apresentada ao estudante seguindo essa seqüência, ou então o aluno pode escolher a informação desejada através da interação do *mouse* ou teclado. Na primeira situação, o software tem o controle do que pode ser apresentado ao aprendiz: o aprendiz pode mudar de tópicos simplesmente apertando a tecla *ENTER* ou o *software* altera a seqüência de acordo com as respostas dadas pelo aprendiz. Na outra situação, o aprendiz tem o controle e pode escolher o que deseja ver. Em geral, *softwares* que permitem ao aprendiz controlar a seqüência da informação são organizados em forma de



hipertexto. Em ambos os casos, o conteúdo que está disponível ao aluno foi definido e organizado previamente. Ele está restrito a esse conteúdo e o computador assume o papel de uma máquina de ensinar. A interação do aluno consiste na leitura da tela ou na escuta da informação fornecida, no avanço do material, na escolha da informação e nas respostas das possíveis perguntas que o tutorial pode conter. Observando este comportamento, vê-se que o aprendiz está fazendo coisas, mas não se tem qualquer pista sobre o processamento dessa informação e se o aprendiz está entendendo o que está fazendo. Ele pode estar processando a informação fornecida, mas não existem meios para certificar se isso está acontecendo.

Para verificar se a informação foi ou não processada, é necessário apresentar ao aprendiz situações problema, onde ele é obrigado a usar as informações fornecidas. Alguns tutoriais tentam fazer isto, mas, em geral, o problema se resume em verificar se o aprendiz memorizou ou apenas reteve uma aplicação direta da informação.

A limitação do tutorial está justamente na capacidade de verificar se a informação foi processada pelo aluno e, portanto, passou a ser conhecimento agregado aos esquemas mentais. Por exemplo, é difícil um tutorial ter condições de corrigir a solução de um problema aberto com mais de um tipo de solução, onde o aprendiz pode exercitar sua criatividade e explorar diferentes níveis de compreensão de um conceito. A maioria dos tutoriais não é desenvolvida com essa intenção. Nesse caso, a tarefa de verificar se houve ou não construção de conhecimento ou se a solução do problema é criativa, ainda tem que ser feita pelo professor. No entanto, é difícil para o professor estar ao lado do aprendiz e seguir o que ele faz e o que está pensando enquanto usa um tutorial.

Portanto, os tutoriais e os *softwares* do tipo exercício-e-prática enfatizam a apresentação das lições ou de exercícios e a ação do aprendiz se restringe a virar páginas de um livro eletrônico ou realizar exercícios, cujo resultado pode ser avaliado pelo próprio computador. Essas atividades podem facilmente ser reduzidas ao fazer, ao memorizar informação, sem exigir que o aprendiz compreenda o que está fazendo. Cabe ao professor interagir com o aluno e criar condições para tê-la ao nível da compreensão, como, por exemplo, propor problemas para serem resolvidos e verificar se a resolução está correta. O professor nesse caso deve criar situação para o aluno manipular as informações recebidas, de modo que elas sejam transformadas em conhecimento (Valente, 1999).

## 2.1.2 MODELAGEM E SIMULAÇÃO

Conforme Rocha (1999), modelagem, do ponto de vista educacional, é uma técnica bastante comum, utilizada para se estudar o comportamento de muitos fenômenos reais. O processo de se modelar um fenômeno real ou hipotético para se observar / analisar seu comportamento no tempo, consiste em três fases principais:

- a) construção de um modelo que represente aspectos relevantes do sistema sendo estudado;
- b) experimentação e análise do modelo criado;
- c) comparação do modelo construído com sistemas.

Chama-se simulação a parte do processo de modelagem que envolve basicamente a fase de execução do modelo e análise dos resultados. Esses sistemas têm embutido um modelo do domínio e o usuário experimenta com o fenômeno modelado, alterando os parâmetros de entrada e observando / analisando os resultados da simulação (saídas).

Sistemas computacionais para modelagem e simulação podem consistir ambientes de aprendizado poderosos, por envolver o aluno no ciclo básico de expressão, avaliação e reflexão sobre o domínio considerado. Considerando um cenário típico de modelagem e simulação, o usuário constrói um modelo do fenômeno / objeto que deseja estudar com parâmetros e características definidas por ele, fornecidas em geral por um editor de modelos presente no ambiente computacional. Construído o modelo, o sistema o executa (simula) e apresenta os resultados (saídas) da simulação, em geral por meio de representações gráficas, animações, e outros. O usuário, no caso o aluno, pode então observar e analisar os resultados obtidos efetuando mudanças, se necessário, em seu modelo original (Rocha, 1999).

Para um melhor entendimento sobre o tipo de software simulação, podemos dar como exemplo o ambiente LOGO.

## 2.1.3 JOGOS

Em geral, os jogos tentam desafiar e motivar o aprendiz, envolvendo-o em uma competição com a máquina ou com seus colegas. A maneira mais simples de se fazer isso é, por exemplo, apresentando perguntas em um tutorial e contabilizando as respostas certas e erradas. Neste caso, pode-se dizer que as ações do aprendiz são mais semelhantes ao que acontece em um tutorial.

Uma outra utilização dessa abordagem pode ser mais semelhante com o que acontece com as simulações, onde as leis e regras do jogo já estão definidas *a priori*. Neste caso, o aluno deve jogar e, com isso, elaborar hipóteses, usando estratégias e conhecimentos já existentes ou elaborando novos conhecimentos. Os jogos podem ser bastante úteis enquanto criam condições para o aprendiz colocar em prática os conceitos e estratégias que possuem. No entanto, o aprendiz pode estar usando os conceitos de forma correta ou errônea. É necessário que o professor documente as situações apresentadas pelo aprendiz durante o jogo e torne a discutir as situações com o aprendiz, recriando-as e apresentando conflitos e desafios, com o objetivo de propiciar condições para o mesmo compreender o que está fazendo (Rocha, 1999).

### **2.1.4 EXERCÍCIO-E-PRÁTICA**

São geralmente utilizados para fazer uma revisão do material visto em aula, principalmente material que envolve memorização e repetição, como aritmética e vocabulário. A vantagem deste tipo de *software* é que o professor dispõe de uma enorme gama de exercícios que cada aluno pode resolver de acordo com seu grau de conhecimento e interesse (Rocha, 1999).

## **2.2 SOFTWARE EDUCACIONAL**

O computador pode ser um importante recurso para promover a passagem da informação para o usuário ou promover a aprendizagem. No entanto, da análise dos *softwares* é possível entender que o aprender não deve estar restrito ao *software*, mas à interação professor-aluno-*software*. Alguns *softwares* apresentam características que favorecem a atuação do professor, como no caso da programação; outros, em que certas características não estão presentes, requerem um maior envolvimento do professor para auxiliar o aluno a aprender, como no caso do tutorial (MEC, 2001).

Assim, a análise dos *softwares* educacionais em termos do aprendizado e do papel que o professor deve desempenhar para que o aprendizado ocorra, permite classificá-los em posições intermediárias entre os tutoriais e a programação. No entanto, cada um dos diferentes *softwares* usados na educação apresentam características que podem favorecer, de maneira mais explícita, o processo de construção do conhecimento. É isso que deve ser analisado quando se escolhe um *software* para ser usado em situações educacionais.

No entanto, é necessário entender que qualquer tentativa para analisar os diferentes usos do computador na educação é problemática e pode resultar em uma visão muito simplista sobre o *software* e seu uso. Porém, pode ser um exercício interessante e ajuda a entender o papel do computador e como ele pode ser efetivo no processo de construção do conhecimento (MEC, 2001).

## **2.2.1 CARACTERÍSTICAS DO SOFTWARE EDUCACIONAL**

De acordo com Valente (1995), muitas são as características necessárias a um *software* educacional, entretanto destacam-se as seguintes:

- a) gerar motivação e atrativos:
  - ser interativo;
  - mostrar linguagem clara ao usuário;
  - sem a aparência de julgamento, fazer com que o usuário identifique e corrija suas respostas;
- b) liberdade no aprendizado:
  - sempre que possível, o aprendizado deve ser conduzido pelo usuário, sem que ele se sinta controlado pelo sistema;
  - possibilidade de resoluções diversas para um mesmo problema, segundo a capacidade do usuário, não limitando sua criatividade;
  - reconhecer o valor do erro;
  - há limitações dos *softwares* em resolver todos os tipos de situações, os usuários devem também receber o incentivo de busca em outras fontes de pesquisa;
- c) programação sólida e efetiva:
  - apresentação da informação deve ser rica e variada para melhor absorção;
  - clareza na sua finalidade;
  - permitir o acesso a módulos anteriores;
  - permitir o uso por mais de um usuário em atividades iterativas.

## **2.2.2 VANTAGENS DO SOFTWARE EDUCACIONAL**

O uso do computador adapta-se bem na tarefa de ensino, apresentando alguns benefícios, conforme Mielke (1991):

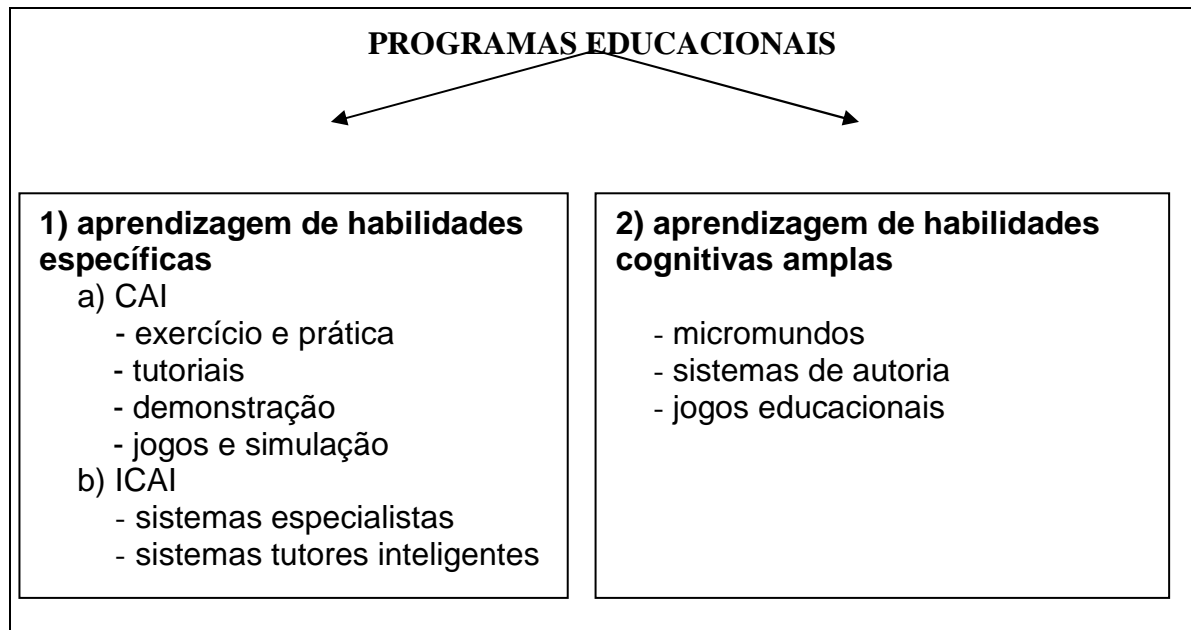
- a) sua capacidade de memorização permite não somente o registro das respostas do estudante, mas também o acesso seletivo às informações armazenadas;
- b) permite uma apresentação mais atrativa de um conteúdo, através de multimídia;
- c) permite uma comunicação interativa, recebendo respostas do aluno, tratando-as e emitindo respostas quanto ao resultado, gerando assim uma progressão pedagógica;
- d) supera obstáculos, como grandes distâncias, através do aprendizado à distância;
- e) repete incessantemente, os mesmos programas e exercícios, porque os alunos muitas vezes podem ficar com dúvidas e deixam de perguntar ao professor, pelo fato do professor já ter explicado uma ou mais vezes o mesmo assunto;
- f) permite uma evolução do conteúdo controlada pelo ritmo e interesse do aluno;
- g) suprime a rigidez atual quanto ao lugar e hora da exposição do conteúdo;
- h) possibilita que o professor se concentre mais nos aspectos criativos da formação, deixando os aspectos repetitivos para o computador.

