

Ferramenta para auxílio ao ensino de sistemas de  
tempo-real

Alisson Guindo Casagrande  
Odnei Cuesta Lopes

Orientador:  
Prof. Dr. Aleardo Manacero Júnior

São José do Rio Preto, SP - Brasil

Novembro de 2002

# Ferramenta para auxílio ao ensino de sistemas de tempo-real

Alisson Guindo Casagrande

Odnei Cuesta Lopes

Projeto Final de Curso submetido ao Departamento de Ciências de Computação e Estatística do Instituto de Biociências, Letras e Ciências Exatas (Ibilce) da Universidade Estadual Paulista Júlio de Mesquita Filho, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado por:

Prof. Dr. Aleardo Manacero Júnior

(Presidente)

Prof. Valencio ou Adriano

Prof. Fabio

---

Alisson Guindo Casagrande

---

Odnei Cuesta Lopes

---

Prof. Dr. Aleardo Manacero Júnior

São José do Rio Preto, SP - Brasil

Novembro de 2002

# Agradecimentos

## **Eu, Alisson, agradeço:**

Aos docentes deste instituto, especialmente ao professor Dr. Aleardo Manacero Júnior, pela amizade, atenção e a orientação durante todo período deste projeto.

Aos meus pais pelos ensinamentos, a preocupação e por todo o esforço que tiveram, para que fosse possível a conclusão de mais esta etapa de aprendizado.

A Milene pelo carinho, apoio e por ter sido companheira mesmos nos dias mais difíceis.

A todos os amigos, especialmente aos que conheci no Ibilce, com os quais sempre pude contar.

Enfim, agradeço a todos que me ajudaram, mesmo que indiretamente para que fosse possível a conclusão deste projeto.

A todos estes, serei eternamente grato.

## **Eu, Odnei, agradeço:**

A Deus pela oportunidade desta vida e por tudo que tem me permitido alcançar.

Aos meus pais por proverem a todos os momentos o que me foi necessário sempre acompanhado de muito amor, carinho e lições de vida.

Aos docentes que ensinaram-me ao longo de todos estes anos não só lições teóricas, mas de profissionalismo e ética. E especialmente ao professor Aleardo, que orientou meu crescimento ao longo dos últimos anos com prestatividade e clareza.

Aos meus amigos e colegas de curso, especialmente ao Alisson, que dividiram comigo os momentos bons e me apoiaram nos difíceis.

Espero que nossa amizade e companheirismo se fortaleçam ao longo da grande jornada que a vida nos reserva.

# Dedicatória

*Dedico aos meus pais (Aparecido e Daisy),  
às minhas irmãs (Aline e Aniele) e a Milene - Alisson.*

*Dedico este trabalho a minha família, de  
modo especial a minha mãe, que abrilhanta o papel  
heróico reservado às mulheres pela sua abnegação e  
amor a família - Odnei.*

## Resumo

O problema dos atuais sistemas de ensino auxiliado por computador é que eles sobrecarregam o aluno com informações desconsiderando seu estilo pessoal de aprender, ou seja, como ele gosta de absorver e processar o conteúdo do curso. Este projeto visa a construção de um módulo para o aperfeiçoamento de uma ferramenta de auxílio ao ensino de sistemas de tempo-real (RTsim) para que a mesma ofereça estilos pessoais de aprendizagem.

Os vários modelos de estilos de aprendizagem são divididos em quatro classes, referentes a personalidade, ao processamento de informações, a interação ambiental e a interação social. Visto que a classe de modelos referentes ao processamento de informações é mais adequada em uma ferramenta computacional, adotou-se o modelo de Kolb para a implementação desse projeto, pois é um dos mais representativos dessa classe.

No modelo de Kolb os indivíduos são classificados em quatro estilos, a partir dos perfis ativo/reflexivo e concreto/abstrato. Define-se, a partir destes perfis, as abordagens para cada um dos estilos.

Estruturou-se o conteúdo do curso utilizando-se a Linguagem Extensível de Marcação (XML), para que o mesmo pudesse ser apresentado de quatro modos sem gerar redundância e inconsistências de conteúdo. O controle de avanço no conteúdo é condicionado à execução de testes relativos a cada seção do mesmo sendo implementado em Java.

A adição deste módulo ao RTsim acrescentou a funcionalidade de apresentação de conteúdo e abriu a possibilidade de um acompanhamento da progressão dos alunos no curso pelo professor. Esta funcionalidade não causou diminuição no desempenho da ferramenta, apesar do aumento das necessidade de processamento.

## **Abstract**

The problem with today's computer-based learning systems is that they overload the student with information regardless their personal learning style, i.e., how the students prefer to take and process the course content. The goal of this project is the construction of a module to improve the functionality of a real-time systems leaning tool (RTsim) by the inclusion of personal learning styles on it.

There are several learning styles models, witch can be grouped in four classes: personality level, information processing, instructional preferences and social interaction models. The best class to be implemented in a computational tool is the information processing class, that has the Kolb's model as one of the most important models.

In the Kolb's model, the students are classified in four styles, according on their preferences between active experimentation/reflective observation and concrete experience/abstract conceptualization. The approaches for content presentation for each style were specified.

The course content have been structured using the eXtensible Markup Language (XML), aiming the content personalization for the four styles without introducing redundancies or inconsistencies. The student's advance throughout the course content is controled by tests applied to the student. This control has been implemented in Java.

The inclusion of this module to RTsim increased its funcionality on presenting content and introduced the possibility of student's assessment by the teacher. All of this additional processing did not reduce the tool's performance.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	2
1.2	Descrição do texto . . . . .	2
<b>2</b>	<b>Estilos de aprendizagem e tecnologia para implementação</b>	<b>4</b>
2.1	O modelo geral de Curry . . . . .	5
2.2	Modelos de estilos de aprendizagem . . . . .	6
2.2.1	Modelo de Dunn & Dunn . . . . .	7
2.2.2	Modelo de Kolb . . . . .	11
2.2.3	O indicador de tipos de Myers-Briggs . . . . .	18
2.3	Usando estilos de aprendizagem para ensinar . . . . .	23
2.4	Tecnologia para implementação . . . . .	24
2.4.1	Definição do tipo do documento (DTD) . . . . .	25
2.4.2	O documento XML . . . . .	27
2.4.3	Linguagem extensível de folha de estilo (XSL) . . . . .	29
2.5	Considerações finais . . . . .	30
<b>3</b>	<b>Aplicando estilos de aprendizagem em ensino auxiliado por computador</b>	<b>32</b>
3.1	Definição do modelo de aprendizagem . . . . .	32
3.2	Planejamento do método de abordagem . . . . .	34
3.2.1	Divergentes - <i>Reflexivo/Concreto</i> . . . . .	34
3.2.2	Assimiladores - <i>Reflexivo/Abstrato</i> . . . . .	35
3.2.3	Convergentes - <i>Abstrato/Ativo</i> . . . . .	35

3.2.4	Acomodadores - <i>Ativo/Concreto</i> . . . . .	36
3.2.5	Abordagens para os grupos . . . . .	36
3.3	Implementando a ferramenta de auxílio ao ensino . . . . .	37
3.3.1	Componentes do servidor . . . . .	38
3.3.2	Componentes do cliente . . . . .	41
3.3.3	Funcionamento do RTTutor . . . . .	45
3.4	Estruturação do conteúdo do curso . . . . .	46
3.4.1	Implementação do conteúdo . . . . .	47
3.4.2	Especificação da DTD e suas implicações . . . . .	49
<b>4</b>	<b>Resultados Obtidos</b>	<b>53</b>
4.1	Capacidades acrescentadas . . . . .	53
4.2	Facilidade de uso e manutenção . . . . .	54
4.3	Portabilidade . . . . .	54
4.4	Desempenho . . . . .	55
<b>5</b>	<b>Conclusões e perspectivas futuras</b>	<b>58</b>
5.1	Conclusões . . . . .	58
5.2	Perspectivas futuras . . . . .	59
<b>A</b>	<b>Inventário de estilos de aprendizagem de Kolb</b>	<b>63</b>
A.1	Descrição dos estilos de aprendizagem . . . . .	65
A.1.1	Divergentes . . . . .	65
A.1.2	Assimiladores . . . . .	67
A.1.3	Convergentes . . . . .	68
A.1.4	Acomodadores . . . . .	68



# Lista de Figuras

2.1	Grade de estilos de aprendizagem . . . . .	13
2.2	Grade de McCarthy . . . . .	18
2.3	Trecho de uma DTD exemplo . . . . .	25
2.4	Trecho de XML exemplo . . . . .	28
2.5	Trecho de código XSL para o exemplo . . . . .	31
3.1	Componentes dos módulos servidor e cliente . . . . .	37
3.2	Esquema do RTTutor . . . . .	38
3.3	Questão do teste de Kolb . . . . .	42
3.4	Gerenciador de Conteúdo . . . . .	44
3.5	Funcionamento do RTTutor . . . . .	45
3.6	Trecho da DTD . . . . .	50
3.7	Trecho do documento XML . . . . .	51
3.8	Trecho do XSL . . . . .	52
A.1	O circulo de aprendizado . . . . .	66
A.2	Grade de estilos de aprendizado . . . . .	66

# Lista de Tabelas

4.1	Relação de portabilidade dos módulos . . . . .	55
4.2	Tempos de resposta sem carga no servidor . . . . .	56
4.3	Tempos de resposta com carga no servidor . . . . .	56
A.1	Questões do teste de Kolb . . . . .	65

# Capítulo 1

## Introdução

Atualmente, modernas tecnologias de informação e de comunicação são utilizadas no relacionamento humano e, de modo especial, no relacionamento professor-aluno. Um exemplo disto é a instrumentalização do processo de ensino-aprendizagem, com materiais informativos e/ou ambientes educacionais virtuais de apoio à aprendizagem, tendo como objetivo enriquecer o processo educacional.

No entanto, a tecnologia tem transportado a problemática do ensino convencional para as ferramentas de ensino computacionais, ou seja, elas empregam uma única abordagem que geralmente reflete o modo com que o professor gosta de aprender. Por outro lado pode-se incorrer em um outro erro tentando-se exibir um grande volume de informações, visando abranger todos os tipos de preferências encontrados nos estudantes.

Os profissionais de computação podem tomar posturas que alimentem o modelo vigente ou buscar novas posturas, ou seja, restringirem-se a procurar soluções através de respostas de cunho tecnológico-instrumental, ou buscar métodos orientados por estudos sobre o comportamento humano, substituindo o grande fluxo de informações atual, por um fluxo orientado de conhecimento.

Este fluxo orientado de conhecimento é alcançado pela construção de um ambiente educacional que leve em consideração a complexidade de interações entre o planeta, a sociedade, os homens e os recursos técnicos. Assim, este ambiente deve integrar conteúdos e abordagens diferentes para os vários estilos de aprendizagem.

Nota-se ainda, que este modelo de interação é uma das preocupações da engenharia

de software:

*A interface com o usuário é o mecanismo por meio do qual se estabelece um diálogo entre o programa e o ser humano. Se os fatores humanos (entendimento do usuário e seu comportamento) tiverem sido levados em conta, o diálogo poderá ser harmonioso e um ritmo será estabelecido entre o usuário e o programa. Se os fatores humanos tiverem sido ignorados, o sistema quase sempre será visto como “não amigável” [Pre95].*

## 1.1 Objetivos

Durante a disciplina de sistemas de tempo-real, utiliza-se como ferramenta auxiliar ao ensino um simulador de escalonadores de sistemas de tempo-real - RTsim, que foi inicialmente desenvolvido como um mecanismo de auxílio para projetistas destes sistemas. Devido a esta origem esta ferramenta não possui características adequadas ao processo de ensino por não apresentar conteúdo, facilitando somente a resolução de exercícios de escalonamentos.

É neste contexto que se insere a necessidade de uma ferramenta complementar que terá como objetivo auxiliar o ensino de sistemas de tempo-real, a qual estará fundamentada no entendimento dos fatores humanos e, portanto oferecendo abordagens diferentes para estilos diversos.

## 1.2 Descrição do texto

Apresenta-se inicialmente, no capítulo 2, os fundamentos teóricos sobre alguns modelos de estilos de aprendizagem, que tratam das características de cada indivíduo relativas ao modo como o mesmo aprende, e diretrizes de como utilizá-los para ensinar. Ainda no capítulo 2, vê-se tecnologias de implementação que evitem redundâncias, facilite a manutenção e portabilidade tanto do conteúdo quanto da ferramenta. E finalmente, apresenta-se algumas abordagens que são propostas pelos modelos.

A seguir, no capítulo 3, avalia-se segundo as características apresentadas por cada modelo, qual se adequa de modo mais eficaz tanto a detecção do estilo do aluno, quanto as abordagens propostas para a implementação em uma ferramenta computacional. A partir de então, define-se o método de abordagem segundo as propostas do modelo para as características específicas de cada um dos estilos e também para as características comuns entre eles. Por fim, apresenta-se detalhes de implementação da ferramenta e como as abordagens foram mapeadas para a estruturação do conteúdo.

No capítulo 4, expõe-se os resultados obtidos com a construção da ferramenta e as características apresentadas pela mesma em relação a facilidade de uso e manutenção, portabilidade e desempenho.

Finalmente, no capítulo 5, apresenta-se algumas conclusões sobre o desenvolvimento deste projeto e também indica-se perspectivas de novos projetos complementares e possíveis direções de trabalhos na área.

## Capítulo 2

# Estilos de aprendizagem e tecnologia para implementação

Tem-se a idéia de que os estudantes aprendem e se desenvolvem através da exposição ao conteúdo - que é muito valorizado - e portanto, no processo de aprendizagem tradicional o professor simplesmente apresenta sua concepção do o conteúdo ao estudante.

Entretanto, os estudantes não são todos iguais e, portanto, eles aprendem de maneira diferente. Assim, desejando obter sucesso no processo de ensino é necessário utilizar-se estilos de ensino que se enquadrem no modo de aprendizagem dos estudantes.

O ambiente escolar é melhor quando inclui momentos intuitivos e subjetivos, acrescentando horas criativas, coleta de informação, análise sistemática, tanto quanto avaliações reflexivas e material bem escrito. O ensino é melhor quando planeja-se maneiras diversas de ensinar que acomodem as potencialidades de cada estudante, facilitando assim o aprendizado.

Comparando-se um ambiente complexo como o acima, com uma sala de aula típica, onde a maioria dos professores tende a confiar quase exclusivamente nas apresentações seqüenciais, verbais, combinadas com atividades individuais de leitura e escrita, constatou-se que nesta segunda abordagem os estudantes são forçados a pensar de forma complexa antes de conseguirem terminar um projeto, pois eles são expostos a uma única abordagem do conteúdo que pode ser e, geralmente é, limitada na variedade de elementos adequados a diversos estilos. A perda das oportunidades de acoplar uma variedade de orientações na

abordagem transforma-se em uma falha óbvia para aqueles que reconhecem a inevitável diferença entre os diversos pontos de vista. Dessa forma, percebe-se o mal que isso causa quando vê-se a falta de motivação, a resistência à abordagem, a falta de atenção e de inspiração, que decorrem do fato de que a grande maioria dos estudantes não pode aprender bem com a orientação limitada que é apresentada dentro da sala de aula.

Neste capítulo apresenta-se alguns modelos que são provenientes de pesquisas na área de estilos de aprendizagem, como também algumas tecnologias que tornam possível e eficiente a implementação destes modelos.

## 2.1 O modelo geral de Curry

Devido a variedade dos modelos usados para caracterizar os estilos de aprendizagem, o modelo de Curry [Cur87] [Cur90], que pesquisa as diferenças da aprendizagem humana é útil. Curry divide estes estudos em três níveis, que podem ser examinados como as diferentes camadas de uma “cebola”, utilizando-se da combinação das características nos diversos níveis para explicar as variações humanas. Recentemente, a “cebola”, foi redividida nos quatro níveis descritos abaixo, sendo que os principais modelos citados nessa descrição serão melhor descritos na próxima seção.

- **Nível de preferências instrutivas e ambientais**, é referente aos traços mais visíveis, ou seja, as preferências ambientais do estudante no momento do aprendizado. O modelo de Dunn & Dunn, por exemplo, identifica 5 dimensões que marcam as várias preferências, que são as preferências ambientais, emocionais, sociais, fisiológicas e psicológicas.
- **Nível de interação social**, em que se considera as maneiras como pessoas inseridas em contextos sociais específicos usam certas estratégias. Um modelo bem conhecido é o de William Perry [Ctl02] que mostra como os estudantes universitários desenvolvem-se através de diferentes níveis de maturação intelectual de acordo com o que vivem na faculdade.

- **Nível de processamento de informação**, é a camada intermediária da “cebola”, e representa como as informações são assimiladas, ou seja, classificadas, armazenadas e utilizadas. A idéia mais conhecida sobre as diferenças no processamento de informações é a discussão sobre os lados (direito/esquerdo) do cérebro. Um modelo mais complexo é o de Kolb [Kol76], que é um estudo experimental sobre o aprendizado, sendo também um dos modelos mais utilizados.
- **Nível de personalidade**, é o centro da “cebola”, representando como a personalidade, em um nível mais profundo, fornece a forma e a orientação com as quais o indivíduo examina o mundo. Um dos modelos mais conhecido é o de Myers-Briggs. Este modelo categoriza os indivíduos como extrovertidos/introvertidos, sensoriais/intuitivos, racionais/sentimentais, e decididos/indecisos. A classificação dos indivíduos ao longo da escala indica suas tendências e atitudes perante o mundo. Modelos desse nível enquadram as preferências pessoais na forma de ser. A questão não resume-se simplesmente a treinar habilidades ou atitudes, mas reconhecer que os indivíduos agem de modo fundamentalmente diferente na sala de aula. Deste modo, não é possível obter o sucesso no ensino sem considerar os motivos que objetivaram o ingresso do estudante no curso ou disciplina [Ct102].