### **2.2.3 TAXONOMIA DE SOFTWARE EDUCACIONAL**

As taxonomias mais tradicionais utilizadas para classificar os programas educacionais não contemplam as modalidades que utilizam técnicas de Inteligência Artificial (IA) e os ambientes cooperativos. Inclusive, muitas delas não levam em consideração um aspecto fundamental: o tipo de aprendizagem proporcionada pelo ambiente. A necessidade de se fazer uma reflexão neste sentido originou a taxonomia sugerida por Giraffa (1999), onde os programas educacionais são divididos em dois grupos, conforme fig. 1:

- a) programas com a aprendizagem do aluno dirigida a um conjunto de habilidades específicas;
- b) programas para aprendizagem de habilidades cognitivas amplas (AEAC).

**Figura 1 - Taxonomia para programas educacionais**



Fonte adaptado de Giraffa (1999) *apud* Cabral (1999)

No quadro 1 da fig. 1 estão os programas educacionais onde a aprendizagem proporcionada pelo ambiente está centrada na aquisição de habilidades específicas, por exemplo o manuseio do *mouse*, dividindo-se em dois grupos:

- a) *Computer Assisted Instruction* (CAI), surgidos na década de 50 e oriundos de projetos da área de Educação;
- b) *Intelligence Computer Assisted Instruction* (ICAI), surgidos na década de 70, a partir de projetos de pesquisas da área de IA.

O fato destes dois grupos serem originados de grupos de pesquisas diferenciados (Educação e IA), faz com que a identificação das modalidades dos programas que constituem cada categoria, apresentem mais ou menos consenso na comunidade de Informática na Educação. Os CAI, oriundos da Educação, apresentam propostas baseadas nas teorias aceitas pela comunidade de educadores. No entanto, o que é considerado como teoria na área de Educação, nem sempre satisfaz os requisitos de uma teoria formal na área de Ciência da Computação. A maioria das teorias considerada em Educação, não apresenta um modelo computacional necessário sob o ponto de vista da Ciência da Computação (Giraffa, 1999) *apud* (Cabral, 1999).

No grupo dos CAI na primeira categoria encontram-se as seguintes modalidades:

- a) programas de reforço ou exercício: visa a aquisição de uma habilidade ou a aplicação de um conteúdo já conhecido pelo aluno, mas não totalmente dominado. O aluno pratica e testa conhecimentos de forma dirigida e procedural. As versões mais atuais destes programas utilizam recursos hipermídia mantendo estas características;
- b) tutorial: o aluno seleciona dentre as diversas opções disponíveis, o conteúdo que deseja estudar. Mesmo aqueles que utilizam estrutura de hipermídia ou foram escritos em versão WEB, mantêm as características do ensino tradicional e não apresentam grandes revoluções, sob o ponto de vista pedagógico;
- c) jogos educacionais e simulações: são modalidades que adquiriram uma projeção maior nos últimos anos devido ao desenvolvimento dos recursos de *hardware* e *software* disponíveis no mercado. As diferenças conceituais entre os jogos e as simulações podem ser caracterizadas pelo fato de que o jogo é um processo intrinsecamente competitivo (em que coexistem a vitória e a derrota) e uma simulação é a simples execução dinâmica de um modelo previamente definido.

O grupo dos ICAI apresenta uma estrutura diferenciada dos CAI. Os programas utilizam técnicas de IA e os resultados da psicologia cognitiva com uma arquitetura modular. Baseia-se no conteúdo e independe do método de ensino utilizado (estratégias e táticas empregadas na interação com o aluno) em relação ao domínio. Os sistemas ICAI foram projetados como uma tentativa de fazer com que o programa educacional deixe de ser um mero “virador de páginas eletrônico” e se torne um elemento mais ativo no processo de ensino-aprendizagem .

Dentro dos ICAI encontra-se a modalidade Sistemas Tutores Inteligentes (STI), onde existe um modelo de aluno que objetiva personalizar o trabalho conforme as diferenças individuais de cada usuário. Os modelos de aluno (forte e fraco) determinam se o STI é um tutor ou um assistente. Os tipos de modelo do aluno pretendem reproduzir o tipo de aluno que está interagindo com o sistema (ambiente) através da utilização de diversas técnicas de modelagem. Estas técnicas consideram desde modelos estereotipados simples até modelos sofisticados envolvendo estados mentais (Cabral, 1999).

O quadro 2 da fig. 1 contempla os AEAC, que permitem a aprendizagem de habilidades cognitivas amplas. Deseja-se que os alunos obtenham um nível mais elevado de

aprendizagem e que ultrapassem as habilidades mais simples, com isso um indivíduo criativo deverá desenvolver as seguintes habilidades:

- a) consciência sobre o conhecimento, isto é, o indivíduo deverá ser sensível aos pormenores do conteúdo e suas aplicações;
- b) fluência nas suas idéias;
- c) flexibilidade para perceber alternâncias nas situações que se apresentam ao longo do trabalho;
- d) capacidade de reorganização;
- e) capacidade de desenvolver sua originalidade, ou seja, gerar idéias novas, sem imitar a realidade conhecida.

Neste grupo estão todos os programas educacionais que utilizam como referencial teórico o paradigma que enfatiza o trabalho do aluno e o “*aprender a aprender*”.

A categoria 2 do quadro dos programas educacionais está dividida da seguinte forma, segundo Cabral (1999):

- a) micromundos: surgiram na década de 60 através dos trabalhos de Papert e se caracterizaram por ser uma proposta contrária aos CAI. Neste sentido, o ambiente permite que o aluno trabalhe de forma diversificada, segundo seu próprio ritmo e ainda possibilita que ele construa sua solução utilizando recursos de programação inerentes ao ambiente. Foram criados para desenvolver habilidades cognitivas no aluno, para proporcionar o pensamento reflexivo. Trabalham basicamente com a solução de problemas e não apresentam conteúdos pré-fixados e respostas;
- b) sistemas de autoria: ferramenta de criação que possibilita ao aluno viabilizar seu projeto de trabalho. Oferece ao aluno a possibilidade de explorar um conjunto amplo de habilidades cognitivas exercendo sua criatividade;
- c) jogos educacionais: apresentam uma concepção diferenciada daqueles mostrados na categoria 1. Nestes ambientes, existe um modelo de simulação onde o tipo de ação executada pelo aluno fará diferença no resultado do jogo. Não é mais uma mera questão de ganhar ou perder. O aluno atinge ou não um determinado grau de controle do jogo através do resultado que aparece na tela no final do tempo da partida. São ambientes mais sofisticados envolvendo um maior grau de complexidade, tanto no seu projeto, como na sua implementação.



## 2.3 EVENTOS INSTRUCIONAIS

Segundo Campos (1994), as formas de aprender atividades instrucionais podem ser retidas em uma das três áreas que são:

- a) campo cognitivo: é a área onde o ser humano aprende associando as habilidades intelectuais, tal como assimilação de informações ou conhecimento, sendo comum em treinamento e educação. O cognitivo aprende objetivos e especifica a aquisição do conhecimento particular da informação;
- b) campo psicomotor: é a área onde o ser humano aprende associando o movimento físico e habilidades. Envolve a demonstração de algumas habilidades físicas ou o desempenho de uma tarefa;
- c) campo afetivo: é a área onde o ser humano aprende associando atitudes, sentimentos, interesses, opiniões, pareceres mundiais e valores.

O processo de aquisição da instrução utiliza uma seqüência de eventos, que acompanham o processo interno de aprendizado do ser humano. Se, em uma lição todos os eventos forem seguidos, os objetivos da lição serão certamente atingidos.

Um dos pesquisadores que proliferou a área do treinamento foi Gagné. Seu interesse estava em como fazer as pessoas aprenderem. Gagné admite uma variedade de tipos de aprendizagem e a sua relação com a instrução é bem elaborada, pois para ele, a “aprendizagem é um processo que permite a organismos vivos modificarem seus comportamentos de maneira rápida e mais ou menos permanente, de forma que a mesma modificação não tenha que se repetir a cada nova situação. A prova de que a aprendizagem ocorreu consiste na verificação de uma mudança comportamental relativamente persistente. Infere-se nessa observação que o organismo procedeu a uma mudança interna e portanto, que aprendeu” (Gagné, 1974).

Segundo Campos (1994), cada tipo de aprendizagem requer diferentes eventos da instrução. Com isso, Gagné descreveu os eventos instrucionais relacionados com as modalidades de *softwares* educacionais, conforme apresentado no Quadro 1:

**Quadro 1 - Eventos instrucionais conforme as modalidades de *softwares* educacionais**

Eventos Instrucionais	CAI			
	Exercício	Tutorial	Simulação	Jogo
Despertar a atenção	*	*		*
Apresentar os objetivos		*	*	
Verificar os pré-requisitos		*		
Apresentar estímulos materiais		*	*	*
Orientar a aprendizagem		*		
Solicitar o desempenho	*	*	*	*
Fornecer <i>feedback</i>	*	*	*	*
Analisar o desempenho		*		
Aperfeiçoar a atenção e a transferência		*		

Fonte: Campos (1994)

### 2.3.1 OS NOVES EVENTOS INSTRUCIONAIS

Há nove passos de eventos instrucionais, um para cada processo interno de aprendizado. Para que um comportamento desejado ocorra, devem existir condições adequadas para o ensino. Os três primeiros passos dos eventos instrucionais são focalizados na preparação do aprendiz, chamando atenção no processo de aprender e os demais se preocupam como a aprendizagem será realizada e como os alunos absorverão os conteúdos. Cada evento será descrito a seguir (Reynolds, 1996):

- a) despertar a atenção: o primeiro passo é focalizar a atenção do aprendiz no trabalho que será aprendido. Gagné sugere textos que chamem a atenção, ou tons de áudio e misturas de cores na tela do computador. O uso de vídeo e gráfico permite que métodos mais sofisticados ganhem a atenção do aprendiz. Utilizar os recursos disponíveis pelo computador e uni-los ao material de aprendizado para que o ensino torne-se mais instrutivo;
- b) apresentar os objetivos: deve existir uma tela, no início de cada lição informando os objetivos da lição que será estudada e descrevendo o que será encontrado. O desempenho do aprendiz deve estar na conclusão. As ajudas devem focalizar a atenção do aprendiz, enquanto inicia o processo interno de aprendizagem, por exemplo, se o aluno não respondeu um exercício em um determinado tempo, e

não acionou nenhum outro recurso do *software*, a resposta iria aparecer na tela automaticamente;

- c) verificar os pré-requisitos: é vital que o aprendiz avalie o novo material no contexto do conhecimento prévio em ordem de como a nova informação é apresentada. O conhecimento anterior deve ser recobrado na memória de longo prazo quando se trabalha a memória. Um meio simples de o aprendiz lembrar conceitos aprendidos previamente, envolve formular perguntas que envolvam os assuntos já vistos em situações anteriores;
- d) apresentar estímulos materiais: mostra-se ao aprendiz como as informações estão distribuídas, focaliza a atenção do aprendiz em pontos importantes e demonstra exemplos e não-exemplos. É dado ao aprendiz mensagens de lições passadas, numa adaptação de método e descoberta. Na matemática, um bom recurso seria a utilização de gráficos nas lições para ajudar o aprendiz visualizar os resultados físicos do procedimento descrito em uma função por exemplo;
- e) orientar a aprendizagem: fornecer um meio de codificar uma nova informação é maneira positiva para os instrutores utilizarem, mostrar dicas e pistas do que está acontecendo. Nas partes mais difíceis de ensinar uma lição, mostrar o que está ocorrendo é uma forma do aluno reter melhor as informações recebidas. Para cada informação deve-se encontrar uma melhor maneira do aluno saber o que está aprendendo e induzi-lo a que caminho percorrer;
- f) solicitar o desempenho: o processo interno de responder (pensar sobre um estímulo e formular uma resposta) é uma parte crítica no processo de aprendizado. O resultado dos testes nem sempre corresponde ao aprendido, na formulação dos testes dever-se-ia buscar a recuperação das informações recebidas, fazendo com que os alunos pratiquem essas informações;
- g) fornecer *feedback*: extrair o desempenho do aprendiz é só uma parte do que deve ser realizado, o essencial é proporcionar ao aprendiz o *feedback*, fazer com que o aprendiz esteja ciente da qualidade de sua performance. Quando se aprende tarefas procedurais, o *feedback* deve ser imediato para aprender com máxima eficiência. Além da coordenação, o tipo de *feedback* dado é um fator crítico em ensinamentos de procedimentos. A ajuda pode ser oferecida por exibir um ícone para o

aprendiz pedir auxílio, ou pode ser exibido automaticamente depois de um tempo pré-determinado sem nenhuma resposta;

- h) avaliar o desempenho: durante o treinamento o aprendiz alimenta confiante que as habilidades foram aprendidas, onde deverá ser capaz de demonstrar o procedimento solicitado e se conseguiu assimilar as informações. Toda ajuda e *feedback* é removida e o aprendiz deve executar o procedimento, o computador facilmente acompanha as ações do aprendiz. Os resultados cercam as áreas específicas de fraqueza do aprendiz;
- i) aperfeiçoar a atenção e a transferência: a simulação é usada para induzir o aprendiz. Para que isso proceda, o aprendiz deve lembrar de trabalhar todas as memórias (de curto e longo prazo) para as tarefas aprendidas, e deve ser capaz de aplicá-las em várias situações. O aprendiz contará com as mensagens de ajuda, o processo torna-se sistemático no domínio de uma sub-tarefa, a partir da construção gradual de sub-tarefas chega à construção de uma tarefa maior. Esse método aumentará a retenção, e deve melhorar a transferência de aprender.

## 2.4 MULTIMÍDIA

Conforme Bizzotto (2000), o termo multimídia tem sido utilizado para designar *softwares* que utilizam em conjunto, diversas mídias como som, textos, imagem e vídeo, sendo que a característica distintiva de um *software* multimídia é a interatividade. Os recursos multimídia estão cada vez mais sendo utilizados para oferecer às pessoas aprendizado e acesso a novas informações que antes eram só vistas em forma de livros, revistas, jornais, televisão e rádio. Rosenborg (1993) afirma que o termo multimídia se refere a diversos tipos de informações que podem ser editadas e armazenadas em tipos específicos de dispositivos. Uma mensagem multimídia é uma unidade de comunicação que, quando transmitida, produz uma combinação das várias mídias citadas anteriormente.

Valente (1999) afirma que o *software* multimídia vem ficando cada vez mais interessante, criativo e está explorando um número maior de possibilidades. É possível ao aluno navegar em um espectro amplo de tópicos, como também penetrar a fundo neles. Assim, o uso de sistemas multimídia já prontos é uma atividade que auxilia o aprendiz a adquirir a informação, porém não processá-la. Não existe garantia que essa informação irá se transformar em conhecimento. Cabe ao professor suprir essas situações para que isso ocorra.

### 2.4.1 CARACTERÍSTICAS DO SOFTWARE MULTIMÍDIA

Segundo Winn (1997) *apud* Dalfovo (1997), um *software* multimídia apresenta determinadas características próprias, as quais podem ser classificadas em três dimensões conforme descrito a seguir:

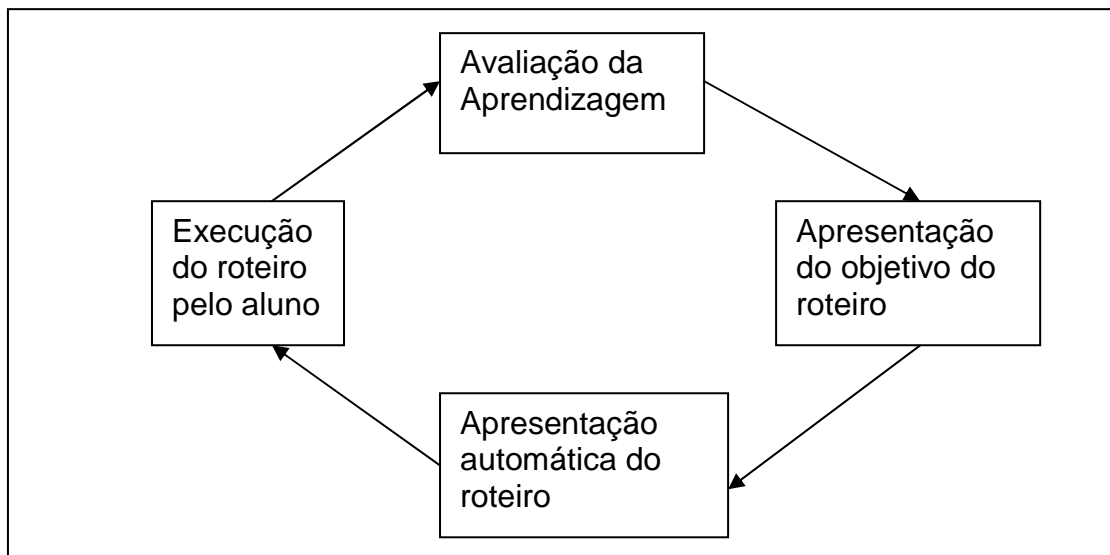
- a) autonomia: neste caso o ambiente funciona por si só, ou seja, não necessita de nenhuma ação do usuário. Exemplos característicos de baixa autonomia são os tutoriais e programas de exercício-e-prática que ficam parados no aguardo de alguma ação do usuário. Como sistemas autônomos, poderiam ser incluídos as simulações em tempo real e alguns jogos. Nestes casos, o ambiente segue o seu desenvolvimento próprio sem esperar por uma atitude do usuário;
- b) presença: esta característica está relacionada em fazer com que o usuário se sinta como se estivesse realmente no lugar representado. Com ambientes baseados em computador, o senso de presença aumenta à medida que a interface se torna mais intuitiva, mais transparente, tornando-se o mais próximo do mundo real;
- c) interação: está intimamente relacionada com a forma como o usuário interage com o ambiente, ou seja, o aprendizado ocorre de acordo com as observações e ações que o usuário toma em relação a determinados assuntos. A interação depende única e exclusivamente das ações e reações do usuário, ou seja, o usuário deixa de ser um receptor passivo, passando a interagir com o *software*.

Diferentes tecnologias são caracterizadas por diferentes graus de autonomia, presença e interação. Um sistema de televisão, por exemplo, possui alta autonomia, baixa interação e presença variável dependendo do tipo de programa. *Softwares* tutoriais possuem alta interação, mas baixa presença e autonomia. Jogos de computadores possuem alta interação, mas baixa presença e autonomia (Winn, 1997).

### 3 DESENVOLVIMENTO DO TUTORIAL

No desenvolvimento do *software* tutorial para o aprendizado no ambiente LOGO, utilizou-se como referência alunos de ensino fundamental. Foi levada em conta a grande diferença dos perfis dos alunos de ensino fundamental, sendo que a mesma sala possui desde alunos com total desconhecimento do ambiente LOGO até alunos com um completo domínio deste ambiente. Assim, no desenvolvimento do tutorial buscou-se atender a estes perfis diferenciados, permitindo que o aluno possa escolher a seqüência de aprendizado que melhor se adapta ao seu caso específico. No entanto, uma seqüência padrão, é proposta para aqueles que não possuem conhecimento anterior. Em termos de estratégia de aprendizagem, o tutorial segue um ciclo de aprendizagem, conforme apresentado na fig. 2.