## 2.2 Modelos de estilos de aprendizagem

Todos os modelos partem do princípio de que os estudantes aprendem melhor quando os ensinamentos são apresentados de maneira que eles se identifiquem e se sintam seguros. Se na sua orientação extraem a teoria de experiências concretas, então o seu aprendizado será melhor “fazendo” algo do que “refletindo” sobre os conceitos. Se seu estilo pessoal é orientado à abstração, então aprenderão melhor de forma abstrata. De fato, um indivíduo não pode fixar seus conhecimentos enquanto estes não lhe sejam apresentados na forma que ele confia.

Alunos que gostam de ouvir aprenderão melhor assistindo aulas; os que são individualistas obterão conhecimento através de leituras realizadas individualmente. Contudo, estes são somente dois exemplos de um grande grupo de preferências encontradas entre



as capacidades intelectuais dos indivíduos. Quando as experiências de aprendizagem se limitam a estas abordagens, os estudantes que sentem-se seguros com outros estilos serão certamente menos bem sucedidos. Ambientes de aula limitados inibem um ou mais grupos de alunos que preferem estilos que não são usados.

Uma instituição de ensino pode melhorar as possibilidades de aprendizado dos estudantes apenas aumentando o contato dos mesmos com variadas abordagens com as quais os estudantes possam se identificar. Neste caminho, o uso de tecnologia fornece novas potencialidades para a construção de ambientes de aprendizado em torno de estilos específicos.

A seguir, expõe-se de forma mais detalhada, os principais modelos de estilos de aprendizagem.

### **2.2.1 Modelo de Dunn & Dunn**

Para entender como o uso do modelo de estilos de aprendizado de Dunn & Dunn[Ler02] pode ajudar o educador, é preciso entender como o modelo é organizado e como pode ser utilizado para o ensino e para o aprendizado. Os componentes principais do modelo detalhados adiante são:

- A. Os princípios do modelo;
- B. Os elementos do estilo de aprendizado.
  - 1. A identificação do estilo de aprendizado;
  - 2. Impacto do modelo nas situações de ensino.

#### **A. Os princípios do modelo**

O modelo de estilos de aprendizagem de Dunn & Dunn identifica algumas regras principais ou argumentos teóricos, que devem ser seguidos pelos administradores e equipes que utilizarem o modelo. Essas regras são:

- 1. A maioria dos indivíduos pode aprender;

2. Ambientes de ensino, recursos, e abordagens atuam diversificando as capacidades do estilo de aprendizado;
3. Todas as pessoas possuem capacidades, mas pessoas diferentes possuem capacidades diferentes em uma ampla variedade de combinações;
4. Preferências de ensino individuais existem e podem ser medidas de modo confiável;
5. Possuindo ambientes de ensino, recursos, e abordagens diferentes os estudantes conseguem estatisticamente melhores notas em testes de assimilação e atitudes;
6. A maioria dos educadores pode aprender como usar estilos de aprendizagem como base para ensinar;
7. Muitos estudantes podem aprender a se apegar nas capacidades inerentes ao seu estilo de aprendizagem quando deparam-se com matérias novas ou difíceis.

## **B. Os elementos do estilo de aprendizagem**

O uso do modelo de estilo de aprendizagem de Dunn & Dunn envolve dois tipos de atividades principais:

1. A identificação dos estilos de aprendizagem individuais; e
2. O planejamento e implementação do ensino para acomodar as capacidades do estilo de aprendizagem individual dos estudantes.

O conjunto de preferências é uma serie de 21 “elementos de estilos de aprendizagem” como definido por Dunn & Dunn. Estes elementos estão agrupados em 5 categorias de “estímulos”: preferências ambientais, emocionais, sociológicas, fisiológicas, e psicológicas (processamento cognitivo).

1. Estímulos ambientais:

- (a) *Som*: refere-se a preferência do estudante em ouvir sons enquanto aprende, e pode variar entre silêncio, barulhos ou músicas.
- (b) *Luz*: refere-se ao nível de luz que é preferido enquanto se estuda ou aprende, e pode variar entre pouca luz, luz indireta ou luz direta.
- (c) *Temperatura*: refere-se a qual nível de temperatura que o estudante prefere enquanto estiver estudando e/ou em atividades de aprendizado. As preferências deste elemento podem variar entre ambiente quente ou frio.
- (d) *Projeto do ambiente*: refere-se ao ambiente de estudo e o modo como a mobília do mesmo está disposta. Pode variar do tradicional, ou seja, estudar usando mesa e cadeira, ou um arranjo informal com diferentes tipos de mobília, como sofás, cadeiras reclináveis, carpetes e travesseiros no chão, etc.

2. Estímulos emocionais:

- (a) *Motivação*: refere-se ao nível e tipo de motivação que o estudante tem no aprendizado, ou seja, o quanto o estudante está interessado no ensino escolar. Pode variar de auto-motivação; motivado através de seus interesses; contatos com parceiros; ou motivação através de incentivos de uma pessoa mais experiente.
- (b) *Persistência*: refere-se a persistência do estudante nas tarefas do aprendizado, ou seja, à atenção e interesse que o estudante dispensa na execução de uma tarefa pertencente ao processo de aprendizado. Pode variar entre preferir trabalhar em uma tarefa até terminá-la ou trabalhar em várias tarefas ao mesmo tempo.
- (c) *Responsabilidade*: refere-se a preferência de trabalhar independentemente, com pouca supervisão, cobrança, ou prestação de contas, ou seja, o aluno prefere trabalhar sem uma pessoa mais experiente dizendo o que ele deve fazer ou prefere ter alguém freqüentemente orientando o seu estudo?
- (d) *Estrutura*: refere-se a preferência por um aprendizado com atividades e tarefas estruturadas. Pode variar do desejo de ser informado exatamente o que é a

tarefa que esta aprendendo, como ele deve proceder e o que é esperado dele, ou preferir que lhe seja apresentado um objetivo e lhe seja dada a liberdade para decidir como alcançar o objetivo por conta própria.

### 3. Estímulos sociológicos:

- (a) *Individualismo*: refere-se a preferência por trabalhar de forma individual ou em grupo. Varia em estudar sozinho, com a ajuda de um grupo ou dependendo do tipo de tarefa que estão aprendendo.
- (b) *Pares*: Refere-se ao trabalho junto com outro estudante. Varia em preferir estudar com uma ou outra pessoa.
- (c) *Parcerias*: refere-se a preferência do estudante por estudos em pequenos grupos de estudantes com muita interação, discussão e completar as tarefas como um time, ou a preferência por estudar sozinho.
- (d) *Interação e supervisão*: refere-se a preferência por interação e orientação de um supervisor (pessoa experiente) ou preferir menos interação e supervisão.
- (e) *Variação*: refere-se à preferência de envolvimento em uma variedade de tarefas enquanto aprende. Por exemplo, o aluno gosta de rotinas ou prefere uma variedade de processos e atividades enquanto aprende.

### 4. Estímulos fisiológicos:

- (a) *Percepção*: refere-se a se o estudante se sente mais ou menos estimulado quando as atividades envolvem materiais visuais (vendo pinturas, mapas ou lendo), atividades auditivas (ouvir fitas, leituras e músicas), ou prefere envolvimento tátil, como tomar notas, e/ou estudar em projetos que envolvem atividades práticas fazer coisas (por exemplo, projetos de ciência, relatórios, etc.).
- (b) *Ingestão*: refere-se a necessidade do estudante de comer, beber, ou mastigar enquanto realiza as atividades do aprendizado.
- (c) *Tempo*: refere-se a variação dos níveis de energia em diferentes horários do dia. Varia entre preferir realizar uma tarefa que necessita de concentração no início ou no fim de certo período do dia (manhã, tarde ou noite).

- (d) *Mobilidade*: refere-se a preferência do estudante por sentar-se por longos períodos de tempo ou mover-se constantemente, mesmo que inconscientemente, enquanto aprende.
5. Estímulos psicológicos:
- (a) *Analítico-global*: refere-se a preferência do estudante por consideração dos conceitos de forma global ou de forma sequencial ao longo do tempo. Os estudantes que tem preferência por estudo global estão interessados no significado do que estão estudando em relação ao todo; precisam começar com uma visão geral do conteúdo antes de partir para o detalhamento dos conceitos. Por outro lado, os que preferem um estilo analítico de aprendizado gostam de aprender em detalhes cada tópico em uma seqüência cronológica, e a partir do momento em que eles conhecem todas as partes eles podem juntá-las e compreender o todo.
- (b) *Hemisfério*: refere-se ao lado do cérebro (direito ou esquerdo) que predomina no momento do aprendizado. Se o lado esquerdo predomina, o estudante tende a ser mais analítico e seqüencial, enquanto que se o direito predomina a tendência é ser global e simultâneo.
- (c) *Impulsivo-reflexivo*: refere-se ao montante de tempo que o estudante leva pensando para chegar a uma solução ou conclusão. Varia entre tomar decisões rapidamente, ou preferir ter um tempo para ponderar sobre as várias alternativas que se lhe apresentam antes de tomar uma decisão.

### 2.2.2 Modelo de Kolb

O Modelo de Kolb[Kol76] trabalha como um inventário de estilos de aprendizagem para fazer a identificação dos estilos de aprendizagem. Este inventário é composto de algumas sentenças às quais estão associadas alternativas. Cada alternativa recebe um peso de acordo com o que o estudante acredita que melhor descreve suas atitudes e sentimentos no momento em que ele está aprendendo.

## As quatro dimensões do inventário de estilos de aprendizagem de Kolb

A partir dos pesos que o estudante atribui para as alternativas são calculados quatro índices: experiência concreta, conceitualização abstrata, observação reflexiva e experimentação ativa. O significado desses índices é visto a seguir:

- **Experiência concreta:** Kolb estabelece que um alto índice em experiência concreta representa uma receptividade a abordagem baseada em experiências, de modo que o aprendizado se baseia em ponderações baseadas em sentimentos. Os indivíduos deste estilo tendem a ser empáticos. Eles geralmente acham abordagens teóricas inúteis e preferem tratar cada situação como um caso único. Aprendem melhor por meio de exemplos específicos nos quais se sintam envolvidos. Estes estudantes tendem a se relacionar melhor com outros estudantes, do que com uma autoridade como o professor.
- **Conceitualização abstrata:** um alto índice em conceitualização abstrata indica um modo de aprendizado analítico e conceitual, que se baseia pesadamente em raciocínio lógico. Estes indivíduos tendem a ser mais orientados a coisas e símbolos, do que a outras pessoas. Aprendem melhor quando orientados por uma autoridade de modo impessoal, com ênfase teórica e análise sistemática. Eles se sentem frustrados e aprendem pouco pelo aprendizado através de descobertas de modo desestruturado, como em exercícios e simulações.
- **Observação reflexiva:** um alto índice em observação reflexiva indica uma abordagem por tentativas, imparcial e reflexiva. Estes indivíduos aprendem baseando-se fortemente em cuidadosas observações e fazendo julgamentos das mesmas. Eles preferem aprender assistindo aulas, o que lhes dá a possibilidade de exercer o seu papel de observador e juiz imparcial; tendem a ser introvertidos.
- **Experimentação ativa:** um alto índice em experimentação ativa indica uma disposição forte em realizar atividades práticas. Estes indivíduos aprendem mais facilmente quando participam de projetos práticos, discussões em grupo e fazendo

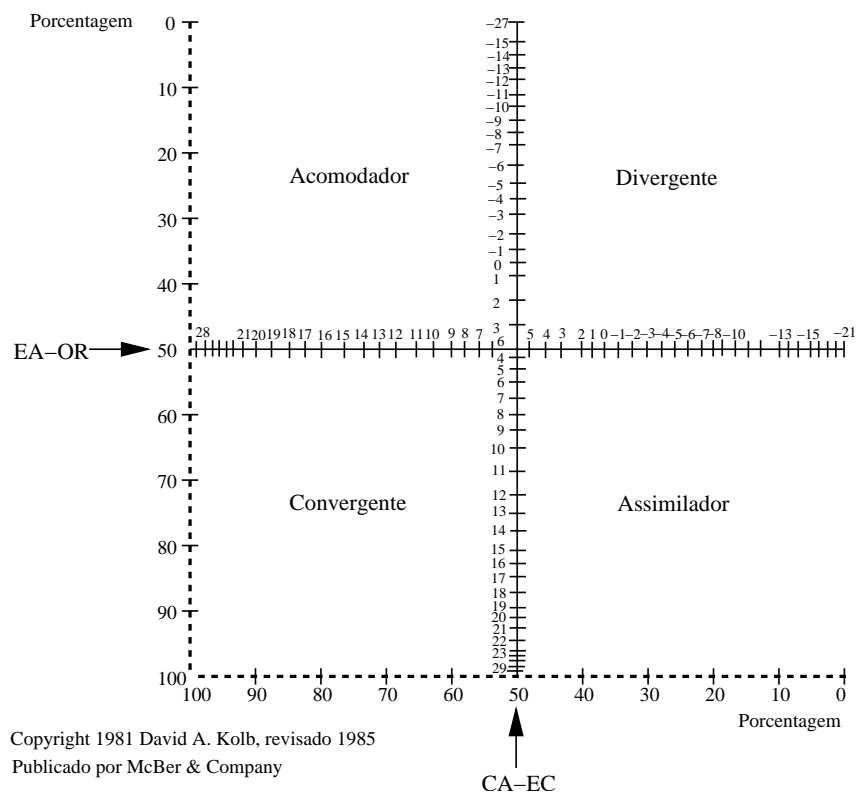


Figura 2.1: Grade de estilos de aprendizagem

tarefas em casa. Eles não gostam de situações de aprendizado passivo como assistir a aulas, e tendem a ser extrovertidos.

### A grade de estilos de aprendizagem

Após obter os índices em cada dimensão acima, o estudante deve fazer a composição dos mesmos da forma: Conceitualização abstrata/Experiência concreta (CA - EC) e Experimentação ativa/Observação reflexiva (EA - OR) e então marcar seus pontos em eixos graduados. Assim, como uma função de duas variáveis, o estudante pode colocar estes valores num gráfico (figura 2.1) e então seu estilo de aprendizagem será o quadrante no qual a intersecção das retas, que passam pelos pontos marcados nos eixos, estiver.

### Descrição dos estilos

Kolb denomina os estilos de aprendizagem, como: Divergente (“Diverger”), Assimilador (“Assimilator”), Convergente (“Converger”) e Acomodador (“Accommodator”). Indexa-

se esses estilos segundo as dimensões definidas acima da seguinte forma:

- Experiência concreta - Concreto;
- Conceitualização abstrata - Abstrato;
- Observação reflexiva - Reflexivo;
- Experimentação ativa - Ativo.

Segue abaixo uma descrição dos estilos de aprendizagem baseando-se nesses índices:

- **Tipo 1 - Divergentes (concreto, reflexivo)** a questão típica deste estilo de aprendizagem é “Por quê?”. Os estudantes deste estilo respondem bem a explicações de como o material estudado se refere a suas experiências, interesses e carreiras futuras. Divergentes captam as informações através de experiência concreta, baseando-se nos sentimentos, precisam expressar seus sentimentos quando estão aprendendo, procurando por significados pessoais sobre o que aprendem, e querem interagir com professores e colegas. Eles precisam ser convencidos de que o material estudado é importante para eles como pessoas e/ou para outras pessoas as quais eles sintam empatia ou se identifiquem. Aprendem através de discussões e são interessados em como as informações afetam as pessoas. Trabalham para manter o grupo harmonioso e são chamados Divergentes porque podem ver as coisas de diferentes perspectivas e geram novas idéias facilmente. Assim saem-se muito bem em reuniões do tipo “brainstorming”.

Para obter sucesso com estes estudantes deve-se atuar como motivador, motivando-lhes a aprender o conteúdo e mostrando a eles como o conteúdo se encaixa em suas vidas. Divergentes querem interagir pessoalmente com seus instrutores e serem reconhecidos como indivíduos. Assim, o instrutor deve testemunhar o crescimento pessoal do estudante.

- **Tipo 2 - Assimiladores (abstrato, reflexivo)** a questão típica deste estilo de aprendizagem é “O que?”. Estudantes deste estilo são bem sucedidos quando as informações são apresentadas de maneira organizada e lógica, e obtêm aproveitamento melhor se tiverem tempo para refletir.



Assimiladores captam as informações através de conceitualização abstrata e as processam através de observação reflexiva. Eles gostam de informações que justifiquem o que estão aprendendo, querem saber o que os especialistas pensam e procuram compreender os conceitos. São bem sucedidos na escola tradicional. De fato, a maioria dos estudantes, apesar de terem seus estilos próprios, se adaptam ao estilo do Assimilador para obter sucesso na escola. Eles são chamados Assimiladores porque utilizam-se de pedaços de informações, analisam, organizam, e assimilam-os como um todo. Gostam de organizar, e tem tendência a serem detalhistas. Tem sucesso nos procedimentos quando se baseiam na literatura. Nas suas abordagens são cuidadosos, metódicos e cautelosos, tentando evitar erros.