**Figura 2 - Ciclo de aprendizagem utilizado pelo tutorial desenvolvido**



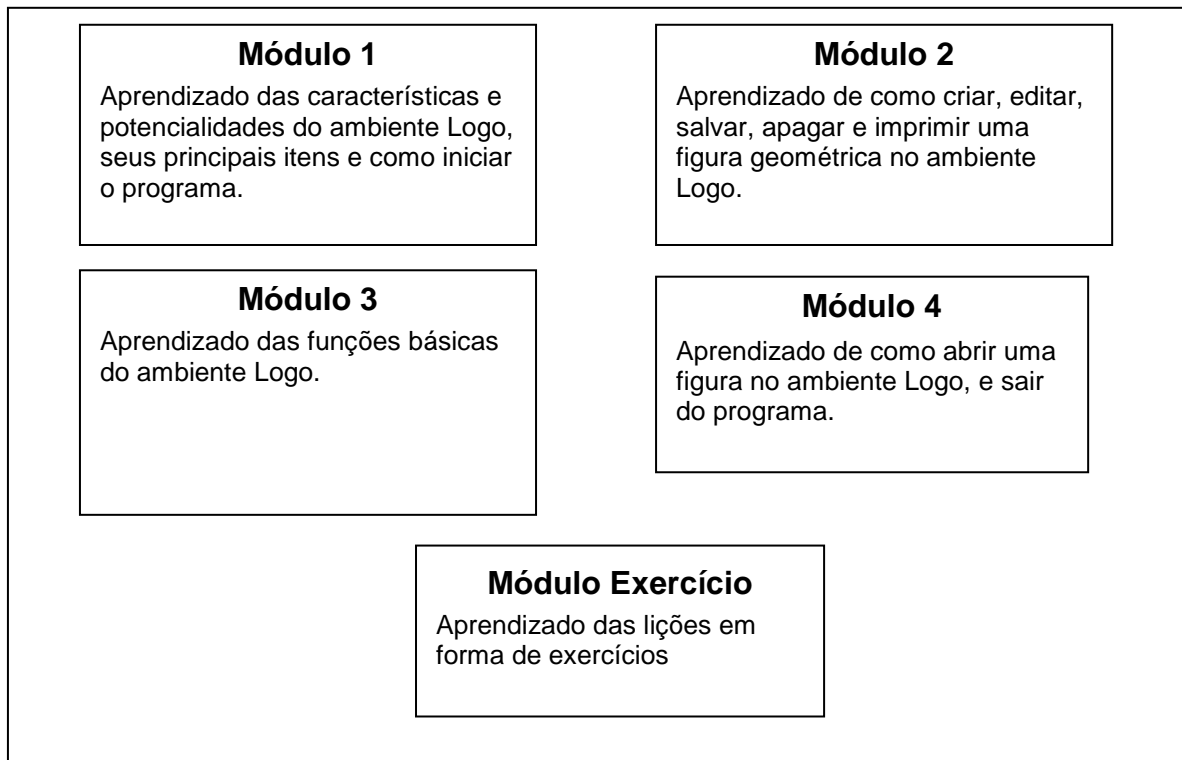
Em cada início de lição, é informado ao aluno o objetivo daquela lição, ou seja, o que o aluno deverá saber após ter realizado o roteiro proposto. Após a leitura do objetivo da lição pelo aluno, pressiona-se qualquer tecla para o início da animação. Mostra-se passo-a-passo, através de um ponteiro de *mouse* com destaque em vermelho, quais os procedimentos e ações que o aluno deve seguir para atingir o objetivo proposto inicialmente.

Após a apresentação automática do roteiro, o aluno deve realizar e repetir o roteiro apresentado, através do exercício, dessa vez sem a ajuda do tutorial. Nesta parte interativa, o aluno é informado constantemente se está seguindo os passos na seqüência correta. Após

conclusão do exercício, o aluno é avisado do mesmo, e após poderá seguir o aprendizado, passando para o módulo seguinte, reiniciando o ciclo.

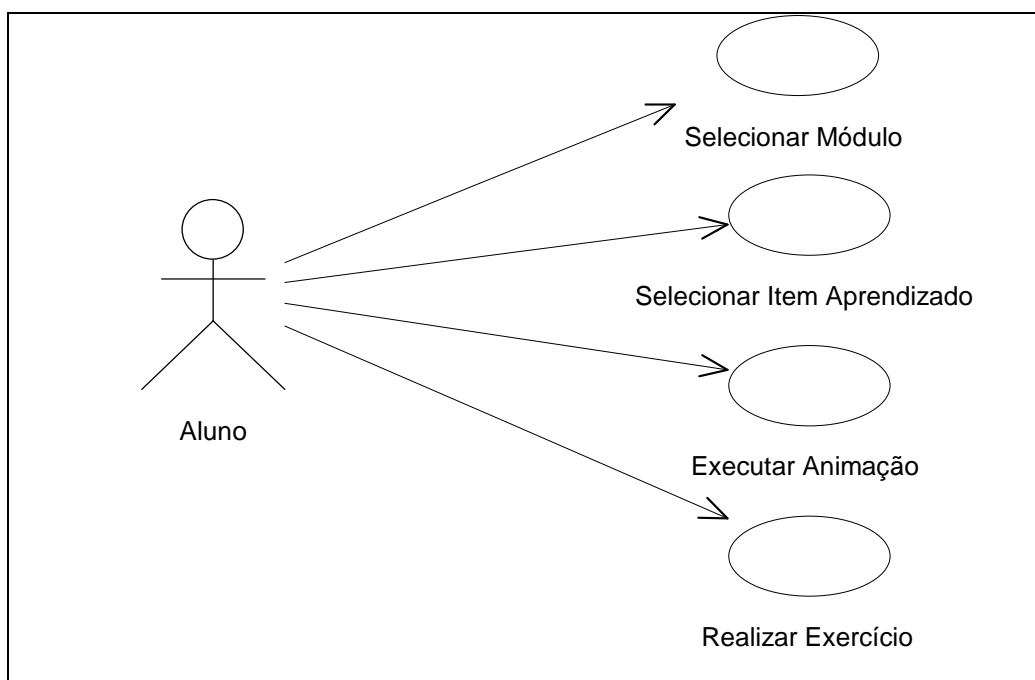
Em termos de conteúdo, o tutorial desenvolvido foi dividido em 5 módulos, conforme apresentado na fig. 3.

**Figura 3 - Módulos do tutorial**



### 3.1 ESPECIFICAÇÃO

Nesta seção serão apresentados os diagramas da metodologia OOHDM utilizada na especificação do tutorial, sendo que em OOHDM, uma aplicação é construída em um processo composto por quatro passos, quais sejam: projeto conceitual, projeto navegacional, projeto da interface abstrata e implementação. Além disso, são apresentados os principais casos de uso do *software* oriundos da UML a seguir na fig. 4.

**Figura 4 - Diagrama de Caso de Uso**

Através da fig. 4 é identificado o ator e os casos de uso que fazem parte da implementação e suas funcionalidades estão descritas a seguir:

- a) selecionar módulo: o aluno seleciona dentre os quatro módulos disponíveis para o aprendizado o que melhor atende ao seu perfil;
- b) selecionar item aprendizado: após escolha dentre uma das quatro opções de aprendizado disponíveis, o aluno escolhe um item de aprendizado dentre os vários disponíveis;
- c) executar animação: após escolha do item de aprendizado que deseja estudar, o tutorial executa a animação;
- d) realizar exercício: o aluno, após acompanhar a animação, realiza o exercício proposto pelo tutorial para fixar o conteúdo exposto.

### 3.1.1 PROJETO CONCEITUAL

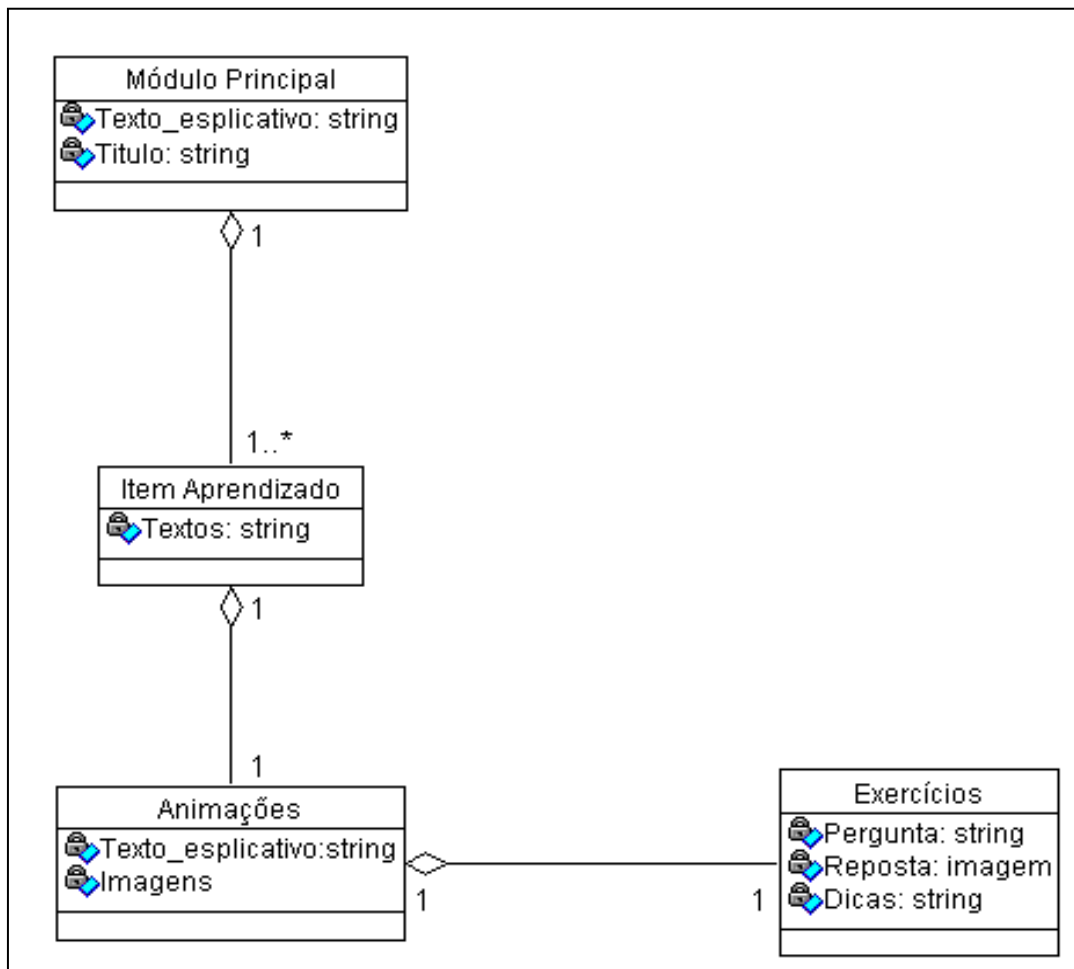
Nesta etapa, realizou-se a construção do esquema conceitual que representa os objetos e relacionamentos existentes no domínio da aplicação, sendo que foram feitas as seguintes atividades: definição das classes e relacionamentos de acordo com a semântica do domínio (Schwabe, 2000).