Eles aprendem bem lendo sobre os tópicos e preferem não trabalhar em grupos, exceto para seguir normas. Para ser efetivo, o instrutor deve se comportar como um especialista. Assim, o papel do instrutor é unicamente ser uma autoridade no assunto e uma fonte de informações. Assimiladores gostam de assistir aulas e/ou palestras e preferem que o professor explique o conteúdo de forma organizada e precisa.

- **Tipo 3 - Convergentes (abstrato, ativo)** a questão típica deste estilo de aprendizagem é “Como?”. Os estudantes deste estilo gostam de ter a oportunidade de trabalhar ativamente em tarefas bem definidas, e de aprender por tentativa e erro em um ambiente que permita-lhes errar com segurança.

Os Convergentes captam as informações através de conceitualização abstrata e as processam de forma ativa. Gostam de testar as informações, tentar coisas, perceber coisas a parte, ver como as coisas funcionam, e aprender fazendo. Testam a viabilidade e utilidade das informações. Assim, não gostam de permanecerem sentados por um longo período, assistindo aulas ou lendo uma grande quantidade de material. Eles dão ênfase a fazer as coisas sem perder tempo; são aprendizes que convergem mais rapidamente ou chegam rapidamente a tomar decisões, que procuram por uma resposta correta, e que cortam caminho, passando somente pela parte mais essencial da matéria. Gostam de exemplos a seguir, em detrimento de assistir a aulas ou

palestras. O ambiente de laboratório é ideal para eles. Tendem a ver o trabalho em grupo e as discussões como perda de tempo, porque podem fazer o trabalho mais rapidamente de forma individual.

Para obter sucesso com os convergentes, o instrutor deve trabalhar como um técnico, provendo práticas orientadas e respostas. Esses estudantes preferem que o professor os ajude a colocar as lições em prática. O instrutor precisa se tornar menos ativamente envolvido, agindo como uma fonte de informações e permitir ao Convergente ter um papel mais ativo.

- **Tipo 4 - Acomodadores (concreto, ativo)** a questão típica deste estilo de aprendizagem é “E se?”. Os estudantes deste estilo gostam de aplicar as informações em novas situações para resolver problemas reais. Os Acomodadores captam as informações através de experiências concretas e processam-nas através de experimentação ativa. Eles são entusiastas e preferem aprender através de descobertas próprias. Seguem seus próprios horários e ordem quando estudam e não gostam de muitos procedimentos e regras. Por outro lado, gostam de interação com outras pessoas, de discussões em grupo, de obter informações e criar algo novo com elas, e ainda de descobrir coisas por sua própria conta. São solucionadores de problemas e assumem riscos, aprendendo com seus erros; são chamados de Acomodadores porque usam o que aprenderam adaptando estes conhecimentos para uso próprio; utilizam sua criatividade para realizar esta adaptação e fazer as coisas cada vez melhores.

Para obter sucesso com alunos deste estilo o instrutor deve possibilitar ao estudante descobrir as coisas por conta própria. O papel do instrutor é o de avaliador e revisor. Esses estudantes precisam do professor para encorajá-los às descobertas, permitindo-lhes se auto-instruirmos. Eles preferem alguma interação com o instrutor, mas pouca supervisão e querem que o instrutor permaneça em segundo plano, para servir como um recurso extra, e para avaliar o que eles estão fazendo.

## Projetando métodos de ensino

Pesquisas sugerem que a aprendizagem mais eficaz ocorre quando os métodos de ensino se encaixam no estilo preferido dos estudantes e que alguns métodos de ensino são mais úteis para alguns estudantes que para outros.

Assim, visto que os conteúdos geralmente não são desenvolvidos considerando todos os estilos de aprendizagem, e levando em consideração que algumas informações são mais viáveis quando utiliza-se um método ao invés de outro, os estudantes devem ter a capacidade de se adaptar a mais de uma abordagem.

- **Experiência concreta:** Aulas teóricas não são muito úteis enquanto que o trabalho em grupo e em pares conduzem freqüentemente ao sucesso. As atividades devem ser planejadas para desenvolver as capacidades que devem ser aprendidas. O instrutor deve agir como um treinador/técnico para este aprendiz autodidata.
- **Observação reflexiva:** Aulas são úteis a este estudante que espera que o instrutor atue como um especialista, expondo seu ponto de vista e fornecendo atividades e orientações. Este estudante quer que seu desempenho seja medido por critérios diferentes dos seus.
- **Conceitualização abstrata:** Estudos de caso, aulas teóricas e exercícios de reflexão ajudam este estudante, de resto, muito pouca coisa pode ajudá-lo.
- **Experimentação ativa:** Resolver problemas, pequenos grupos de discussões ou jogos, discussões em dupla e estudos auto-direcionados ajudam este estudante, que gosta de ver tudo e de determinar seus próprios critérios para a relevância dos materiais.

## Modelo de Bernice McCarthy

Bernice McCarthy[Mcc02] usou o inventário de estilo de aprendizagem de Kolb para criar um modelo que descrevesse os estilos de ensinar e as atividades requeridas para satisfazer aos estilos de aprendizagem descritos por Kolb. Quando a maioria dos estudantes de uma classe são Divergentes e Assimiladores, o instrutor deve assumir um papel ativo de

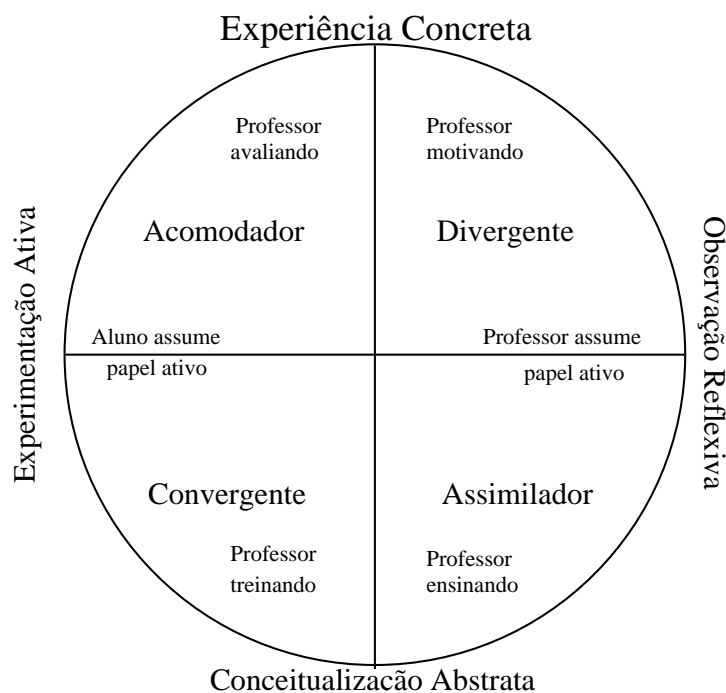


Figura 2.2: Grade de McCarthy

liderança e quando a maioria dos estudantes são Convergentes e Acomodadores o instrutor deve deixá-los assumir um papel mais ativo.

McCarthy explica também que o papel do instrutor é diferente para cada estilo de aprendizagem (figura 2.2).

Para o Divergente, o instrutor deve trabalhar motivando o aluno, mas ao trabalhar com Assimiladores ele deve trabalhar expondo conceitos. O Convergente requer práticas e acompanhamento pessoal, enquanto que o Acomodador necessita integrar aplicações com experiências.

### 2.2.3 O indicador de tipos de Myers-Briggs

O indicador de tipos de Myers-Briggs (ITMB) é um instrumento amplamente utilizado e é uma ferramenta muito útil na compreensão do papel das diferenças individuais no processo de aprendizagem. As pontuações obtidas no ITMB indicam a preferência de um estudante em cada uma das quatro dimensões disjuntas propostas pelo modelo, que são atitude extrovertida/introvertida, percepção sensorial/intuitiva, decisão racional/sentimental e ação

decidida/indecisa. Essas dimensões são descritas a seguir, sendo que as duas primeiras são particularmente úteis na compreensão dos estilos de aprendizagem.

- **Extrovertidos versus introvertidos:** indica se um estudante prefere dirigir a atenção para o mundo externo, as pessoas e coisas ou para o mundo interno, os conceitos e idéias.

Esta preferência diz-nos de onde as pessoas obtém sua energia. Os introvertidos obtém suas energias do seu mundo interior de idéias, conceitos, e abstrações. Eles podem ser sociáveis mas precisam de tranqüilidade para recuperar suas energias. Querem compreender o mundo, então se concentram e tendem a reflexão. Os introvertidos refletem em cada situação e decisão. Seu lema é: “todas as coisas vêm para aqueles que esperam!”.

Os extrovertidos obtém suas energias das coisas e pessoas. Preferem a interação com outras pessoas, e tendem a ser orientados à ação. Eles pensam enquanto fazem e desejam expressar cada pensamento. Seu lema é: “Porque deixar para amanhã o que você poderia ter terminado ontem!”.

- **Sensoriais versus intuitivos:** indica se um estudante prefere perceber o mundo através da observação direta da realidade a sua volta ou através de impressões e da imaginação de possibilidades.

Alguns estudantes usam seus cinco sentidos para absorver a informação, outros preferem utilizar um “sexto” sentido. Pessoas sensitivas são orientadas a detalhes e buscam por fatos tomando-os por verdade.

As pessoas intuitivas procuram padrões e relacionamentos entre os fatos que recolheram. Confiam em seus palpites e intuições e procuram por uma visão geral.

- **Racionais versus sentimentais:** indica como o estudante toma decisões, se é através de lógica ou usando critérios justos e valores humanos.

Alguns estudantes tomam decisões de modo impessoal baseando-se em análise lógica, e princípios básicos; outros baseiam-se em valores humanos. Os estudantes racionais

avaliam a justiça, ou seja, o que é mais justo tomando como base a lógica da situação, colocando ainda, um peso maior em critérios objetivos para tomar uma decisão.

Os estudantes sentimentais avaliam a harmonia, enfocando valores e necessidades humanas no momento de tomar decisões, tendendo a ser bons em persuasão e em resolver diferenças entre membros do grupo.

- **Decididos versus indecisos:** indica se o estudante vê o mundo como um ambiente estruturado e planejado ou como um ambiente espontâneo.

Alguns estudantes preferem postergar a ação e procurar por mais dados. Outros gostam de tomar decisões rapidamente. Pessoas decididas são inovadoras e auto-disciplinadas. Eles dão enfoque em terminar as tarefas, se preocupam somente em saber os fundamentos, e partem para a ação rapidamente. Eles planejam seus trabalhos e seguem os planos. Os prazos finais são sagrados.

Os estudantes indecisos são curiosos, flexíveis, e espontâneos. Começam muitas tarefas, querem saber tudo sobre cada uma, e freqüentemente acham difícil terminá-las. Os prazos finais servem somente para serem prorrogados, enquanto mais informações são recolhidas.

A partir das descrições para cada tipo em cada dimensão é possível delinear alguns perfis desejáveis no processo ensino-aprendizagem, como os descritos a seguir:

- **Ensinando estudantes introvertidos:** Os estudantes introvertidos querem desenvolver estruturas que integrem ou conectem as informações que eles estejam aprendendo. Para um introvertido, informações desconexas não são conhecimentos, são meramente informações. O conhecimento é a interconexão do material e a visão global.

Os instrutores devem ensinar a estes estudantes como agrupar e interconectar informações. Os estudantes introvertidos apreciarão este enfoque, enquanto que os extrovertidos não.

- **Ensinando estudantes extrovertidos:** Os estudantes extrovertidos aprendem ensinando outras pessoas. Eles não sabem se compreenderam o assunto até que

tentem explicá-lo a si mesmo ou a outras pessoas. Somente assim eles se convencem de que compreenderam o material.

Os estudantes extrovertidos apreciam o trabalho em grupo. A aprendizagem baseada em solução de problemas e a aprendizagem colaborativa são boas técnicas de ensino para este grupo.

Todos os dois tipos de estudantes suportam ensino através de explicações, desde que se forneça o intervalo de tempo requerido pelos introvertidos.

- **Ensinando estudantes sensitivos:** Estudantes sensitivos preferem aulas organizadas, lineares, e estruturadas, com métodos instrucionais sistemáticos.
- **Ensinando estudantes intuitivos:** Os estudantes intuitivos preferem as várias formas de aprendizagem por descobertas. Eles devem possuir uma visão geral, que lhes indique como a matéria está relacionada, para poderem compreender um assunto. Eles podem desenvolver esquemas razoavelmente corretos do conceito ou comparar e contrastar tabelas. Felizmente, estudantes sensitivos podem aprender a fazer o mesmo.
- **Ensinando estudantes racionais:** Os estudantes racionais preferem objetivos e metas claros. Estes estudantes querem saber exatamente qual é o objetivo, tem ações orientadas de modo cognitivo, são afetivos e psicomotoramente precisos. Estes estudantes querem saber o que têm que fazer para aprender o conteúdo.
- **Ensinando estudantes sentimentais:** Os estudantes sentimentais preferem trabalhar em grupos, especialmente nos harmoniosos. Eles apreciam exercícios em pequenos grupos.
- **Ensinando estudantes decididos:** Qualquer coisa que pode ajudar os estudantes decididos a planejar seu trabalho e seguir seu plano será considerado útil. Casos de sucesso, aulas/exposições rápidas e dicas de pesquisa são bem aceitas. O professor pode ajudar os estudantes decididos encorajando-os a melhoria própria.

- **Ensinando estudantes indecisos:** Os estudantes indecisos frequentemente adiam as ações até o último minuto. Eles não são preguiçosos, muito pelo contrário, procuram informações até o último minuto (e às vezes continuam procurando). Assim, o educador deve decompor um projeto ou atividade complexa em uma série de atividades menores e estabelecer prazos finais para cada uma. Isto ajuda a manter os estudantes indecisos no caminho certo e dentro do cronograma.

### **Dimensões dominantes e auxiliares do indicador de tipos de Myers-Briggs**

Quando um estudante ou um instrutor terminam o teste de Myers-Briggs, a pontuação do modelo seleciona, dentre as quatro dimensões discutidas, a dominante, a auxiliar, a terciária e a menos preferida. Muitas pesquisas estão sendo conduzidas na área de ensino e aprendizado levando em conta os estilos de aprendizagem descritos pelas dimensões dominante e auxiliar do ITMB.

Os estudantes ocupam os quatro quadrantes de atitudes (extrovertidos versus introvertidos) e de percepção (sensitivo versus intuitivo). O resultado indica os seguintes estilos:

- **O estudante introvertido e sensitivo:** este estudante pode ser descrito como um pensador. O conhecimento é muito importante para que ele estabeleça “a verdade”. Ele obtém melhores resultados quando trabalha com coisas palpáveis e comprovadas por fatos.
- **O estudante extrovertido e sensitivo:** este estudante pode ser descrito como prático. O conhecimento é importante para ele somente para finalidades práticas. Ele obtém melhores resultados quando lida com informações de fatos concretos e informações práticas.
- **O estudante intuitivo e introvertido:** este estudante pode ser descrito como um pensador e inovador. O conhecimento é importante para ele pela sua própria essência. Eles obtém melhores resultados quando lidam com conceitos, teorias e quando desejam se aprofundar em algum conhecimento. Gostam de trabalhar de forma introspectiva.



- **O estudante intuitivo e extrovertido:** este estudante é chamado de inovador e orientado à ações. Vê o conhecimento como base para mudanças. Ele obtém melhores resultados em situações novas e considera desafios como oportunidades de fazer as coisas acontecerem.

## 2.3 Usando estilos de aprendizagem para ensinar

As pesquisas sobre os estilos de aprendizagem deram aos educadores um novo sentido para realizar mudanças em suas salas de aula. A mudança mais difundida foi a de criar novas formas de abordagem para trabalhos intelectuais. Agrupamentos sociais diferentes, atividades alternativas, e mais projetos complexos foram introduzidos como um esforço para criar oportunidades para os estudantes usarem suas formas potenciais de tratar o material do curso.

Apesar da larga escala de modelos, os conceitos de estilos de aprendizado vêm ganhando crescente atenção dos educadores porque fornecem uma caracterização suficientemente estável para planejar estratégias pedagógicas. Estas estratégias são mais responsáveis em relação as necessidades dos estudantes, e parecem fornecer oportunidades melhores de aprendizado, dando assim, um sentido novo ao ensino alternativo. A seguir estão listadas algumas recomendações gerais dos professores que aplicaram os estilos de aprendizagem em suas salas de aula:

- Os estudantes aprenderão melhor quando forem trabalhadas as preferências em que eles são bem sucedidos;
- Os estudantes serão mais completos e, portanto, melhores quando puderem expandir suas preferências;
- Quando o ensino acomodar várias preferências, mais estudantes serão bem sucedidos;
- Os professores podem construir atividades que incluam preferências específicas e múltiplas de aprendizado;

- Deve-se adicionar alternativas ou, ciclos de aprendizagem que incorporem todos os estilos ou, ainda, realizar tarefas individuais complexas.

## 2.4 Tecnologia para implementação

Viu-se na seção anterior, que fazendo uso de estilos de aprendizagem tem-se características comuns entre estilos diferentes. Como por exemplo, no modelo de Kolb, entre os estilos Acomodador e Divergente há em comum a preferência por experiências concretas. Assim, uma tecnologia de implementação do conteúdo que não ofereça suporte a definição de trechos de conteúdo comuns para estes casos, certamente ocasionaria redundâncias, dificultando tanto a criação quanto a manutenção do conteúdo.

Deste modo, é necessário a criação de uma estrutura que possibilite a definição dos elementos genéricos, elementos de grupos e específicos. Isso pode ser feito através da criação de uma linguagem para essa estruturação ou poderia-se utilizar uma linguagem que permita a extensão de modo conveniente.

Nesse contexto, se insere a linguagem de marcação extensível (XML). Vê-se brevemente a seguir como a linguagem XML [Arm02] é formada, apresentando alguns conceitos importantes para o entendimento do projeto.

A linguagem XML possui uma construção diferente das outras linguagens de marcação, tal como HTML, pois estas linguagens não definem nenhuma estrutura para o armazenamento do conteúdo, apenas armazenam a formatação com que o conteúdo será apresentado. Como exemplo de formatação cita-se cores e fontes utilizadas para a apresentação. Já o XML segue o conceito de definir uma gramática que deve ser obedecida ao se escrever o conteúdo.

Para que isso seja alcançado a linguagem é dividida em três partes, definição de estruturas, dados e regras de transformação, podendo elas estarem em um mesmo arquivo ou em arquivos separados. A seguir é feita uma descrição destas partes e alguns de seus componentes importantes.

### 2.4.1 Definição do tipo do documento (DTD)

A DTD é uma parte da definição da linguagem XML com a qual se pode especificar a estrutura do documento XML. Não é necessário que alguma estrutura seja definida, porém se isso não ocorrer é impossível validar o documento XML. É com a DTD que se especifica quais elementos estarão presentes no documento e quais as estruturas desses elementos, sua ordem de ocorrência, seu conteúdo e atributos. Deve-se perceber, no entanto, que a DTD não define nada sobre a exibição das informações.

As declarações presentes na estruturação de um documento podem ser: os elementos e as suas respectivas estruturas, os atributos de cada elemento, entidades presentes no documento e as notações [Dec00].

A DTD é composta por duas partes: um prólogo e um corpo. O prólogo traz as informações de versão do XML e codificação do texto do corpo. No corpo existem as definições de entidades, elementos e listas de atributos dos elementos. Ve-se na figura 2.3 um trecho de uma DTD exemplo.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!ENTITY % expressaoRegularTexto "#PCDATA|negrito|italico|sublinhado">

<!ELEMENT negrito (#PCDATA|italico|sublinhado)*>
<!ELEMENT italico (#PCDATA|negrito|sublinhado)*>
<!ELEMENT sublinhado (#PCDATA|negrito|italico)*>

<!ELEMENT paragrafo (%expressaoRegularTexto;)>
  <!ATTLIST paragrafo
    alinhamento (centralizado|direita|esquerda) #IMPLIED
  >
<!ELEMENT texto (paragrafo)+>
```

Figura 2.3: Trecho de uma DTD exemplo

Todos os arquivos XML devem ser iniciados por um prólogo da seguinte forma:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

que define a versão da linguagem XML utilizada e a codificação de caracteres do texto. O próximo comando contém a definição de um entidade, repetida abaixo:

```
<!ENTITY % expressaoRegularTexto "#PCDATA|negrito|italico|sublinhado">
```

nele expressãoRegularTexto é o nome da entidade o que se segue é o que a entidade define, que neste caso é uma expressão regular. Para se utilizar a entidade usa-se a notação %nomeDaEntidade; como vê-se no exemplo abaixo:

```
<!ELEMENT paragrafo (%expressaoRegularTexto;)>
```

neste elemento a referência a entidade expressaoRegularTexto será substituída pela expressão regular apresentada em sua definição, o que torna este elemento equivalente ao código abaixo:

```
<!ELEMENT paragrafo (#PCDATA|negrito|italico|sublinhado)>
```

Um elemento, como visto acima, é definido da seguinte forma:

```
<!ELEMENT negrito (#PCDATA|italico|sublinhado)*>
```

em que a declaração “ELEMENT” é seguida pelo nome do elemento e a expressão regular que o define.

Existe ainda a possibilidade de um elemento possuir atributos, o que deve ser definido como segue:

```
<!ELEMENT paragrafo (%expressaoRegularTexto;)>
  <!ATTLIST paragrafo
    alinhamento (centralizado|direita|esquerda) #IMPLIED>
```

os atributos são definidos dentro de uma lista de atributos que segue a seguinte sintaxe: “!ATTLIST” seguida pelo nome do elemento que a possui, o nome do atributo, a expressão regular que o define e mais um qualificador que pode ser “#REQUIRED”, “#IMPLIED” e “#FIXED”. Estes qualificadores referem-se ao atributo ser necessário, opcional ou fixo, respectivamente, sendo que no caso de fixo deve vir um valor para o mesmo entre aspas.

Como já visto, a DTD define uma gramática livre de contexto. Assim, como em toda gramática é necessário que haja um símbolo inicial, que é a raiz da árvore de derivação para a mesma, portanto é preciso que se defina um elemento que sirva como raiz para o documento XML.

O elemento “texto” será o elemento raiz do documento sendo que, como em toda árvore, só é possível existir um elemento raiz em cada documento XML. Ele é composto por um ou mais parágrafos, como visto na expressão regular que descreve seu conteúdo. A notação empregada é a mesma da definição de expressões regulares, ou seja, utiliza-se metasímbolos, que são os elementos que foram definidos na DTD, entremeados por caracteres que especificam o número de ocorrências dos mesmos. Esses caracteres podem ser:

- Quantificador \* que representa a ocorrência de 0 ou mais vezes;
- Quantificador + que representa a ocorrência de 1 ou mais vezes;
- Quantificador ? representa a ocorrência de 0 ou 1 vez;

## 2.4.2 O documento XML

É no documento XML que as informações estão armazenadas, e este documento deve ser escrito de acordo com o que foi especificado na DTD. Caso isso não ocorra o documento XML não será válido. Porém o documento XML pode ser escrito sem a presença de uma DTD e neste caso ele somente será inválido se houver quebra da hierarquia das marcações (Tags), como exemplo: <negrito><italico> forma invalida </negrito></italico>, em que existe uma quebra de hierarquia pois a marcação **negrito** foi fechada antes da marcação *italico*. A forma correta seria <negrito><italico> certo </italico></negrito>. Deve-se notar ainda que mesmo com a presença da DTD é necessário que não haja quebra da hierarquia.

O documento XML também possui um prólogo igual ao da DTD, no qual são especificados a versão da linguagem XML, a codificação do texto e se o documento XML é ou não *standalone*, ou seja, se o documento XML possui DTD interna (*standalone*= “yes”) ou se a DTD é externa (*standalone*= “no”).

Na parte seguinte do documento XML encontra-se o DOCTYPE que é a especificação da DTD para o documento da qual se segue o nome do *elemento-raiz*, que é um elemento que deve ser único no documento XML, pois ele segue a estrutura de árvore e é assim que todos os outros elementos podem ser alcançados; segue-se ainda a especificação da estrutura da DTD se o documento for *standalone="yes"*, caso contrário deverá existir a palavra SYSTEM seguida do nome do arquivo que contém a especificação da DTD.

Finalmente encontram-se os elementos que representam a informação. Vê-se no XML, assim como na DTD, que não há presença de informações de como os dados devem ser exibidos. Na figura 2.4, vê-se um exemplo de um documento XML válido para a DTD especificada na figura 2.3 da sub-seção 2.4.1.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE texto SYSTEM "texto.dtd">
<texto>
  <paragrafo>Este é um paragrafo comum</paragrafo>
  <paragrafo>Aqui veremos um <negrito>texto em negrito</negrito>
    e um <italico> texto em italico</italico>,
    tambem podemos ter <sublinhado>texto sublinhado.</sublinhado>
  </paragrafo>
  <paragrafo>A
    <negrito>
      <italico><sublinhado>combinação</sublinhado></italico>
    </negrito>
    dos <sublinhado><italico>três</italico></sublinhado>
    itens ou a <italico><negrito>combinação</negrito></italico>
    <sublinhado>dois a dois</sublinhado>
    também é <sublinhado><negrito>possível</negrito></sublinhado>
  </paragrafo>
  <paragrafo alinhamento="centralizado">Este é um parágrafo centralizado
  </paragrafo>
</texto>
```

Figura 2.4: Trecho de XML exemplo

### 2.4.3 Linguagem extensível de folha de estilo (XSL)

O processo de filtragem das informações que serão apresentadas, como também as transformações necessárias para o controle do processo de apresentação das informações são efetuadas através da linguagem extensível de formatação de páginas (XSL).

Existe um outro modo de apresentar as informações, contidas no XML, utilizando-se folhas de estilo em cascata (CSS) que atua percorrendo o documento e aplicando os formatos definidos sem a possibilidade de incluir qualquer alteração [Dec00].

Já o XSL processa o documento de maneira programável, ou seja, é possível pela aplicação de regras formatar as informações contidas nos elementos e nos atributos presentes no documento XML, bem como incluir informações novas, ignorar parte das informações e filtrá-las baseando-se em desvios de fluxo de acordo com testes aplicados sobre as informações.

Nota-se assim, que o XSL é bem mais poderoso que o CSS e há uma forte tendência a utilizá-lo para a seleção das informações que serão exibidas a cada um dos estilos.

O arquivo XSL também é um arquivo XML, e portanto deve possuir um prólogo igual ao da DTD e também deve ser bem formado, possuindo um único elemento-raiz e suas marcações devem também obedecer a hierarquia das marcações.

A definição de regras é realizada pelo comando `<xsl:template match="nomeDoElemento">`, o qual especifica o que deve ser feito quando este elemento for encontrado.

Como vê-se no XSL da figura 2.5, há um prólogo como o da DTD seguido por um complemento próprio do XSL que especifica o seu espaço de nomes como especificado por `http://www.w3c.org/199/xsl/Transform` e também a versão da XSL.

Complementando ainda o prólogo tem-se a definição do método de formatação da saída que será utilizado no momento da transformação. O tipo especificado deve ser um tipo *mime* como o que utilizou-se acima. Este método de formatação, dever ser empregado para que no momento da transformação a conversão de determinados elementos XML seja apropriada para a exibição. Como exemplo pode-se citar os elementos XML vazios, que são definidos por `<nomeDoElemento atributo1="valor1" ... atributoN="valorN" />`, que ao serem transformados por exemplo em HTML passam a não possuir a barra de fechamento.

Segue-se definindo a regra para o elemento-raiz (“/”), na qual inclui-se as marcações de início e corpo de um documento HTML. Através do comando `<xsl:apply-templates/>` processa-se os elementos filhos recursivamente aplicando as outras regras definidas no XSL e finalmente fecham-se as marcações de corpo e início do documento HTML.

Na definição da regra para o elemento `paragrafo` tem-se os comandos `<xsl:element name="p">` e `<xsl:attribute name="align">`, que criam respectivamente um elemento de nome “p” e um atributo de nome “align”. Dentro do elemento “p” que é criado testa-se com o comando `<xsl:if test="@alinhamento">` a existência do atributo alinhamento para o elemento `paragrafo`, caso exista, utiliza-se o comando para criação de atributo citado acima para criar um atributo de nome “align” que na sequência receberá o valor do atributo alinhamento pertencente a `paragrafo` obtido através do comando `<xsl:value-of select="@alinhamento"/>`. Na sequência utiliza-se novamente o comando “apply-templates”, visto anteriormente, para processar os elementos filhos deste elemento.

Os elementos `negrito`, `italico` e `sublinhado` seguem o mesmo padrão do elemento `paragrafo`, exceto pela verificação de atributos.

Existe vários outros comandos XSL como: `copy-of`, `if`, `choose`, `for-each`, etc, dos quais faz-se uso em situações específicas dependendo do modo como for necessário transformar o documento XML.

## 2.5 Considerações finais

Para ser eficiente no objetivo de ensinar faz-se necessário a utilização de metodologias de ensino que considerem o estilo pessoal de aprender do estudante. Assim, buscou-se na literatura modelos que fossem adequados a proposta de uma ferramenta computacional para o auxílio ao ensino.

Para colocar as abordagens propostas pelos modelos presentes na literatura em prática buscou-se tecnologias que fornecessem melhor suporte para os diferentes estilos mapeados por estes modelos, de modo a evitar a redundância de informações, visto que notou-se que há parcelas de informações que são comuns entre os estilos definidos pelo mesmo modelo.



```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3c.org/199/xsl/Transform"
  version="1.0">
  <xsl:output method="text/html"/>
  <xsl:template match="/">
    <html>
      <body>
        <xsl:apply-templates/>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="negrito">
    <xsl:element name="b">
      <xsl:apply-templates/>
    </xsl:element>
  </xsl:template>
  <xsl:template match="italico">
    <xsl:element name="i">
      <xsl:apply-templates/>
    </xsl:element>
  </xsl:template>
  <xsl:template match="sublinhado">
    <xsl:element name="u">
      <xsl:apply-templates/>
    </xsl:element>
  </xsl:template>
  <xsl:template match="paragrafo">
    <xsl:element name="p">
      <xsl:if test="@alinhamento">
        <xsl:attribute name="align">
          <xsl:value-of select="@alinhamento"/>
        </xsl:attribute>
      </xsl:if>
      <xsl:apply-templates/>
    </xsl:element>
  </xsl:template>
</xsl:stylesheet>

```

Figura 2.5: Trecho de código XSL para o exemplo

# Capítulo 3

## Aplicando estilos de aprendizagem em ensino auxiliado por computador

Neste capítulo aborda-se o modo como aplicar as diretrizes dos modelos de estilos de aprendizagem no ensino, especialmente no ensino auxiliado por computador. Identifica-se ainda qual o modelo mais adequado para a utilização neste projeto e, em seguida, especifica-se estratégias de abordagem a serem implementadas baseando-se no modelo escolhido e nos objetivos do projeto.

### 3.1 Definição do modelo de aprendizagem

O reconhecimento de que estudantes tem estilos de aprendizagem diferentes pode ser útil para ajudar o instrutor criar um ambiente centrado no estudante.

Em um ambiente centrado no estudante o instrutor se assegura do sucesso do estudante seguindo as regras abaixo:

1. Preparar as instruções de modo a suprir a necessidade dos vários estilos de aprendizagem, usando uma combinação de experiências, reflexão, conceitualização, e experimentação. A preparação das instruções utiliza uma grande variedade de abordagens, incluindo som, música, figuras, animações e experiências, tanto quanto leitura;
2. Preparar um programa de ensino que enfatize habilidades pessoais, genéricas, inter-

personais, e sociais, tanto quanto as habilidades do professor.

A tecnologia fornece novas potencialidades para reconstruir ambientes de aprendizado em torno dos estilos de aprendizado específicos [Ct102].

Percebe-se nesse estudo sobre os modelos de aprendizado que o modelo ideal a ser adotado é o Modelo de Camadas de Curry [Cur87][Cur90][Ing99], que engloba todos os aspectos do aprendizado e todos (ou quase todos) os modelos criados até então. Porém como pode-se notar, este modelo somente classifica os modelos de aprendizagem propondo um direcionamento ao educador em suas abordagens de ensino, não sendo um modelo de fato.

A implementação de todo o modelo de Curry é algo inviável pelos aspectos que não podem ser controlados pela ferramenta ou que são altamente relacionados com características intrínsecas do indivíduo. Dentre estes aspectos pode-se notar que a camada mais externa do modelo, que se refere ao ambiente, pode ser ajustada facilmente pelo estudante. A camada mais interna, por sua vez, trata dos aspectos mais intrínsecos da personalidade, ou seja, ela se relaciona com a forma que o estudante desenvolve suas opiniões pessoais [Mye62], sendo algo muito complexo de implementar devido a grande variedade de possibilidades que surgem quando combinadas todas as características básicas de personalidade.

A camada intermediária do modelo, referente a forma que as pessoas processam e assimilam as informações, é muito mais importante à ferramenta do que as necessidades ambientais, de comportamento, ou sociais. O Modelo de Kolb [Kol76], que identifica as características de aprendizagem que estão presentes nesta camada, é ainda um modelo simples de implementar, suprimindo todas as necessidades de ensino da ferramenta de modo direto. Além disso não é necessário o emprego de outros métodos de ensino, pois, o material será apresentado em um modo com o qual o estudante se identifica e sente-se seguro para aprender [Ct102].

Portanto, pode-se concluir que o melhor modelo de estilos de aprendizagem, na ferramenta para auxílio ao ensino de sistemas de tempo-real, é o de Kolb. Assim, apresenta-se a especificação das abordagens que serão empregadas pela ferramenta, em cada um dos estilos de aprendizagem do modelo de Kolb, baseando-se para isso nas características apre-

sentadas para as várias dimensões do inventário de Kolb (concreto, reflexivo, abstrato, ativo).

## **3.2 Planejamento do método de abordagem**

Para que a ferramenta alcance seus objetivos, otimizando o aprendizado do estudante, definiu-se a melhor forma de abordagem para cada estilo pessoal de aprendizado. A partir da escolha do modelo de Kolb, especifica-se com base nos estilos propostos por este modelo as estratégias de abordagens citadas abaixo.