Classes são o elemento mais importante de qualquer sistema orientado a objetos, sendo que uma classe é uma descrição de um conjunto de objetos com os mesmos atributos, relacionamentos, operações e semântica. Toda classe deve ter um nome que a distingua das outras classes. Conforme fig. 5, tem-se quatro classes cujos nomes são: Módulo Principal, Item Aprendizado, Animações e Exercícios.

Toda classe contém atributos. Um atributo representa uma propriedade do que está sendo modelado, que é compartilhada por todos os objetos da classe, sendo que descrevem os dados contidos nas instâncias de uma classe. Dentro da classe de nome Módulo Principal da fig. 5, tem-se os atributos Texto\_ explicativo e Título onde cada atributo tem seu valor assumido na classe, por exemplo, o atributo Texto\_ explicativo tem como valor textos referentes ao tutorial. Enquanto o atributo Título na classe Item Aprendizado tem-se o atributo Textos que contém texto do aprendizado. Na classe Animações tem-se o atributo Texto\_ explicativo que tem como valor os textos e o atributo imagens onde estão as imagens do tutorial. Na classe Exercícios tem-se o atributo Pergunta que tem como valor textos com perguntas referentes aos exercício, o atributo Resposta que tem como valor textos para representar respostas. Tem-se também na classe Exercícios o atributo Dicas que tem como valor textos para representar as dicas do tutorial.

**Figura 5 - Diagrama Projeto Conceitual**



### 3.1.2 PROJETO NAVEGACIONAL

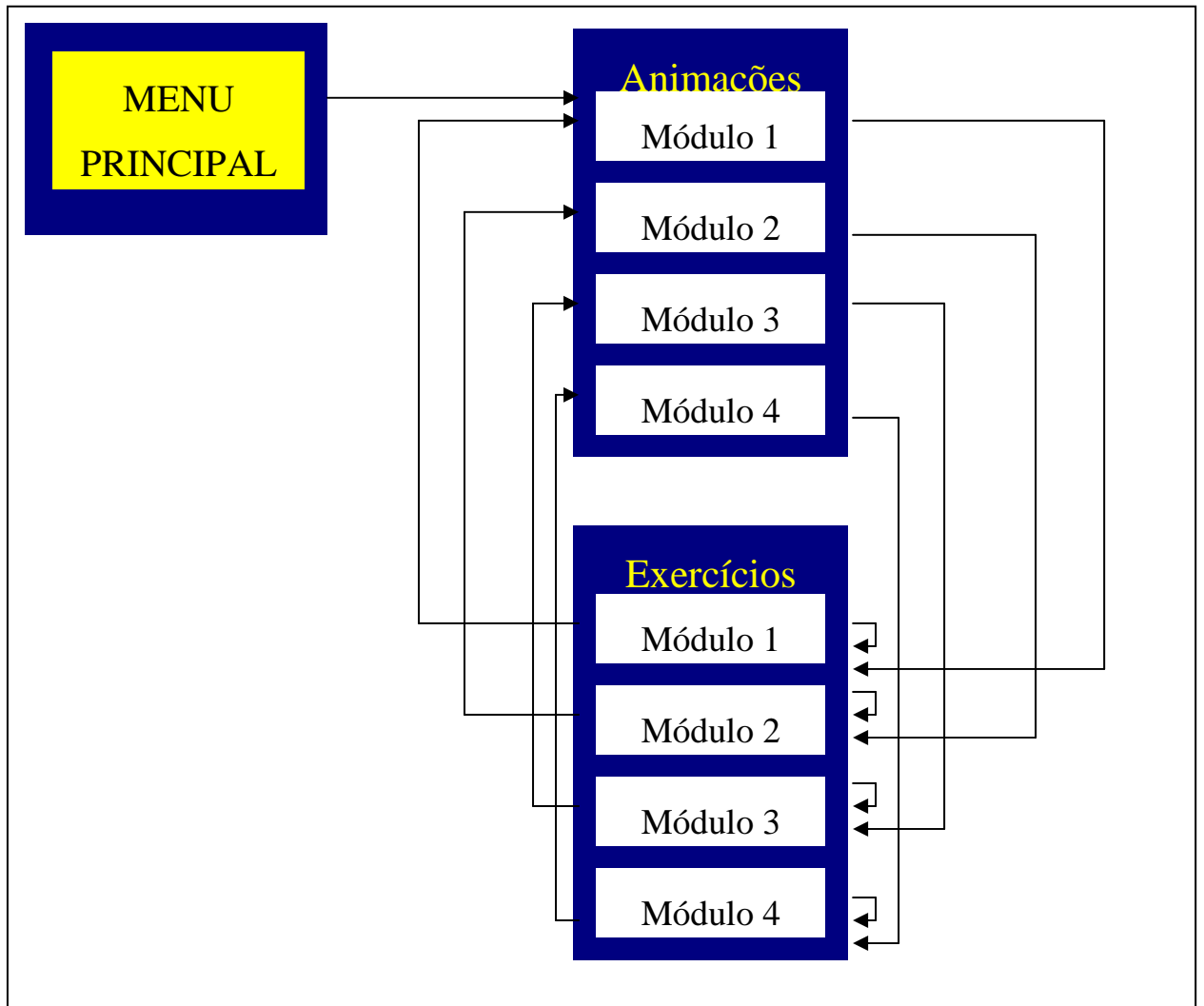
No projeto navegacional, leva-se em conta que a aplicação é vista como uma visão navegacional sobre o domínio conceitual. Isso mostra que uma das principais características desta metodologia que a difere das outras é a noção de navegação (Rossi, 2002).

Na fig. 6 tem-se o diagrama do modelo navegacional, iniciando através do item Menu Principal, onde é possível selecionar um dos módulos. Para cada módulo tem-se animações e exercícios.

No item animações pode-se escolher o módulo, onde o aluno poderá ter acesso aos exercícios referentes aquele módulo, após terminada a animação.

No item exercícios o aluno poderá optar em prosseguir com o exercício, como poderá retornar ao mesmo exercício, ou até mesmo voltar para o item animações.

Figura 6 – Modelo Navegacional



### 3.1.3. PROJETO DE INTERFACE

Não foi realizado o projeto da interface abstrata devido ao número excessivo de textos e botões presentes em cada módulo.

Assim, será mostrada a funcionalidade de cada uma das operações a seguir:

**Quadro 2 - Funcionalidades das operações do projeto de interface**

<b>Método</b>	<b>Classe</b>	<b>Descrição</b>
Rollover cursor	Módulo Principal, Item Aprendizado	Faz com que o cursor do mouse altere ao entrar na área definida por um sprite.
Play the frame	Todas	Este behavior desloca o playhead para um frame específico, através da programação com lingo.
Display text	Item Aprendizado	Este behavior aguarda ser ativado por outro behavior, por exemplo, o tooltip ou por comandos do lingo para exibir um texto dinamicamente.
Tooltip	Item Aprendizado	Este behavior responde ao evento mouseenter de um sprite, fazendo com que seja exibido um texto em um field após um tempo especificado, esse sprite no formato field, deve apresentar o behavior display text.
Push button	Item Aprendizado	Este behavior é usado para criar um botão padrão.
Prepare_exibindo	Item Aprendizado	Este behavior prepara o frame para ser exibido.
Prepare_movendo	Item Aprendizado	Este behavior prepara o frame para ser movido.
Prepare_mudando	Item Aprendizado	Este behavior prepara o frame para ser mudado.
Prepare_iniciando	Item Aprendizado	Este behavior prepara o frame para ser iniciado.
Play sound	Animações	Este behavior executa um Member de áudio em um canal de áudio específico, a partir de um evento do mouse sobre o Sprite, se o behavior for aplicado a um Sprite.
Clique_proximo frame	Exercícios	Este behavior cria um botão para pular para o próximo frame.
Clique_direito frame	Exercícios	Este behavior cria um botão para clicar no botão direito do frame.

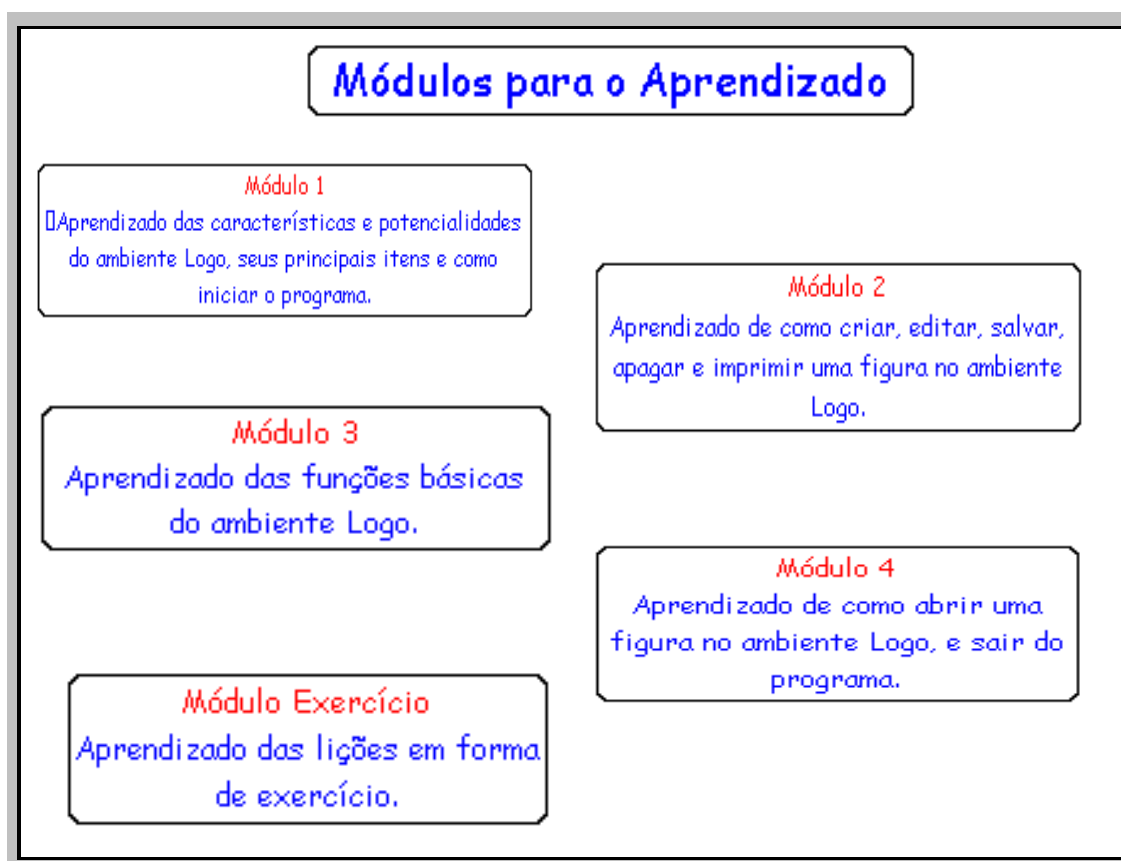
Duplo_clique frame	Exercícios	Este behavior cria um botão para dar um duplo clique no frame.
--------------------	------------	--

## 3.2. IMPLEMENTAÇÃO

Nessa seção, será apresentada a operacionalidade da implementação, bem como sua funcionalidade e alguns códigos implementados. Foi utilizado na implementação deste tutorial o *software* Macromedia Director 8.5 (Bizzoto, 2000).