### **3.2.1 Divergentes - *Reflexivo/Concreto***

Para este estilo de aprendizado o instrutor deve atuar motivando o estudante. Motivá-los a aprender o material e mostrar-lhes como este se enquadra com o que esperam. Deve-se mostrar ao aluno a utilidade do que ele aprendeu ou vai aprender, sempre com o intuito de entusiasmar e motivá-lo.

Os alunos que possuem este estilo de aprendizado querem interagir pessoalmente com o instrutor e serem reconhecidos como únicos. O educador precisa entusiasmar o crescimento pessoal do estudante. Além disso, aprendem bem através de discussões e costumam ver as coisas de forma diferente nas discussões em grupos.

Segundo estas características, o material deve ser apresentado juntamente com questões que façam o aluno parar para pensar. O material deve apresentar, de acordo com as decisões (respostas) do aluno, as justificativas da resposta apresentada. Em um grupo de discussão, esta poderia ser a explicação de outras pessoas.

Como esse aluno gosta de ver as coisas de perspectivas diferentes é importante que o material apresente estas perspectivas, fazendo-o pensar melhor sobre o assunto. Assim, nos testes, pode-se apresentar afirmações seguidas de uma questão e várias alternativas, justificando as afirmações de modo que, para escolher uma resposta, ele tenha que avaliar os vários pontos de vista da questão.

Devido ao fato de alunos deste estilo gostarem de trabalhos em grupos, é necessário que haja uma maneira deles interagirem com outros estudantes do mesmo estilo durante

o estudo e principalmente durante a resolução de problemas. E desta forma, ao decidir em grupo quem possui a razão sobre determinado assunto, ele estará observando todas as perspectivas, solidificando assim seu conhecimento.

### **3.2.2 Assimiladores - *Reflexivo/Abstrato***

Assimiladores aprendem da forma tradicional de ensino, em que a matéria deve ser bem organizada e detalhada. O instrutor deve agir como um especialista, sendo autoritário e fornecendo informações. Grupos devem ser evitados. O educador deve ser cauteloso para a forma de abordagem não fugir ao estilo de aprendizado do aluno, pois como estes alunos seguem procedimentos, eles podem executar as tarefas mesmo que estas não sejam ideais para seu estilo, tornando difícil identificar abordagens incorretas.

O material deve ser apresentado de forma seqüencial, completa e bem organizada, e deve trazer vários exemplos e justificativas para os mesmos, pois representam o ponto de vista de especialistas. Exercícios podem ser apresentados ao final de cada sessão, mas estes exercícios não devem ser extensos, complexos, ou muito práticos, devem apenas ajudar a assimilar o conteúdo. Exercícios tipo teste serão bem aceitos aqui, pois, este tipo de aluno gosta de ler e observar, processando a informação pensando sobre ela e não colocando-a em prática.

Ao final dos capítulos podem ser apresentados exercícios mais complexos, que façam o aluno refletir sobre o que ele acabou de aprender.

### **3.2.3 Convergentes - *Abstrato/Ativo***

Os convergentes precisam que o educador trabalhe como um técnico, propondo trabalhos práticos e exigindo um retorno ou resposta do aluno. Os estudantes precisam ter um papel mais ativo, pois não gostam de ler muito, preferindo laboratórios onde possam aprender por tentativa e erro, aprendendo ao fazer. Acreditam que grupos são desperdício de tempo.

O material deve ser apresentado de forma resumida com apenas o conteúdo que é essencial, porém deve ser possível ao aluno buscar informações complementares em um

material mais completo, que não será utilizado como texto principal e sim como um texto de consulta.

Todos os conceitos e lições apresentados devem ser seguidos de exercícios que façam com que o aluno trabalhe de forma prática. Deve-se evitar exercícios repetitivos, pois os alunos deste estilo não são a favor de refazer trabalhos que já foram realizados.

### **3.2.4 Acomodadores - *Ativo/Concreto***

Para trabalhar com este estilo o instrutor deve permitir que o aluno desenvolva o auto-aprendizado. O instrutor deve avaliar, orientar, e incentivar a auto-descoberta.

Estudantes desse estilo gostam de grupos e de resolver problemas, tem sua própria ordem e horário para as tarefas e não gostam de muitos procedimentos e regras. Costumam correr riscos e aprendem a partir dos erros.

Desta forma a abordagem para este grupo deve ser com um material que não siga uma seqüência fixa, apesar de poder ser sugerida. Assim os alunos podem estudar os módulos de acordo com sua própria vontade, até que o material seja totalmente completado. Como alunos deste estilo gostam de grupos, pode-se criar grupos de discussão fictícios que discutam a matéria, tal como foi sugerido ao estilo divergente. Porém, como este estilo processa a informação de forma diferente dos divergentes, a forma das discussões deve ser diferentes. Além disso deve haver exercícios práticos, como foi sugerido ao tipo convergente. Os exercícios para este tipo de aluno devem mesclar grupos de discussões com trabalhos individuais, devendo ser os mais práticos e reais possíveis.

### **3.2.5 Abordagens para os grupos**

Como os estilos possuem características em comum, (por exemplo o divergente compartilha a preferência por observação reflexiva com o assimilador e a preferência por experimentação concreta com o acomodador), é preciso apresentar partes do material de forma igual a dois ou mais estilos. Nesse caso é interessante fazer com que características comuns a estilos adjacentes (ver figura 2.1) apareçam como um novo grupo de abordagem. Os grupos assim definidos, com as respectivas abordagens, são:

Módulo Servidor	Módulo Cliente
TalkWithClient	KolbsTest
ServerUserData	Login
TutorServer	UserData
Cache	ReceiveFromServer
ManipuladorXMLXSL	Tutor
ArvoreDeConteúdo	GerenciadorConteudo

Figura 3.1: Componentes dos módulos servidor e cliente

- *Experiência concreta:* Apresentar situações mais próximas a realidade, com exemplos direcionados que podem ser utilizados para construir novos conhecimentos.
- *Observação reflexiva:* Apresentar conceitos de forma clara e estruturada.
- *Conceitualização abstrata:* Estudos de caso fortemente apoiados na teoria e exercícios de reflexão.
- *Experimentação ativa:* Os exercícios aplicados a este grupo devem ser práticos.

### 3.3 Implementando a ferramenta de auxílio ao ensino

Nesta seção descreve-se como aplicou-se o modelo de Kolb na construção do módulo RTTutor (Real Time Tutor), com o objetivo de fazer uso de estilos de aprendizagem no ensino de sistemas de tempo-real.

O módulo RTTutor é composto por dois outros módulos, que são o servidor e o cliente. O servidor armazena e recupera os dados dos usuários e do conteúdo do curso. Já o cliente cuida da apresentação do curso ao usuário. Na figura 3.1 lista-se cada um dos principais componentes destes módulos, descritos detalhadamente a seguir, e cujas conexões podem ser vistas na figura 3.2.

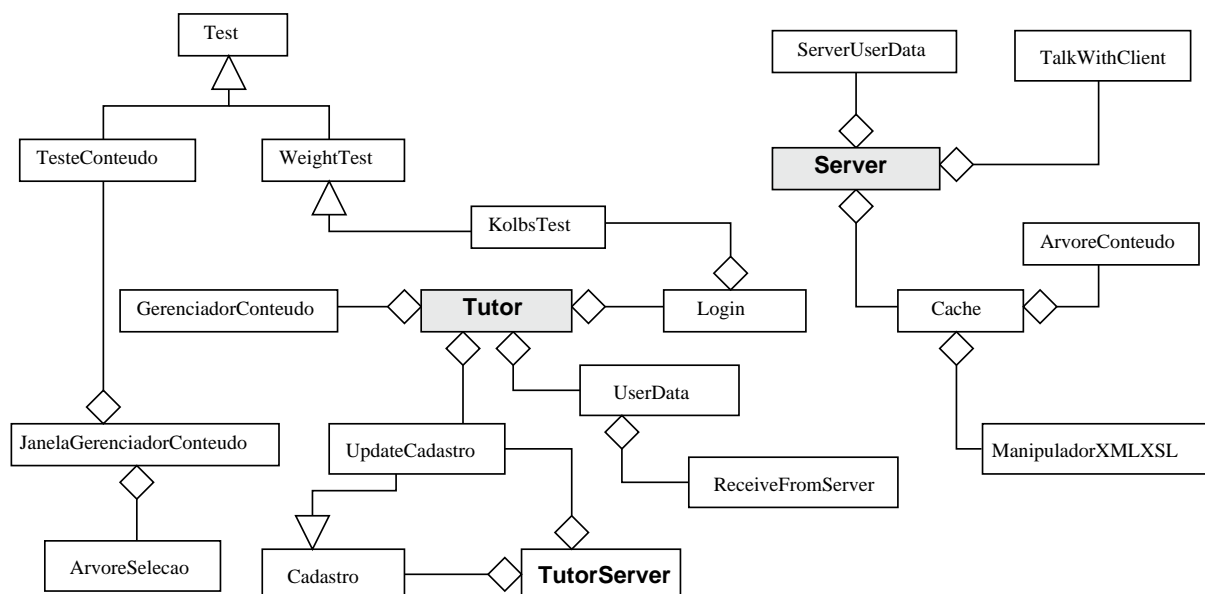


Figura 3.2: Esquema do RTTutor

### 3.3.1 Componentes do servidor

O servidor é responsável por recuperar informações referentes ao aluno e ao conteúdo do curso, enviando-os quando forem requisitados. O conteúdo do curso é enviado por partes (seções) de acordo com as requisições do aluno efetuadas ao clicar na “árvore de seleção” da janela do gerenciador de conteúdo. (ver figura 3.4 na página 44).

Para o conteúdo ser utilizado pelo gerenciador de conteúdo ele deve estar no formato HTML, portanto o servidor tem como uma de suas tarefas efetuar a transformação e a seleção do que será visualizado pelo aluno, a partir de um documento XML, em um documento HTML.

Este processo, por ser um servidor, é caracterizado pela escuta de requisições através da rede, efetuadas pelo componente “UserData” atendendo-as e instanciando classes (“threads”) “TalkWithClient” que assumirão as conexões. A partir da instanciação desta classe o servidor volta a escutar por novas requisições.

- **TalkWithClient:** Encarrega-se de manter a comunicação entre o cliente e prover os serviços oferecidos. As requisições provenientes do cliente à esta classe partem de um objeto da classe “UserData” e toda resposta enviada ao cliente na verdade é realizada através da comunicação desta classe com uma instância da classe



“ReceiveFromServer”.

Mensagens ou ordens enviadas pelo servidor ao cliente sem que tenham sido requisitadas devem ser efetuadas através de chamadas de funções desta classe as quais se encarregarão do envio. A classe “ReceiveFromServer” ao receber este tipo de requisição gera eventos que podem ser capturados e tratados.

- **ServerUserData:** É responsável pela recuperação, armazenamento e manipulação dos dados dos usuários, que são dados pessoais, login, senha e o andamento do curso. Fornecendo assim, uma camada de abstração e isolamento entre os dados e as funcionalidades do sistema.
- **TutorServer:** É uma interface gráfica para o controle e configuração do servidor. Sendo responsável também por prover uma interface para o cadastramento dos usuários no sistema.

Este componente atua como um cliente privilegiado, efetuando acessos ao servidor utilizando a mesma interface de comunicação que os clientes, porém com restrições que garantem que clientes comuns não alterem dados no servidor.

- **Cache:** Devido ao fato da busca de conteúdo no servidor ser executada muitas vezes, quando o servidor estiver com vários alunos conectados, há a possibilidade de uma sobrecarga do servidor. Assim, para evitar esta sobrecarga, faz-se necessário a construção de uma estrutura de *cache*, onde as transformações que já tenham sido efetuadas sejam reaproveitadas, evitando assim um novo processamento e consequentemente a sobrecarga do servidor disponibilizando a informação de maneira mais rápida.

A classe **Cache** é basicamente composta por uma estrutura de tabela, que possui como índices o identificador do componente (seção) e os quatro estilos de Kolb. Os campos desta tabela devem ser do tipo “String”, onde pode-se armazenar o conteúdo HTML (gerado pela transformação do XML utilizando o identificador do componente e o estilo do usuário) ou o valor “NULL” que representa que não houve a transformação até o momento para este par identificador/estilo.

- **ManipuladorXMLXSL:** Quando ocorre uma consulta a esta estrutura de **cache** e não se encontra o conteúdo já preparado, faz-se necessário um pedido de processamento deste conteúdo. Para isso se efetua um pedido a um objeto da classe “ManipuladorXMLXSL” que é responsável por efetuar a transformação apropriada do conteúdo XML para o formato HTML, retornando a um objeto da classe **Cache** o HTML requisitado. A instância de **Cache** por sua vez armazena o HTML em sua tabela e responde o pedido inicial. Este componente também recupera as questões e respostas dos testes de conteúdo.

Para realizar essa transformação, buscou-se bibliotecas que fossem capazes tanto de manipular o documento XML como promover a filtragem a partir deste, formatando ainda a saída de forma conveniente.

Estas bibliotecas foram encontradas na versão mais recente da linguagem Java (SDK1.4.0). Entre as bibliotecas que podem manipular o conteúdo XML estão a biblioteca DOM e a SAX. Já para filtrar este conteúdo gerando uma saída formatada encontrou-se a biblioteca XSLT.

A biblioteca SAX apenas gera eventos, durante a leitura do documento XML, os quais devem ser tratados neste momento. Em contrapartida, a biblioteca DOM gera uma árvore que representa o documento como um todo. Assim, devido a característica do modo de operação do tutor utilizou-se a biblioteca DOM, visto que o processo de leitura deve ser efetuado somente uma vez gerando um objeto com o qual pode-se trabalhar durante o período de utilização da ferramenta.

Empregou-se a biblioteca XSLT (*XSL Transformation*), para efetuar a filtragem e transformação dos trechos do documento XML os quais o usuário seleciona a partir da árvore de seleção.

- **Árvore de conteúdo:** No gerenciador de conteúdo têm-se a “árvore de seleção”, que é um componente visual, semelhante às árvores de diretórios presentes nos gerenciadores de arquivos existentes nos sistemas operacionais atuais. Esta é construída a partir da árvore de conteúdo.

A árvore de conteúdo é formada por nós que possuem a estrutura de tópicos do

conteúdo (conteúdo, capítulo e seção) e os identificadores relativos a estes tópicos. Os identificadores são os id's (atributos) associados aos tópicos no documento XML e são utilizados para recuperar seu conteúdo.

Uma instancia da árvore de conteúdo para cada um dos estilos é inserida no objeto da classe `Cache` e é transferida para o cliente, quando um gerenciador de conteúdo é instanciado para o estilo do usuário deste cliente.

Como as árvores de conteúdo são criadas com base no conteúdo XML, a responsabilidade pela sua criação foi delegada a um objeto da classe `ManipuladorXMLXSL`.

Quando ocorre uma alteração no conteúdo XML, devido a um processo de autoria, o autor (professor) deve disparar no servidor um processo de atualização. Neste processo o servidor efetua uma chamada ao objeto da classe `Cache` e este por sua vez esvazia sua tabela e recria as árvores de conteúdo, após esta etapa o servidor envia um aviso a todas as *threads* da classe “`TalkWithClient`” para efetuar a atualização.

O processo de atualização nos objetos da classe “`TalkWithClient`” é constituído pelo envio de uma ordem de atualização ao cliente, seguido da árvore de conteúdo do estilo apropriado. Esta ordem de atualização chega ao objeto da classe “`ReceiveFromServer`” no cliente e gera um evento que é capturado pelo gerenciador de conteúdo efetuando, assim, a atualização da árvore de seleção. Deste momento em diante todas as novas seleções de conteúdo, efetuadas pelo aluno, tem a capacidade de recuperar o conteúdo atualizado.

Nota-se que um controle de concorrência para a classe `Cache` não é necessário, porém, pode acontecer de dois ou mais usuários do mesmo estilo requisitarem o mesmo conteúdo, que por sua vez pode não se encontrar na instância de `Cache`, assim, vários processos de transformação deste conteúdo podem ser disparados. A implementação de um processo de sincronismo evita a sobrecarga do servidor que pode ocorrer neste caso em particular.

### **3.3.2 Componentes do cliente**

O cliente é responsável por requisitar a validação do usuário no servidor, recuperando os dados sobre o andamento do curso os quais serão utilizados pelo gerenciador de conteúdo. Ele é responsável também por recuperar e exibir o conteúdo do curso segundo as

necessidades e o estilo do usuário.

Observa-se a seguir uma breve descrição dos componentes do cliente:

- **KolbsTest** Este componente (figura 3.3) é uma implementação do inventário de estilos de aprendizagem de Kolb, como visto no anexo A.

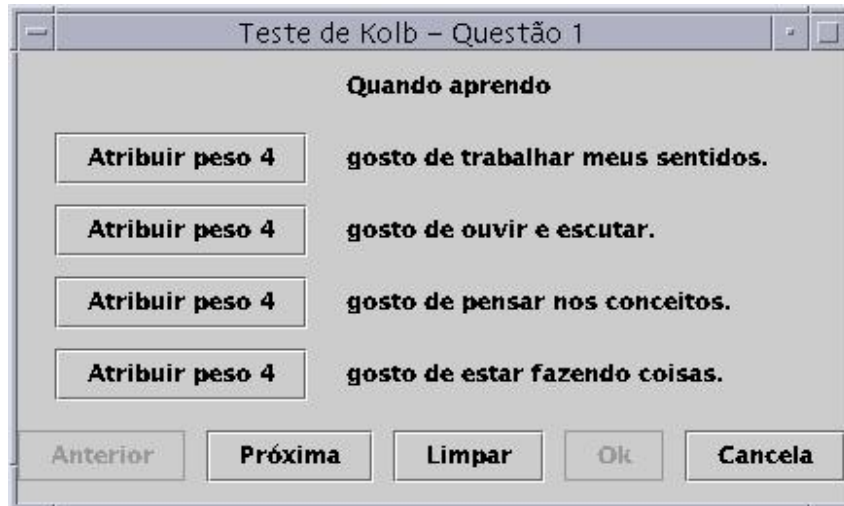


Figura 3.3: Questão do teste de Kolb

A implementação realizou-se com base em uma característica da programação orientada à objetos denominada hierarquia de classes. Deste modo implementou-se uma classe “Test” (figura 3.2) que modela um teste de múltiplas alternativas e através de especialização criou-se uma classe teste de peso (WeightTest) que modela um teste no qual se atribui pesos a cada alternativa associada a uma questão. No final da hierarquia implementou-se o teste de Kolb (KolbsTest), deste modo simplificou-se a implementação de outros testes que foram utilizados, pois grande parte do código pôde ser reutilizado através de herança. Esta é uma característica que, de acordo com a Engenharia de Software [Pre95], faz parte de um bom projeto.

Nota-se na figura 3.3 que inicialmente todos os botões apresentam o mesmo valor de peso, porém quando há uma escolha pelo usuário o botão escolhido fica com o peso de valor 4 associado e os demais passam ter a possibilidade de assumir o peso de valor 3, realizada nova escolha a próxima escolha possível é o peso de valor 2 e

quando este é escolhido o próximo botão assume o peso de valor 1, sendo possível então prosseguir para a próxima questão.

- **Login** É uma forma de autenticar o usuário do sistema no momento de sua utilização. É invocada pelo “Tutor” e devolve a este os dados que carregou do servidor caso o usuário seja autenticado. É responsável ainda por chamar o teste de Kolb (KolbsTest) quando o usuário for autenticado pela primeira vez no sistema.
- **UserData** Este componente é a interface de comunicação de todo o Cliente com o servidor, sendo o responsável por conectar-se ao servidor efetuar os pedidos de informações tais como os dados do usuários, os conteúdos e testes que serão apresentados entre outras funções de controle.

Ao efetuar uma requisição ao servidor este componente passa a escutar através de um “pipe” (canal) com o componente “ReceiveFromServer” a resposta.

- **ReceiveFromServer** Objetos desta classe recebem as respostas e requisições do servidor e se encarregam de encaminhar as respostas a um objeto da classe “UserData”, através de um canal de comunicação entre estes objetos.

Requisições enviadas pelo servidor, sem que tenha havido uma requisição prévia, devem ser processadas gerando eventos que permitam que as classes que estão utilizando o objeto da classe “ReceiveFromServer” capturem e tratem estes eventos.

Vê-se que este componente não necessita ser usado diretamente por outros componentes do cliente que efetuam requisições ao servidor, para este fim usa-se o componente “UserData”. O componente “ReceiveFromServer” somente será utilizado diretamente quando houver necessidade de receber informações do servidor sem que estas tenham sido requisitadas.

- **Tutor** É o componente principal do sistema, a partir do qual é invocada tanto a entrada de login, como o gerenciador de conteúdo, entre outros.
- **Gerenciador de conteúdo** possui uma janela que se divide em duas partes, na parte esquerda da janela vê-se a árvore de seleção, já no lado direito encontra-se a área de apresentação do conteúdo, como vê-se na figura 3.4.

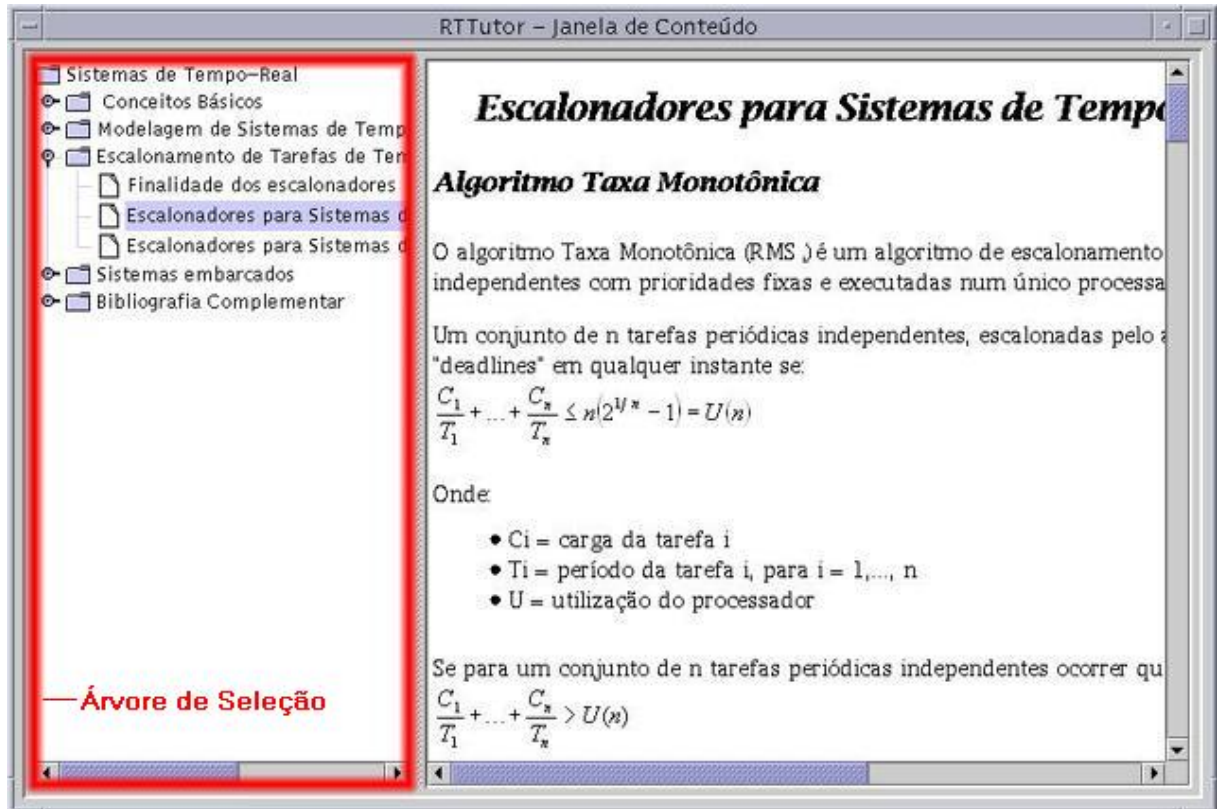


Figura 3.4: Gerenciador de Conteúdo

O componente utilizado para a apresentação do conteúdo não suporta a exibição de conteúdo XML. Deste modo, criou-se um componente intermediário que é responsável por converter o conteúdo escrito em XML para o formato HTML, filtrando-o, utilizando o XSL, de acordo com o estilo do aluno. A árvore de seleção foi idealizada seguindo o mesmo raciocínio utilizado nos gerenciadores de arquivo utilizados nos ambientes gráficos atuais. Assim, quando o usuário seleciona um dos ramos da árvore envia-se uma pesquisa do componente selecionado para o manipulador (ManipuladorXMLXSL) que responde com o texto HTML correspondente ao componente selecionado.

Observa-se assim que o gerenciador de conteúdo é responsável por todas as ações que envolvem a apresentação do conteúdo e a aplicação de testes.

### 3.3.3 Funcionamento do RTTutor

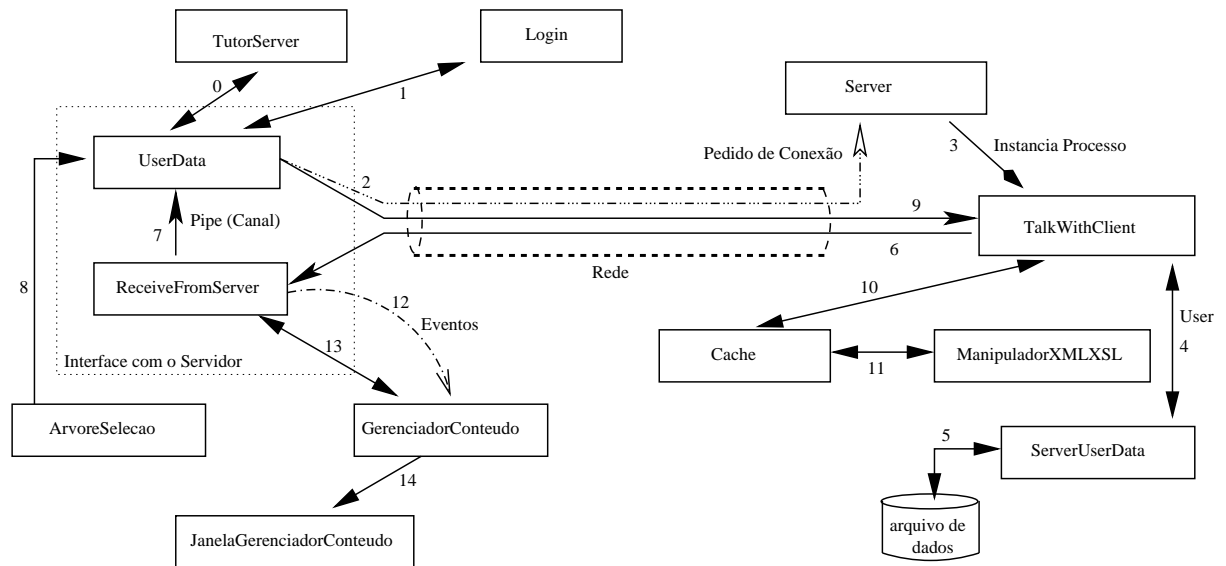


Figura 3.5: Funcionamento do RTTutor

Para o funcionamento do módulo RTTutor (figura 3.5) deve-se primeiramente configurar o servidor através da edição de dois arquivos, o arquivo *rttutor.conf*, no qual indica-se a localização de um segundo arquivo o *server.conf*, em que se insere as configurações do servidor, tais como a porta em que o mesmo atende, o local onde estão os dados dos usuários e o conteúdo do curso e senha de acesso ao servidor.

Após estas configurações inicia-se o servidor, sendo que este processo pode ser realizado de duas formas, a partir de um comando executado em um terminal, ou através da interface gráfica **TutorServer**. Nesta interface gráfica também é possível efetuar as configurações citadas anteriormente.

A interface **TutorServer** faz uso de uma instância da classe **UserData** (0)<sup>1</sup>, para efetuar acessos privilegiados ao servidor - mediante o fornecimento da senha de acesso ao servidor - tais como cadastro e exclusão de usuários e alterações de dados dos mesmos.

Iniciado o servidor é possível que os clientes efetuem conexões e iniciem o uso da ferramenta de ensino. A partir da janela de login é possível configurar o endereço e a

<sup>1</sup>Utiliza-se nesta subseção a notação de números entre parênteses referenciando os arcos da figura 3.5

porta em que o servidor atende. Em seguida com a entrada de nome de usuário e senha instancia-se um objeto da classe `UserData` (1) que será utilizado para efetuar o pedido de conexão ao servidor (2).

Ao receber o pedido de conexão o servidor instancia um processo (thread) da classe `TalkWithClient` (3), que por sua vez efetua uma chamada a um objeto da classe `ServerUserData` (4), para recuperação dos dados do cliente (5). Esta *thread* também assumirá a negociação de serviços com o cliente. Nota-se ainda, que para cada cliente haverá um processo da classe `TalkWithClient`.

No cliente, uma instância da classe `ReceiveFromServer` recebe as respostas do objeto da classe `TalkWithClient` (6) para as requisições efetuadas pela instância da classe `UserData` (9). Este objeto realiza uma análise das respostas recebidas escrevendo-as em um canal dedicado (*pipe*) com o objeto da classe `UserData` (7), quando houver a necessidade de bloqueio da execução do Tutor para espera de resposta do servidor (por exemplo autenticação do usuário). Quando não houver a necessidade de bloqueio do Tutor o objeto da classe `ReceiveFromServer` gera eventos (12), que serão capturados pelas classes registradas como receptoras, as quais, por sua vez efetuarão o processamento dos mesmos.

A interação com o usuário durante a apresentação do curso é realizada através da árvore de seleção a qual gera requisições ao servidor utilizando o objeto da classe `UserData` (8). As respostas a estas requisições geram eventos que devem ser capturados pelo objeto da classe `GerenciadorConteudo`. Estes eventos notificam ao objeto da classe `GerenciadorConteudo` que o servidor enviou informações que devem ser tratadas por ele, sendo necessário que este efetue a recuperação destas informações do objeto da classe `ReceiveFromServer` (13). Estas informações serão repassadas ao objeto da classe `JanelaGerenciadorConteudo` (14) que se encarrega de exibí-las ao usuário.

### 3.4 Estruturação do conteúdo do curso

Tendo em vista que os diversos estilos devem possuir conteúdos adequados a suas necessidades, buscou-se uma tecnologia que pudesse atender a essa personalização sem que houvesse redundâncias de informações - provenientes de características comuns entre os



estilos (grupos). A eliminação de redundâncias é proveitosa pois a necessidade de geração de conteúdo e de sua manutenção fica multiplicada por quatro com a sua existência. O conteúdo do curso é estruturado em elementos genéricos, específicos e mistos.

**Elementos genéricos:** Representam partes do conteúdo que serão apresentadas a todos os estilos.

**Elementos específicos:** São partes do conteúdo que somente serão apresentadas para o estilo em questão, ou seja, um elemento que possui a propriedade **assimilador** será exibido somente para os alunos que são do estilo assimilador.

**Elementos mistos:** Apresentam conteúdo comum das regiões de fronteira entre os diversos estilos. Por exemplo, o “assimilador” e o “divergente” possuem em comum a característica de serem ambos reflexivos, enquanto que “assimilador” e “convergente” têm em comum a preferência por conceitualização abstrata.

### 3.4.1 Implementação do conteúdo

Um ponto crucial na especificação do RTTutor foi a escolha do modo como deve-se efetuar a estruturação e armazenamento das informações referentes ao curso que será apresentado ao aluno.

Nesta fase examinou-se diversas abordagens, com o intuito de encontrar a abordagem que seria mais eficiente, tanto no que diz respeito a evitar redundâncias quanto a simplificar o trabalho de autoria do conteúdo do curso.

Assim, analisou-se basicamente três tipos de abordagens, vistas sucintamente a seguir.

- **Arquivos HTML** Nesta abordagem a idéia principal é a escrita de quatro conteúdos, ou seja, escrever em um arquivo HTML (`assimilador.html`), o conteúdo do curso para o estilo de aprendizado “assimilador”, e assim também para os demais estilos.

Esta é uma abordagem muito simples, permitindo que o curso possa ser escrito facilmente, inclusive utilizando-se de ferramentas de edição HTML existentes no

mercado, tais como o Netscape Composer. Isso evita a exigência de conhecimento técnico do autor no que se refere a linguagem.

Porém, este tipo de abordagem apresenta um sério problema quanto a redundância de informações. Ao escrever-se o conteúdo do curso para estilos de aprendizagem, partes do conteúdo estão presentes em todos os estilos (*elementos genéricos*), e também partes aparecem em grupos (*elementos mistos*).

- **Arquivos HTML evitando redundâncias** Com a necessidade de evitar a redundância de informação, surge a possibilidade de criar-se não apenas quatro arquivos, porém mais arquivos, em que existiriam partes genéricas, por exemplo, sendo compartilhadas entre os quatro estilos. O problema começa a surgir com o aumento da complexidade e a maneira de efetuar a união destes arquivos.

Para isso seria necessário a criação de uma linguagem, que ao ser processada fizesse a união desses arquivos. Contudo, existem linguagens, tais como o CSS e XSL, que poderiam ser utilizadas para a união destes arquivos, sendo necessário portanto a escrita da ferramenta que manipulasse uma destas linguagens.

- **Arquivo XML** As abordagens anteriores apresentam sérios problemas relativos ou a redundância ou ao aumento da complexidade de sua estrutura para evitar esta redundância.

Utilizando a linguagem extensível de marcação (XML), consegue-se a estruturação necessária, e deste modo representa-se o conteúdo através de elementos genéricos, específicos e mistos (grupos). Além disso, esta linguagem possibilita uma maior portabilidade dos dados entre plataformas.

Contudo, a complexidade de escrita de conteúdo aumenta, pois torna-se necessário que o autor do conteúdo tenha conhecimento da gramática definida pela DTD, visto mais detalhadamente a seguir, e da linguagem XML. Porém, a linguagem XML pode ser aprendida com facilidade pois é uma linguagem simples e bastante parecida com HTML.

A parte mais complexa desta linguagem está na definição da DTD que define a

estrutura do curso. Esta definição deve ser efetuada uma única vez, sendo esta durante a implementação da ferramenta, com isso a autoria do conteúdo do curso escrito em XML passa a utilizar desta definição, tornando simples para o autor sua autoria.

Como a melhor opção é o XML, passa-se a analisar a especificação e uso da DTD.