Ao ser iniciado, o *software* apresenta ao usuário uma breve animação de introdução ao tutorial. Após a introdução, surge na tela do usuário a relação dos módulos de aprendizagem no ambiente LOGO, onde em cada tópico há uma breve descrição do que será apresentado nos quatro módulos de aprendizagem, conforme fig. 7.

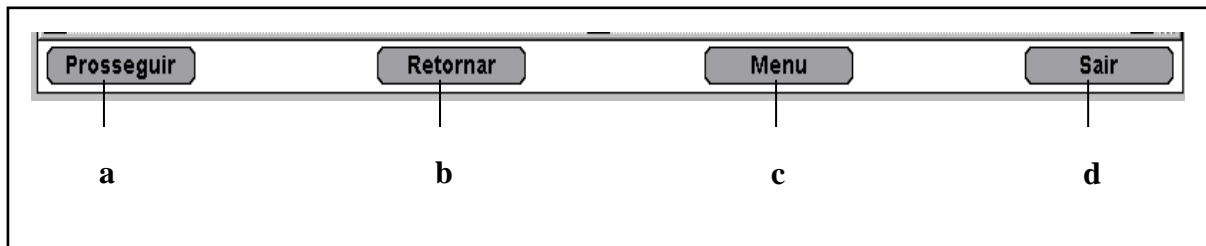
**Figura 7 – Apresentação dos módulos de aprendizagem**



O tutorial, em todos os módulos, oferece uma barra de navegação conforme fig. 8, localizado na parte inferior da tela onde o aluno terá as seguintes opções:

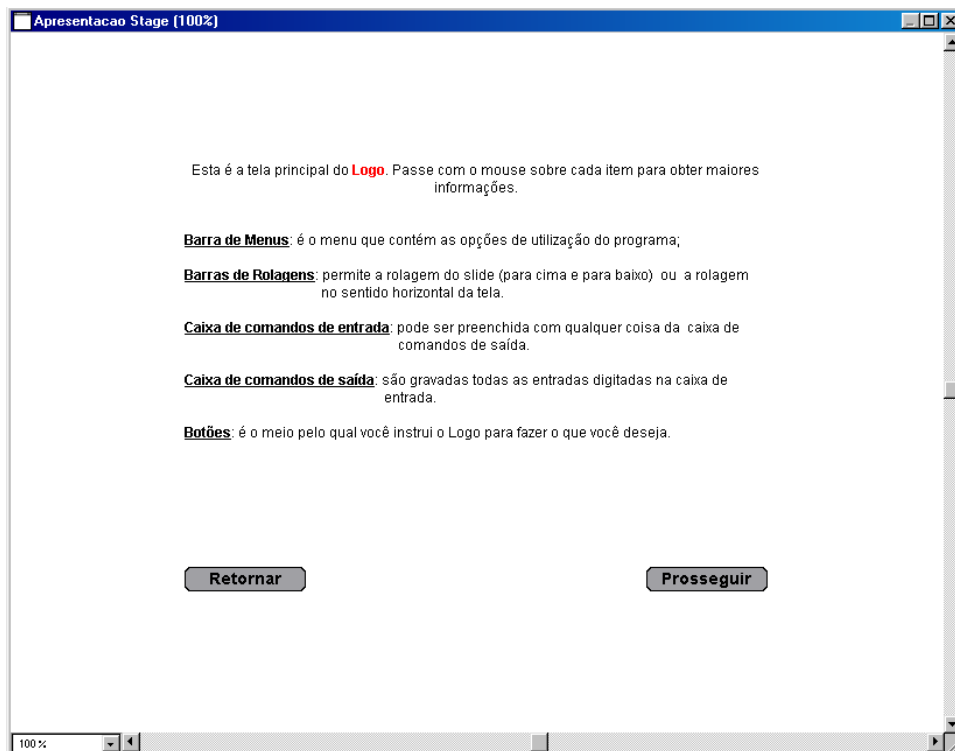
- a) ir para próxima animação;
- b) retornar a animação caso o aluno necessite recomeçar para firmar seu processo de aprendizagem;
- c) voltar para o menu principal;
- d) sair do tutorial.

**Figura 8 - Barra de navegação do tutorial**



No início de cada animação, conforme fig. 9, surge na tela uma descrição na qual o tutorial informa ao aluno o objetivo da lição que será estudada, sendo assim o aluno verifica se possui conhecimento sobre o assunto ou não. O tutorial aguarda uma ação do usuário para iniciar a animação, que pode ser um clique no *mouse* ou pressionar uma tecla.

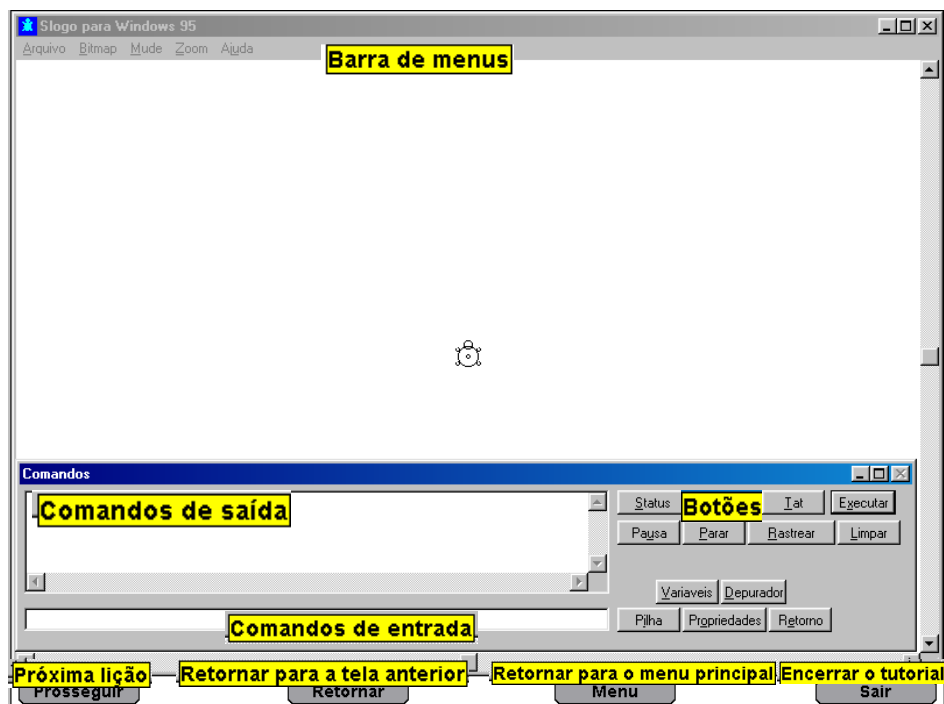
**Figura 9 – Animação dos itens principais, parte I.**



Após o aluno pressionar uma tecla ou dar um clique no *mouse*, o tutorial começa a fazer a animação referente ao item de aprendizado escolhido. O tutorial mostra a escolha do item Prosseguir, dando início ao aprendizado dos itens principais do LOGO.

Ao passar o *mouse* sobre a tela inicial do ambiente LOGO, o tutorial mostrará uma barra com informações de cada item, conforme mostrado na fig. 10.

**Figura 10 – Animação dos itens principais, parte II.**

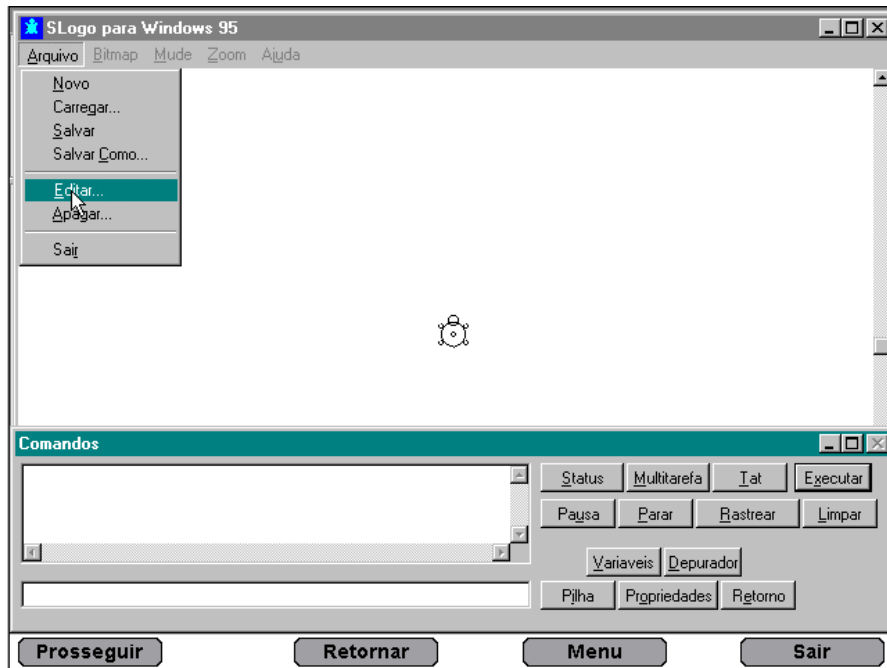


A seguir, para um maior entendimento de como funcionam as animações para cada lição, serão apresentados os passos a serem seguidos pelo aluno para o aprendizado da lição *Editar uma figura*.