### 3.4.2 Especificação da DTD e suas implicações

A estruturação do conteúdo é obtida basicamente pela definição de uma gramática livre de contexto que serve de base para a análise sintática do documento XML. Esta gramática fica armazenada em um arquivo de nome *conteudo.dtd*. Nota-se que esta DTD segue a proposição realizada na seção 3.4, que sugere a especificação de alguns elementos para determinados estilos (elementos específicos) e de outros para grupos (elementos mistos) ou ainda elementos comuns a todos os estilos (genéricos). Esta classificação é realizada pela presença do atributo **proprietario**, que pode assumir os valores: **diverger**, **assimilator**, **converger**, **accommodator**, **generico**, **diverger\_assimilator**, **assimilator\_converger**, **converger\_accommodator**, **acomodator\_diverger**, sendo os quatro primeiros relacionados aos estilos definidos pelo modelo, seguidos pelo qualificador genérico e seguido ainda pelos qualificadores mistos (grupos).

A seguir, na figura 3.6 vê-se um trecho do arquivo DTD, que define a gramática de validação do conteúdo, em que se pode observar as estruturas principais que o compõem.

O elemento **Capitulo** possui respectivamente um título do capítulo (**TituloCap**), uma descrição do capítulo (**DescricaoCap**) e uma ou mais seções (**Secao**). Além disso, possui como atributo um identificador (id **ID**) que deve ser único em todo o arquivo XML e será utilizado para recuperar o conteúdo do elemento identificado pelo identificador **id**, durante a interação com o usuário.

O elemento **Secao** refere-se as seções que compõem cada capítulo. Este elemento possui os mesmos atributos do elemento **Capitulo**. As diferenças surgem, pois os elementos **Secao** podem possuir uma ou mais sub-seções entremeadas por componentes de seção (**CompSec**). Além disso, no final de cada seção pode aparecer um ou mais testes (**Teste**).

```

<!ELEMENT Capitulo (TituloCap, DescricaoCap, Secao+)>
<!ATTLIST Capitulo
    id ID #REQUIRED
    proprietario (generico | assimilator | diverger | accommodator |
    converger | accommodator_diverger |
    diverger_assimilator | assimilator_converger |
    converger_accommodator) #REQUIRED>
<!ELEMENT TituloCap (#PCDATA|%textohtml;)*>
<!ELEMENT DescricaoCap (p)>
<!ELEMENT Secao (TituloSec, DescricaoSec, (CompSec | SubSec)+, Teste*)>
<!ATTLIST Secao
    id ID #REQUIRED
    proprietario (generico | assimilator | diverger | accommodator |
    converger | accommodator_diverger |
    diverger_assimilator | assimilator_converger |
    converger_accommodator) #REQUIRED>
<!ELEMENT TituloSec (#PCDATA|%textohtml;)*>
<!ELEMENT DescricaoSec (p)>

```

Figura 3.6: Trecho da DTD

Os componentes `CompSec` por sua vez, são compostos por elementos definidos pelas duas entidades presentes na DTD (`textohtml` e `estrutura`). A entidade `textohtml` referenciada na DTD da figura 3.6 é definida da forma:

```
<!ENTITY % textohtml "p|em|b|i|u|s|big|small|sub|sup|cite|code|pre|font|a|br">
```

ou seja, ela representa os elementos definidos pela DTD de nome e significado semântico iguais às *tags* HTML. A outra entidade `estrutura` define as estruturas do HTML: tabelas, listas, imagens, linha horizontal. Assim, consegue-se reutilizar os conhecimentos sobre a linguagem HTML na autoria do conteúdo.

O conteúdo do curso é finalmente escrito, em um arquivo XML, de acordo com a gramática definida pela DTD. A seguir, na figura 3.7, vê-se um trecho deste arquivo que contém o conteúdo do curso:

Pode-se notar neste trecho de código que existe um componente de seção genérico e outro específico (`diverger`). Desta forma, para que seja possível apresentar o conteúdo corretamente para um dos estilos de aprendizagem, precisa-se selecionar todos os componentes genéricos, específicos e os dois grupos nos quais o estilo está inserido (elementos mistos), ou seja, os elementos que possuem atributo `proprietario` com os valores

```
<CompSec proprietario="generico">
  <dl>
    <dt>Crítica</dt>
    <dd>em que o atraso em seu atendimento pode acarretar graves
      conseqüências para o sistema. </dd>
    <dt>Não-Essencial</dt>
    <dd>pode ocorrer algum atraso no cumprimento da tarefa, que apenas
      se torna indesejável se ocorrer por um longo período.</dd>
  </dl>
</CompSec>
<CompSec proprietario="diverger">
  <p> Tarefas de tempo-real são encontradas em geral em sistemas que
    exigem alta confiabilidade, por interferirem em situações que
    colocam em perigo trabalhos minuciosos, vidas ou empresas. </p>
</CompSec>
```

Figura 3.7: Trecho do documento XML

generico, estilo e grupo, em que *estilo* assume o valor de um dos estilos definido pelo modelo de Kolb e *grupo* assume o valor dos grupos aos quais o estilo pertence.

Para efetuar esta seleção utiliza-se um arquivo *.xsl*, do qual se vê um trecho de código na figura 3.8.

A exemplo do que ocorre com a DTD, também o arquivo XSL é especificado uma única vez, visto que a finalidade deste é somente promover a seleção dos elementos que já foram especificados pela DTD segundo o estilo e o identificador (id). Desta forma, os códigos XSL estão inseridos no componente ManipuladorXMLXSL, sendo esta a razão da existência de variáveis como a *proprietario*, no trecho apresentado na figura 3.8, pois apesar do XSL ser único para a seleção de seções, capítulos ou do conteúdo é necessário que os atributos *proprietario* e *id* sejam ajustados para a pesquisa necessária neste instante.

```

"<xsl:template match=\"TituloCon\">" +
"  <h1 align=\"center\">" +
"    <xsl:apply-templates select=\"text()|em|b|i|u|s|big|" +
"      small|sub|sup|cite|code|pre|font|a|br\"/>" +
"  </h1>" +
"</xsl:template>" +
"<xsl:template match=\"DescricaoCon\">" +
"  <xsl:apply-templates/>" +
"</xsl:template>" +
"<xsl:template match=\"Capitulo[@proprietario='"+
"generico']\">" +
"<xsl:apply-templates/>" +
"</xsl:template>" +
"<xsl:template match=\"Capitulo[@proprietario='"+
"proprietario + '\"\">" + "<xsl:apply-templates/>" +
"</xsl:template>" + .....

```

Figura 3.8: Trecho do XSL

# Capítulo 4

## Resultados Obtidos

A ferramenta original, RTsim [Man01], destinava-se a simulação de algoritmos para escalonamento de tarefas em sistemas de tempo-real, sendo pouco adequada ao ensino.

Esta ferramenta foi definida por Paula Bueno [Pau96], inicialmente com poucos algoritmos e com uma interface básica, recebendo várias melhorias com a inclusão de novos algoritmos e uma nova interface gráfica. Posteriormente seu código foi submetido a reengenharia [Mio01], mas permanecendo sem características que facilitassem o aprendizado, tais como a apresentação de conteúdo, exercícios adequados, etc.

Expõe-se neste capítulo os resultados alcançados com a especificação e a implementação do módulo RTTutor. Esses resultados divididos em quatro grupos, que são as capacidades acrescentadas ao RTsim, a facilidade de uso e manutenção, portabilidade e desempenho da ferramenta.

### 4.1 Capacidades acrescentadas

Adicionou-se à ferramenta a apresentação de conteúdos segundo os estilos de aprendizagem definidos por Kolb. Deste modo, alcançou-se uma melhora sensível em relação à ferramenta anterior, por meio da criação do conteúdo de acordo com o modelo de aprendizagem de Kolb.

Através de comparações entre os conteúdos apresentados aos diversos estilos definidos por Kolb, percebeu-se que a ferramenta realmente cumpre o que se propõe, já que

apresenta um conteúdo de acordo com as especificações propostas para cada estilo.

Pode-se ver claramente essa diferença quando se compara o conteúdo gerado para o estilo *divergente* com o conteúdo gerado para o tipo *acomodador*, em que o conteúdo apresentado ao primeiro estilo é bem maior. Isso era esperado pois este estilo gosta de diferentes perspectivas sobre o assunto e assim são apresentados exemplos e justificativas juntamente com os conceitos. Já o estilo *acomodador* não gosta de muitos exemplos e justificativas e gosta de aprender com experiências práticas. Assim grandes quantidades de conceitos não são apresentados em seu conteúdo.

Para confirmar numericamente esse resultado, redirecionou-se a saída de uma seção apresentada tanto ao estilo *divergente* quanto ao *acomodador* para um arquivo *.html*. Os arquivos tinham 22Kb e 16Kb respectivamente para *divergente* e *acomodador*. Isso confirma a diferença do material no que diz respeito à quantidade de leitura necessária.

## 4.2 Facilidade de uso e manutenção

Através do emprego das bibliotecas gráficas fornecidas pela tecnologia Java conseguiu-se atingir uma interface com usuário amigável e intuitiva, que reutiliza a familiaridade do usuário com os gerenciadores de arquivos gráficos tal como o Konqueror (KDE) e o MS Windows Explorer (MS Windows).

Alcançou-se também facilidade de manutenção devido ao emprego de técnicas propostas pela engenharia de software tais como reutilização de componentes, hierarquia de classes, encapsulamento e ocultamento de informação. Para este fim empregou-se a tecnologia Java na implementação da ferramenta e a tecnologia XML para a estruturação do conteúdo visando também a construção de componentes reutilizáveis evitando assim a redundância e facilitando a manutenção.

## 4.3 Portabilidade

A linguagem Java é uma tecnologia que garante a portabilidade de código, visto que foi desenvolvida com este intuito. Para a portabilidade de informações usou-se a tecnologia



		Ambiente		
		MS Windows	Linux	Solaris
Módulo	Servidor	✓	✓	✓
	Cliente	✓	✓	✓

Tabela 4.1: Relação de portabilidade dos módulos

XML, que está sendo amplamente utilizada para troca de informações, o que mostra que ela permite a portabilidade dos dados. Deste modo, sendo esta ferramenta totalmente implementada usando tecnologias que garantem portabilidade, alcançou-se também a portabilidade da ferramenta.

Realizaram-se diversos testes entre plataformas, com os módulos da ferramenta como ve-se na tabela 4.1. Dentre estes testes pode-se apresentar a compilação do código todo no sistema operacional Linux, a execução do servidor em Solaris e a execução de diversos clientes em sistemas MS Windows. Em todos os testes os vários módulos funcionaram perfeitamente, o que garante a portabilidade da ferramenta para essas plataformas.

## 4.4 Desempenho

Sabe-se que linguagens interpretadas não são tão eficientes quanto linguagens compiladas. Como a ferramenta foi implementada usando a linguagem Java, que é interpretada, é possível que ela não tenha um alto desempenho.

Vê-se que apesar da perda de desempenho, causada pela interpretação dos bytecodes, o emprego de algumas técnicas tais como a utilização da estrutura de *cache* no servidor minimizou-se esta perda.

A tabela 4.2 mostra a média dos tempos de processamento das requisições efetuadas pelos clientes ao servidor em diversas configurações. Esses valores foram obtidos pela média aritmética dos tempos de resposta para dez execuções.

Para que fosse possível avaliar qual o desempenho em situações reais de uso colocou-se cargas no servidor. Para isso, além do próprio cliente foram usados mais 20 clientes que efetuavam continuamente pedidos de conteúdo. Assim, obteve-se os resultados apresen-

tados na tabela 4.3, porém para um número menor de configurações.

<b>Servidor</b>	<b>Cliente</b>	<b>sem Cache(miliseg)</b>	<b>com Cache(miliseg)</b>
Windows <sup>1</sup>	Sun Blade 100 <sup>2</sup>	749.2	261
Sun Blade 1000 <sup>3</sup>	Linux <sup>4</sup>	751.5	98.8
Sun Blade 1000 <sup>3</sup>	Sun Blade 100 <sup>5</sup>	841.9	169
Sun Blade 100 <sup>5</sup>	Linux <sup>4</sup>	1357.4	98.6
Sun Blade 100 <sup>5</sup>	Sun Blade 1000 <sup>3</sup>	1452.8	193.5
Linux <sup>4</sup>	Linux <sup>4</sup>	2039.1	16.6
Linux <sup>4</sup>	Sun Blade 100 <sup>5</sup>	2133.1	100

Tabela 4.2: Tempos de resposta sem carga no servidor

<b>Servidor</b>	<b>Cliente</b>	<b>sem Cache(miliseg)</b>	<b>com Cache(miliseg)</b>
Sun Blade 1000 <sup>3</sup>	Linux <sup>4</sup>	909.9	97.8
Linux <sup>4</sup>	Linux <sup>4</sup>	1220.5	49.5
Windows <sup>1</sup>	Sun Blade 100 <sup>2</sup>	1024.4	298.3
Linux <sup>4</sup>	Sun Blade 1000 <sup>3</sup>	1277.5	134.3

Tabela 4.3: Tempos de resposta com carga no servidor

Com estes resultados percebe-se que as estações de trabalho Sun, apesar de apresentarem um desempenho inicialmente inferior aos PCs, com a inserção de carga seu desempenho não se altera, ao contrário do que ocorre com os PCs, nos quais há uma queda acentuada no desempenho. Esta situação decorre de características de arquitetura da máquina e políticas de implementação de acesso a memória e escalonamento de processos nos diversos sistemas operacionais.

Nota-se que apesar das máquinas apresentarem arquiteturas diferentes, a quantidade de memória utilizada foi a mesma, já que a máquina virtual java (JVM) utiliza, como configuração padrão, no máximo 64 Mbytes de memória, e foi com tal configuração que os testes foram realizados.

<sup>1</sup>PC Intel Pentium II 350 Mhz 128Mb Ram

<sup>2</sup>Sparc SunBlade 100 1Gb Ram

<sup>3</sup>UltraSparc SunBlade 1000 1Gb Ram

<sup>4</sup>PC Intel Pentium II 350 Mhz 128Mb Ram

<sup>5</sup>Sparc SunBlade 100 256Mb Ram

Para que o servidor possa obter um desempenho melhor e não apresentar problemas de falta de memória quando muitos usuários se conectarem, faz-se necessária a utilização de alguns parâmetros no momento de instanciá-lo. Estes parâmetros são: `-Xms<tamanho>`, `-Xmx<tamanho>` e `-Xss<tamanho>`, que são, respectivamente, o tamanho inicial de memória dinâmica, o tamanho máximo de memória dinâmica e o tamanho da pilha de *threads*.

# Capítulo 5

## Conclusões e perspectivas futuras

Neste capítulo apresenta-se as considerações finais a respeito do desenvolvimento de todo o projeto. Assim, na seção 5.1 descreve-se as principais conclusões baseadas nos resultados apresentados no capítulo anterior e, a seguir, na seção 5.2 apresenta-se algumas propostas de trabalhos futuros que visam melhorar o funcionamento do simulador e do módulo RTTutor.

### 5.1 Conclusões

Buscou-se durante o desenvolvimento deste projeto a melhoria das capacidades de ensino de sistemas de tempo-real providas pela ferramenta RTsim. Como visto através dos resultados apresentados no capítulo anterior, foi disponibilizado material instrutivo para os alunos, resolvendo assim a deficiência proveniente da concepção inicial do projeto da ferramenta.

Para a apresentação desse material utilizou-se uma técnica de ensino que baseia-se em estilos de aprendizagem, adaptando-se a apresentação do conteúdo ao estilo pessoal do estudante.

O modelo de Kolb foi adotado por satisfazer plenamente os objetivos do projeto no que diz respeito a adequação ao estilo do usuário, e por modelar a camada mais diretamente passível de ser atingida por uma ferramenta computacional.

Para que se alcançasse a implementação computacional de tal técnica foi necessário o

emprego de uma tecnologia que tornasse viável a manutenção do conteúdo, eliminando a redundância que decorre de características comuns entre os estilos, utilizou-se para tanto a tecnologia XML.

Com a utilização da arquitetura cliente/servidor abriu-se a possibilidade de criação de um controle estatístico que permite ao professor acompanhar o progresso dos alunos que utilizam o tutor. Alcançou-se ainda, através da centralização do conteúdo, a transparência do processo de atualização do mesmo, sendo que o professor encarrega-se deste processo sem que seja necessário que o aluno se preocupe em buscar novas versões de conteúdo.

O desempenho foi melhorado significativamente com a utilização da estrutura de *cache* que armazena as partes do conteúdo de determinado estilo, e assim, quando um outro usuário de mesmo estilo efetua uma requisição do mesmo conteúdo esta pode ser atendida sem que haja a necessidade de nova transformação do conteúdo XML para HTML.

Finalmente, este módulo se enquadra como solução à proposta de construção de um tutor que se adapte ao estilo do usuário apresentada no projeto de final de graduação de Marcelo de Brito Miola[Mio01].

## 5.2 Perspectivas futuras

Este trabalho abre possibilidades para diversos trabalhos complementares, descritos adiante:

- **Ferramenta para escalonamento manual:** O RTsim efetua escalonamentos de um conjunto de tarefas de tempo real, para um conjunto de dados e apresenta os resultados destes. No entanto, para os estilos de aprendizagem que apresentam como característica a preferência por efetuar trabalhos práticos, tem-se a necessidade que efetuar o trabalho inverso, ou seja, não somente observar escalonamentos prontos, mas realiza-los manualmente.

Neste cenário, sugere-se a construção de uma ferramenta que permita ao estudante realizar manualmente o escalonamento de um conjunto de tarefas segundo algum algoritmo e submeter este escalonamento a uma comparação com o mesmo conjunto de tarefas e algoritmo realizadas pelo simulador (RTsim).