Através do ponteiro do *mouse*, o usuário observa seus movimentos e ações para ir realizando seu aprendizado. A fig. 10 mostra a tela inicial do ambiente LOGO, dando início ao aprendizado da lição *Editar uma figura*.

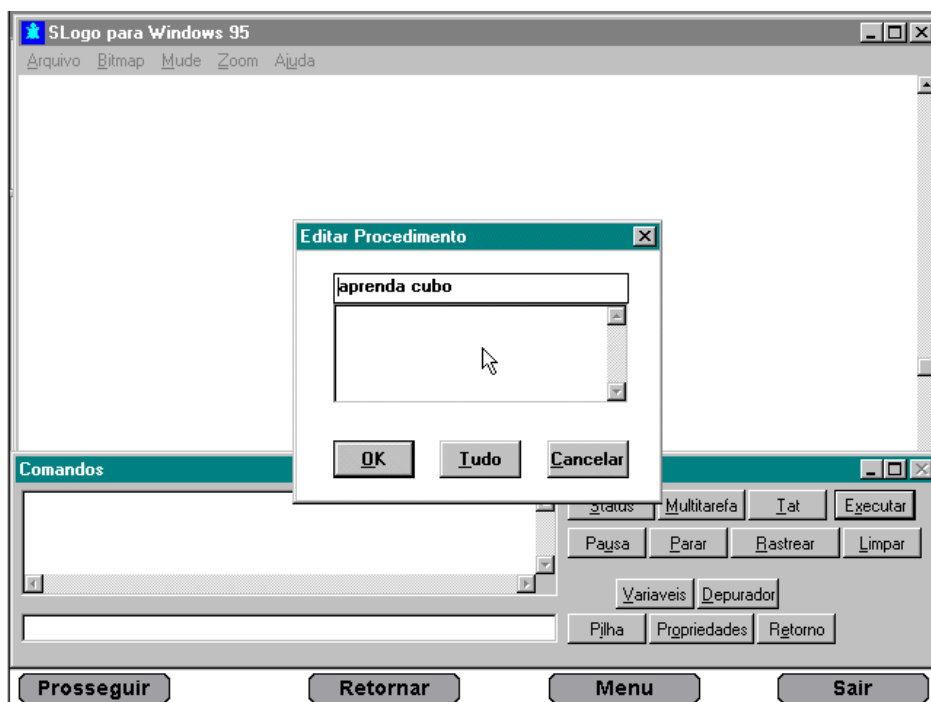
Na fig. 11, o tutorial mostra a escolha do menu Arquivo, sendo que em seguida surgirá o submenu para que o tutorial vá até a opção editar e dê um clique.

**Figura 11 – Animação para Editar Figura, parte I.**



A seguir, na fig. 12, tem-se a tela de escolha para editar procedimento. Observa-se a inserção do comando pelo tutorial, sendo que será realizada a digitação no item Editar Procedimento.

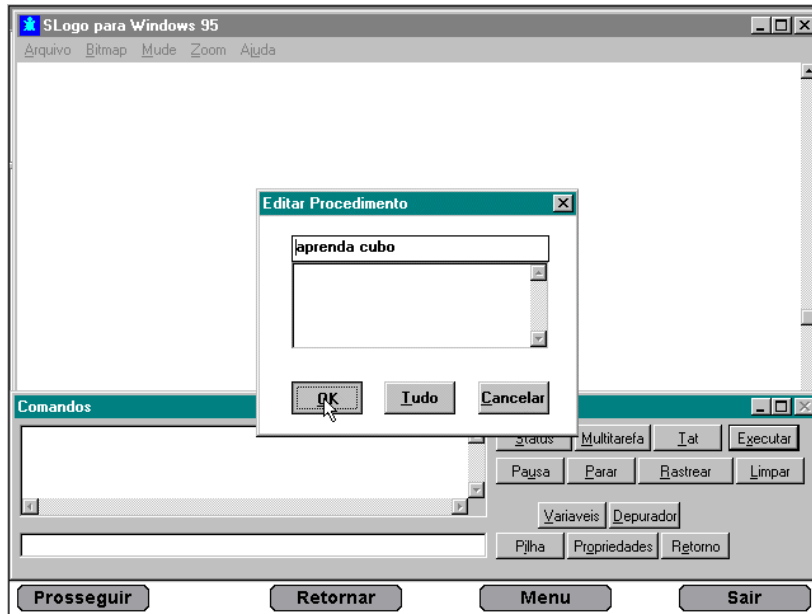
**Figura 12 – Animação para editar figura, parte II.**





Em seguida na fig. 13, tem-se a confirmação da digitação através do botão OK.

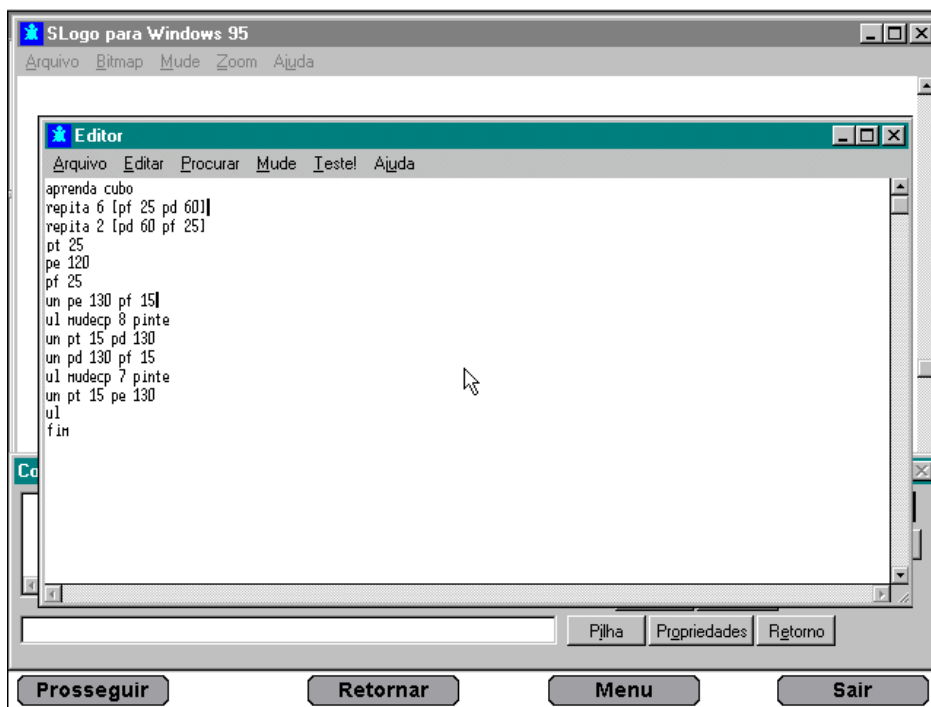
**Figura 13 – Animação para editar figura, parte III**



Após a confirmação, o tutorial mostrará os comandos na tela do editor, conforme fig.

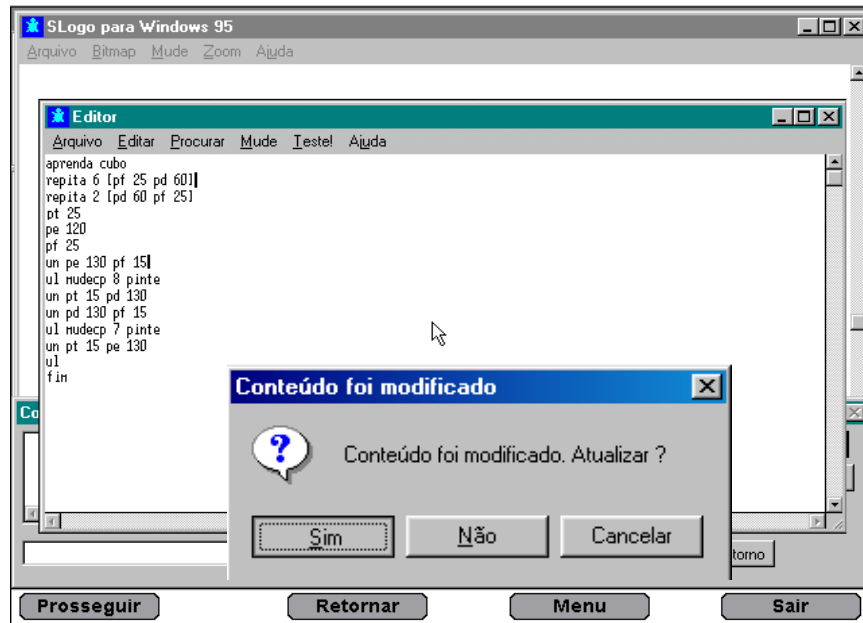
14.

**Figura 14 - Animação para editar figura, parte IV**



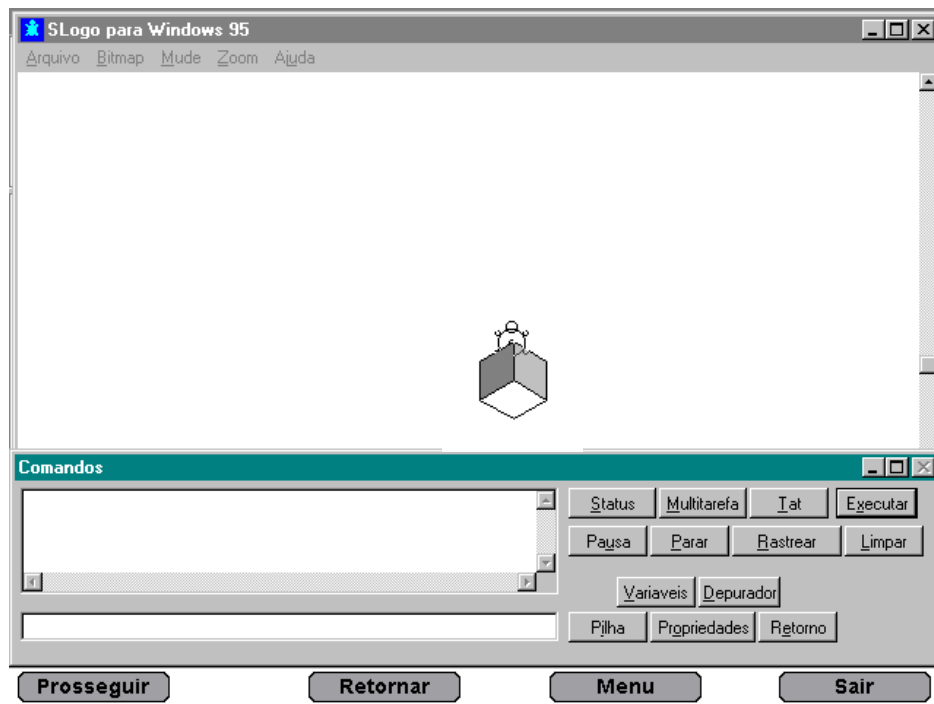
Ao sair do editor, aparece a pergunta para o tutorial se quer atualizar o editor. O tutorial clicará no botão Sim, como mostrado na fig. 15.