- **Sala de discussão (Chat):** Devido ao fato de dois dos estilos de aprendizagem definidos pelo modelo de Kolb apresentarem a necessidade de discutir o conteúdo com outros estudantes, sugere-se a criação de uma ferramenta de *chat* que supre esta necessidade, atentando-se ainda para os papéis que devem ser desempenhados pelos estudantes de cada estilo. Estes papéis também estão definidos pelo modelo, sendo que, por exemplo, os “acomodadores” devem atuar como coordenadores de grupos.
- **Ferramenta para autoria de conteúdo:** Construindo-se uma ferramenta de autoria gráfica o autor do conteúdo poderá ocupar-se somente com a seleção e construção de elementos para os diversos estilos propostos por Kolb, sem que haja a necessidade de se ocupar em aprender XML.
- **Extensão das funcionalidades do servidor:** A implementação do servidor deste projeto pode ser estendida para prover o serviço de escalonamento de conjuntos de tarefas de tempo real. Dessa forma o RTsim que atualmente é composto por duas partes, a interface com o usuário e o núcleo escalonador de tarefas, passaria a ter o núcleo escalonador incorporado ao servidor e a interface com o usuário se tornaria um cliente que efetua chamadas ao servidor.

Uma questão decorrente desta nova configuração, é a de que a inclusão desta nova funcionalidade sobrecarregaria o servidor, mas a grande ocorrência de escalonamentos idênticos realizados por diversos estudantes abre a possibilidade de eliminar essa sobrecarga pela extensão da estrutura de *cache*, para que a mesma possa armazenar escalonamentos já realizados.

# Referências Bibliográficas

- [Arc02] Arciniegas, F. - *C++ XML*; Makron Books, 2002.
- [Arm02] Armstrong, E - *Understanding XML em The Java(tm) Web Services Tutorial*; Sun Microsystems, Palo Alto, Estados Unidos, 2002.
- [Cur87] Curry, L (1987). - *Integrating concepts of cognitive or learning style: A review with attention of psychometric standards*; Canadian College of Health Service Executives, Ottawa, Canada.
- [Cur90] Curry, L. (1990). - *A critique of the research on learning styles*; Educational Leadership, October: 50-52.
- [Ctl02] *CTL Learning Styles Site*;  
<http://www.indstate.edu/ctl/styles/model.html>,  
último acesso em: 04/11/2002
- [Dec00] Décio, O. C. - *Guia de consulta rápida XML*; Novatec, 2000.
- [Idn02] *IDNM - Instructional Design For The New Media*;  
[http://campus.skillnet.ca/course\\_demos/Learn/mod2/mod2-toc.htm](http://campus.skillnet.ca/course_demos/Learn/mod2/mod2-toc.htm)  
último acesso em: 04/11/2002
- [Ing99] Ingham J., Hein T. Buduy D. - *Teaching and Learning with Style*; Frontiers in Education - 1999
- [Kol76] Kolb D. A. - *Learning Style Inventory*. Boston: McBer. (1976)

- [Ler02] *The Dunn and Dunn Learning Style Model of Instruction*;  
www.unc.edu/depts/ncpts/publications/learnstyles.htm, último acesso em:  
04/11/2002
- [Man01] Manacero Jr, A., Miola, M. B., Nabuco, V. A. - *Teaching real-time with a scheduler simulator*; Frontiers In Education: The future-impact on engineering and science education. Reno: IEEE, 2001. v. CD-ROM. p.T4D15 - T4D19
- [Mcc02] McCarthy, B. - *The 4MAT System*,  
<http://www.valdosta.peachnet.edu:80/whuitt/psy702/instruct/4mat.html>  
último acesso: 04/11/2002
- [Mio01] Miola, M. B. - *Reengenharia aplicada ao RTsim*; monografia de projeto final apresentada em 12/2001, S.J. do Rio Preto, SP.
- [Mye62] Myers I. - *Myers-Briggs Type Indicator*; Palo Alto, CA: Consulting Psychologists Press, 1962.
- [Pau96] Paula Bueno, L. P. - *Simulador para análise de desempenho de escalonadores de tempo-real - uma evolução*; monografia de projeto final apresentada em 11/1996, S.J. do Rio Preto, SP.
- [Pre95] Pressman, R. S. - *Engenharia de Software*, Makron Books, 1995.



# Anexo A

## Inventário de estilos de aprendizagem de Kolb

Este teste descreve o modo como os indivíduos aprendem e como trabalham com idéias e situações do dia-a-dia. Na tabela A.1 encontra-se 12 sentenças e quatro possíveis finais para cada uma delas. Deve-se ordenar os finais por ordem de preferência, atribuindo o valor 4 para o que mais se adequa às preferências pessoais, 3 para o seguinte, 2 para o próximo e 1 para o que menos se adequa. O arranjo dos finais é feito de modo que os finais das primeiras linhas em cada sentença caracterizem uma preferência por experiência concreta, as segundas linhas por observação reflexiva, as terceiras por conceitualização abstrata e as últimas por experimentação ativa.

---

1	Quando aprendo	gosto de trabalhar meus sentidos
		gosto de ouvir e escutar
		gosto de pensar nos conceitos
		gosto de estar fazendo coisas

---

2	Aprendo melhor quando	confio em meus sentimentos
		observo e escuto atentamente
		me baseio em raciocínio lógico
		trabalho para ver as coisas prontas

---

3	Quando estou aprendendo	tenho sentimentos e reações fortes sou quieto e reservado procuro colocar razões nas coisas sou responsável pelas coisas
4	Aprendo por	sentimento observação pensamento ação
5	Quando eu aprendo	estou pronto para novas experiências examino todos os lados da questão procuro analisar as coisas e dividi-las em partes gosto de experimentar as coisas
6	Quando estou aprendendo	sou intuitivo sou observador sou lógico sou ativo
7	Aprendo melhor a partir de	relacionamentos pessoais observação teorias racionais uma chance de praticar e experimentar
8	Quando aprendo	me sinto pessoalmente envolvido espero antes de agir prefiro conceitos e teorias prefiro ver resultados a partir de meu trabalho
9	Aprendo melhor quando	me baseio em meus sentimentos me baseio em minhas observações me baseio em minhas idéias posso experimentar coisas
10	Quando estou aprendendo	sou receptivo sou reservado sou racional sou responsável

11	Quando aprendo	me envolvo observo avalio sou ativo
12	Aprendo melhor quando	estou receptivo e aberto estou cuidadoso analiso as idéias sou prático

Tabela A.1: Questões do teste de Kolb

Totaliza-se os pontos atribuídos para os finais das primeiras linhas, obtendo-se o coeficiente de preferência em experiência concreta. Repetindo-se este processo para os finais das demais linhas obtêm-se os coeficientes de observação reflexiva, conceitualização abstrata e experimentação ativa, respectivamente.

Deve-se marcar estes coeficientes no gráfico da figura A.1. Unindo-se estes pontos obtém-se um quadrilátero irregular onde as dimensões que forem maiores denotarão a preferência do aluno pela característica marcada neste eixo.

Para obter qual o estilo do indivíduo deve-se efetuar uma operação de subtração entre as características que se opõem em cada um dos eixos do gráfico da figura A.1, ou seja, para o eixo  $x$  calcula-se  $EA - OR$  e para o eixo  $y$ :  $CA - EC$ . Mapeia-se, em seguida, o ponto determinado por este par ordenado, na figura A.2, este por sua vez, estará inserido em um dos quadrantes que determinam cada um dos estilos definidos por Kolb.

## A.1 Descrição dos estilos de aprendizagem

### A.1.1 Divergentes

A questão típica deste estilo de aprendizagem é “Por quê?”. Os estudantes deste estilo respondem bem a explicações de como o material estudado se refere a suas experiências, interesses e carreiras futuras. Divergentes captam as informações através de experiência

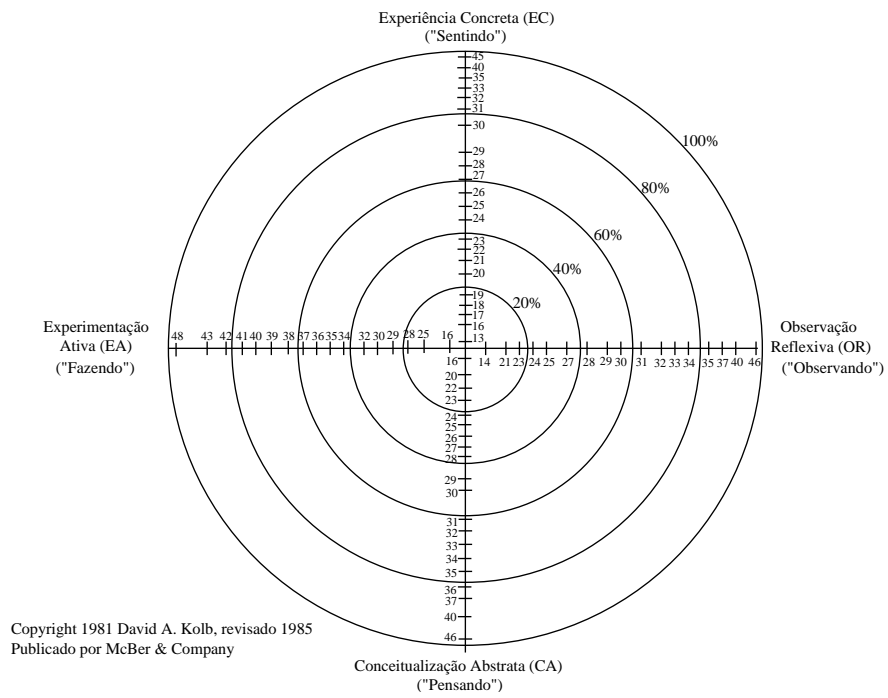


Figura A.1: O círculo de aprendizado

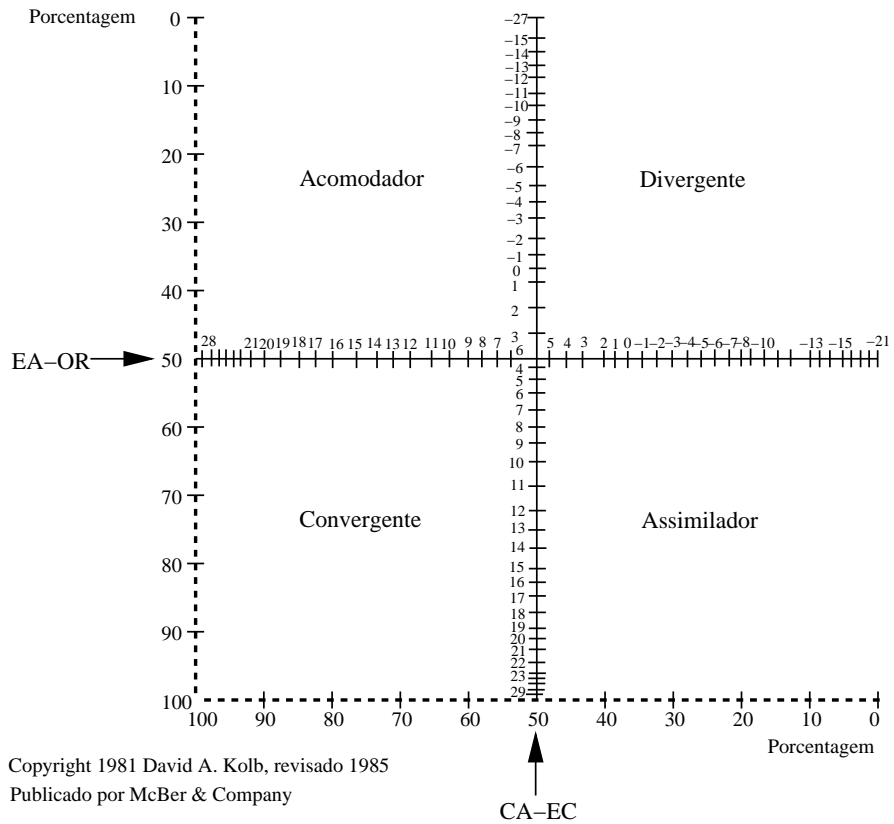


Figura A.2: Grade de estilos de aprendizado

concreta, baseando-se nos sentimentos, precisam expressar seus sentimentos quando estão aprendendo, procurando por significados pessoais sobre o que aprendem, e querem interagir com professores e colegas. Eles precisam ser convencidos de que o material estudado é importante para eles como pessoas e/ou para outras pessoas as quais eles sintam empatia ou se identifiquem. Aprendem através de discussões e são interessados em como as informações afetam as pessoas. Trabalham para manter o grupo harmonioso e são chamados Divergentes porque podem ver as coisas de diferentes perspectivas e geram novas idéias facilmente. Assim saem-se muito bem em reuniões do tipo “brainstorming”.

Para obter sucesso com estes estudantes deve-se atuar como motivador, motivando-lhes a aprender o conteúdo e mostrando a eles como o conteúdo se encaixa em suas vidas. Divergentes querem interagir pessoalmente com seus instrutores e serem reconhecidos como indivíduos. Assim, o instrutor deve testemunhar o crescimento pessoal do estudante.

### **A.1.2 Assimiladores**

A questão típica deste estilo de aprendizagem é “O que?”. Estudantes deste estilo são bem sucedidos quando as informações são apresentadas de maneira organizada e lógica, e obtêm aproveitamento melhor se tiverem tempo para refletir.

Assimiladores captam as informações através de conceitualização abstrata e as processam através de observação reflexiva. Eles gostam de informações que justifiquem o que estão aprendendo, querem saber o que os especialistas pensam e procuram compreender os conceitos. São bem sucedidos na escola tradicional. De fato, a maioria dos estudantes, apesar de terem seus estilos próprios, se adaptam ao estilo do Assimilador para obter sucesso na escola. Eles são chamados Assimiladores porque utilizam-se de pedaços de informações, analisam, organizam, e assimilam-os como um todo. Gostam de organizar, e tem tendência a serem detalhistas. Tem sucesso nos procedimentos quando se baseiam na literatura. Nas suas abordagens são cuidadosos, metódicos e cautelosos, tentando evitar erros.

Eles aprendem bem lendo sobre os tópicos e preferem não trabalhar em grupos, exceto para seguir normas. Para ser efetivo, o instrutor deve se comportar como um especialista. Assim, o papel do instrutor é unicamente ser uma autoridade no assunto e uma fonte

de informações. Assimiladores gostam de assistir aulas e/ou palestras e preferem que o professor explique o conteúdo de forma organizada e precisa.

### **A.1.3 Convergentes**

a questão típica deste estilo de aprendizagem é “Como?”. Os estudantes deste estilo gostam de ter a oportunidade de trabalhar ativamente em tarefas bem definidas, e de aprender por tentativa e erro em um ambiente que permita-lhes errar com segurança.

Os Convergentes captam as informações através de conceitualização abstrata e as processam de forma ativa. Gostam de testar as informações, tentar coisas, perceber coisas a parte, ver como as coisas funcionam, e aprender fazendo. Testam a viabilidade e utilidade das informações. Assim, não gostam de permanecerem sentados por um longo período, assistindo aulas ou lendo uma grande quantidade de material. Eles dão ênfase a fazer as coisas sem perder tempo; são aprendizes que convergem mais rapidamente ou chegam rapidamente a tomar decisões, que procuram por uma resposta correta, e que cortam caminho, passando somente pela parte mais essencial da matéria. Gostam de exemplos a seguir, em detrimento de assistir a aulas ou palestras. O ambiente de laboratório é ideal para eles. Tendem a ver o trabalho em grupo e as discussões como perda de tempo, porque podem fazer o trabalho mais rapidamente de forma individual.

Para obter sucesso com os convergentes, o instrutor deve trabalhar como um técnico, provendo práticas orientadas e respostas. Esses estudantes preferem que o professor os ajudem a colocar as lições em prática. O instrutor precisa se tornar menos ativamente envolvido, agindo como uma fonte de informações e permitir ao Convergente ter um papel mais ativo.

### **A.1.4 Acomodadores**

A questão típica deste estilo de aprendizagem é “E se?”. Os estudantes deste estilo gostam de aplicar as informações em novas situações para resolver problemas reais. Os Acomodadores captam as informações através de experiências concretas e processam-nas através de experimentação ativa. Eles são entusiastas e preferem aprender através

de descobertas próprias. Seguem seus próprios horários e ordem quando estudam e não gostam de muitos procedimentos e regras. Por outro lado, gostam de interação com outras pessoas, de discussões em grupo, de obter informações e criar algo novo com elas, e ainda de descobrir coisas por sua própria conta. São solucionadores de problemas e assumem riscos, aprendendo com seus erros; são chamados de Acomodadores porque usam o que aprenderam adaptando estes conhecimentos para uso próprio; utilizam sua criatividade para realizar esta adaptação e fazer as coisas cada vez melhores.

Para obter sucesso com alunos deste estilo o instrutor deve possibilitar ao estudante descobrir as coisas por conta própria. O papel do instrutor é o de avaliador e revisor. Esses estudantes precisam do professor para encorajá-los às descobertas, permitindo-lhes se auto-instruirmos. Eles preferem alguma interação com o instrutor, mas pouca supervisão e querem que o instrutor permaneça em segundo plano, para servir como um recurso extra, e para avaliar o que eles estão fazendo.