**Figura 15 – Animação para editar figura, parte V**



Após atualizado, o tutorial mostrará na barra comando de entrada o que foi ensinado ao LOGO e aparecerá na tela um cubo, conforme fig. 16.

**Figura 16 – Animação para editar figura, parte VI.**



Ao final da apresentação, o aluno poderá repetir os mesmos passos que lhe foram ensinados na animação, no módulo de exercícios.

No Quadro 3, mostra-se parte do código das propriedades Tipo, Seção e Arquivo, utilizados pelo programa implementado em Lingo.

### Quadro 3 – Parte do código referente às propriedades Tipo, Seção e Arquivo

```

property Tipo, Secao, Exemplo, Arquivo
on getBehaviorDescription Me
return "Verifica qual o botão selecionado e executa a ação
correspondente."
end getBehaviorDescription

on getBehaviorTooltip me
return "↵
Usado em qualquer tipo de membro."&RETURN&RETURN&"↵
Este behavior verifica qual o botão clicado e executa a ação
correspondente ao mesmo."
end getBehaviorTooltip

on getPropertyDescriptionList Me

set Lista = ["Nenhum"]
add Lista, "Tela Principal"
add Lista, "Sair"
repeat with I = 1 to (the labelList.line.count - 1)
add Lista, the labelList.line[I]
end repeat
-- Inicializa a lista de propriedades
set Descricao = [:]
-- Insere na lista as propriedades
addProp Descricao, #Tipo, [#default:"Mover", #format:#String,
#comment:"Ação:", #range:["Sair", "Mover", "Menu", "Exemplo"]]
addProp Descricao, #Secao, [#default:"Nenhum", #format:#String,
#comment:"Próxima tela:", #range:Lista]
addProp Descricao, #Arquivo, [#default:"Insira aqui o nome do
arquivo a ser aberto como uma nova janela", #format:#String,
#comment:"Arquivo:"]

return Descricao
end getPropertyDescriptionList

on mouseUp me
cursor 4
-- Verifica qual o tipo de ação deve ser tomada quando o botão é
pressionado
case Tipo of
"Sair": Quit
"Mover": Navegacao
"Menu": Fecha_Janela
"Exemplo": Abre_Janela
end case

end mouseUp

on Navegacao Me
-- Se for um marcador válido, move o header para a posição indicada
-- por Secao
if Secao <> VOID and Secao <> "Nenhum" then
play frame Secao

```

Conforme mostrado no Quadro 3, a propriedade Tipo, indica qual a ação a ser realizada, como por exemplo sair do programa, mover para um frame específico, voltar para a tela de menu, abrir tela de exemplos. Já a propriedade Seção, quando setada indica qual o próximo frame a ser executado. A propriedade Arquivo quando setada indica qual arquivo a ser aberto como uma janela no programa.

No Quadro 4, mostra-se parte do código utilizado pelo programa referente à função para abrir janela.

**Quadro 4 – Parte do código referente a função abrir janela**

```

on Abre_Janela me
  -- Se o nome do arquivo for o valor padrão, seta o nome do
  arquivo para VAZIO
  if Arquivo = "Insira aqui o nome do arquivo a ser aberto como uma
  nova janela" then
    Arquivo = EMPTY
  end if

  -- Se o nome do arquivo foi setado como um nome válido
  if Arquivo <> VOID or Arquivo <> EMPTY then
    -- Seto o nome da janela
    set Janela to "Exemplo " & string(Marker(0))
    -- Seto o nome do arquivo a ser aberto como uma janela
    set the fileName of window Janela to Arquivo
    -- Seto o nome que aparecerá na barra de título
    -- Ex: o nome do arquivo é "modulo_01_exemplo_01.dir"
    -- a função CHARS irá retornar os dois números que identificam
    qual o exemplo sendo aberto
    set Titulo to "Exemplo " & chars(Arquivo, 18, 19)
    set the title of window Janela to Titulo
    -- Força a janela a aparecer na frente de qualquer outra janela
    aberta
    moveToFront window Janela
    -- Abre a janela
    open window Janela
  end if
end Abre_Janela

```

A parte de código utilizada no programa referente a função fechar janela poderá ser visualizada no Quadro 5.

**Quadro 5 – Parte do código referente a função fechar janela**

```
on Fecha_Janela Me
    -- Verifica se a posição não aponta para a tela principal
    if Secao <> "Tela Principal" then
        -- Seta a quantidade de janelas abertas
        set Temp = count(the windowList)
        -- Seta o nome da última janela aberta
        set Auxiliar to string(the windowList[Temp])
        -- Ajusta o nome da última janela (a função the windowList retorna o
nome no
        -- formato "window <NomeDaJanela>"
        repeat with I = 1 to length(string(the windowList[Temp]))
            if char I of Auxiliar = QUOTE then
                I = I + 1
                Tamanho = length(string(the windowList[Temp]))
                Janela = chars(Auxiliar, I, Tamanho-2)
                exit repeat
            end if
        end repeat
        -- Fecha a janela
        close window Janela
    else
        beep
    end if
end Fecha_Janela
```

## 4 CONCLUSÃO

O presente trabalho teve como objetivo o desenvolvimento de um tutorial hipermídia para permitir o aprendizado no ambiente LOGO, sendo que através dos objetivos iniciais estabelecidos para o presente trabalho, conclui-se que é possível atingir:

- a) interatividade no tutorial com o Lingo, através da inclusão dos exercícios após cada animação;
- b) um maior grau de interação entre aluno e *software*, permitindo que o aluno aprenda segundo seu próprio ritmo;
- c) aplicação da metodologia OOHDM no desenvolvimento do tutorial hipermídia, a qual se mostra adequada para desenvolvimento desse tipo de aplicação através das etapas que fazem parte do conteúdo da metodologia, sendo que o *software* de autoria Macromedia Director permitiu a realização do projeto de forma simples, pois possui grande variedade de recursos.

Observa-se que a atenção aos eventos instrucionais propostos por Gagné, direciona positivamente o processo de desenvolvimento. Isto ocorre porque as fases propostas por Gagné passam a fazer parte dos objetivos de cada lição. Desta forma há uma maior preocupação com a efetividade do tutorial.

### 4.1 SUGESTÕES

Como sugestões para trabalhos futuros pode-se citar a inclusão de voz e textos mais explicativos para cada lição. Outra sugestão é o desenvolvimento de um agente inteligente que acompanhe a evolução da aprendizagem do aluno através do tutorial, assim como adaptar o tutorial na internet.

## REFERÊNCIAS BIBLIOGRÁFICAS

BARANAUSKAS, Maria Cecília C.; ROCHA, Heloísa Vieira; MARTINS, Maria Cecília et al. Uma taxonomia para ambientes de aprendizado baseados no computador. In: livro. **O computador na sociedade do conhecimento**. Campinas: NIED, 1999. p. 49-87.

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

BRIGNOLI, Juliano Tonizetti. **Protótipo de um tutorial para ensino de figuras geométricas planas utilizando recursos multimídia**, 1995. 85f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

CABRAL, Anderson R. Yanzer. **Estudo sobre treinamento baseado em computador**. 48 f. Trabalho individual I. Programa de Pós-Graduação em Ciência da Computação. Mestrado. PUC-RS, Porto Alegre, 1999. Disponível em: <<http://www.inf.pucrs.br/~anderson>>. Acesso em: 17 maio 2002.

CAMPOS, Gilda Helena Bernardino de. **Metodologia para avaliação de qualidade de software educacional: Diretrizes para desenvolvedores e usuários**. 1994. 232 f Tese de Doutorado (Ciências em Engenharia de Produção) - Coordenação de Programas de Pós-Graduação em Engenharia, Universidade Federal do Rio de Janeiro, Rio de Janeiro.

DALFOVO, Regiani. **Protótipo de software para o ensino de introdução a microinformática**. 1997. 60 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

GAGNÉ, Robert M. **Como se realiza a aprendizagem**. Rio de Janeiro: Livros Técnicos e Científicos, 1974.

GIRAFFA, Lucia M.M. **Uma arquitetura de tutor utilizando estados mentais**. 1999. Tese de Doutorado, Universidade Federal do Rio Grande do Sul, Porto Alegre.

MEC. Ministério da educação. Desenvolvido pelo Departamento de Informática de educação a distância, 2001. Apresenta textos sobre o Programa nacional de informática na educação – Proinfo. Disponível em: <<http://www.proinfo.gov.br>>. Acesso em: 30/10/2002.

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, Universidade Federal de Santa Catarina, Florianópolis.

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

ROCHA, Heloísa Vieira. **O computador na sociedade do conhecimento.** Campinas: NIED, 1999.

ROSENBORG, Victoria. **Guia de multimídia.** Rio de Janeiro: Berkeley Brasil Editora, 1993.

ROSSI, Gustavo. **Projeto de aplicações hipermídia.** [2002?]. Disponível em <<http://sol.info.unlp.edu.ar/~fer/oohdm/pg001.htm>>. Acesso em: 09 ago 2002.

SCHWABE, Daniel; ROSSI, Gustavo. **The object-oriented hypermedia design model-OOHDM,** Rio de Janeiro, [2000]. Disponível em: <[http://www.telemidia.puc-rio.br/oohdm/site\\_oohdm/oohdm.html](http://www.telemidia.puc-rio.br/oohdm/site_oohdm/oohdm.html)>. Acesso em: 27/10/2002.

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

VALENTE, José Armando. **Questão do software:** Parâmetros para o desenvolvimento do *software* educativo. Campinas: Núcleo de Informática Aplicada à Educação - NIED, 1995.

VALENTE, José Armando. **Análise dos diferentes tipos de software utilizados na educação.** Campinas: Núcleo de Informática Aplicada à Educação - NIED, 1999.

WEISS, Alba Maria Lemme; Cruz, Maria Lucia R.M. da. **A Informática e os problemas escolares de aprendizagem.** Rio de Janeiro: DP&A editora, 1998.



WINN, William. **The impact of three-dimensional immersive virtual environments on modern pedagogy.** HITL Technical Report. University of Washington, Seattle, 1997.