

**FUNDAÇÃO DE ENSINO “EURÍPEDES SOARES DA ROCHA”  
CENTRO UNIVERSITÁRIO “EURÍPEDES DE MARÍLIA” – UNIVEM  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO  
TRABALHO DE CONCLUSÃO DE CURSO**

**ALINE CRISTINA SCANAVACCA**

**ANÁLISE E PROJETO DE UM FRAMEWORK BASEADO EM UMA  
LINGUAGEM DE PADRÕES NO DOMÍNIO DE SOFTWARE  
EDUCACIONAL**

**MARÍLIA  
2007**

**ALINE CRISTINA SCANAVACCA**

**ANÁLISE E PROJETO DE UM FRAMEWORK BASEADO EM UMA  
LINGUAGEM DE PADRÕES NO DOMÍNIO DE SOFTWARE  
EDUCACIONAL**

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação do Centro Universitário “Eurípedes de Marília”, mantido pela Fundação de Ensino “Eurípedes Soares da Rocha”, para obtenção do Título de Bacharel em Ciência da Computação.

Orientador(a):  
Prof<sup>ª</sup>. Dra. Maria Istela Cagnin Machado

**MARÍLIA  
2007**

SCANAVACCA, Aline Cristina.

Análise e Projeto de um Framework Baseado em uma Linguagem de Padrões no Domínio de Software Educacional / Aline Cristina Scanavacca; orientador: Maria Istela Cagnin Machado. Marília, SP :[s.n.], 2007.

97 f.

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro Universitário Eurípedes de Marília – Fundação de Ensino Eurípedes Soares da Rocha.

1. Engenharia de Software 2. Módulos Educacionais  
3. Padrões 4. Linguagem de Padrões 5. Frameworks

CDD: 005.1



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL**

---

Aline Cristina Scanavacca

**ANALISE E PROJETO DE UM FRAMEWORK BASEADO EM UMA LINGUAGEM DE PADRÕES  
NO DOMÍNIO DE SOFTWARE EDUCACIONAL**

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: 9.5 ( nove e meio )

Orientador: Maria Istela Cagnin Machado

1º. Examinador: Márcio Eduardo Delamaro

2º. Examinador: Valter Vieira de Camargo

Three handwritten signatures are present, each written over a horizontal line. The top signature is for Maria Istela Cagnin Machado, the middle for Márcio Eduardo Delamaro, and the bottom for Valter Vieira de Camargo.

Marília, 06 de novembro de 2007.

## **AGRADECIMENTOS**

A Deus onipotente, criador de toda a ciência.

A meus pais Edivalde e Vera que me incentivaram desde o momento em que decidi  
minha carreira.

A Prof<sup>ª</sup> Maria Istela Cagnin Machado pela orientação deste trabalho.

SCANAVACCA, Aline Cristina. **Análise e Projeto de um Framework Baseado em uma Linguagem de Padrões no Domínio de Software Educacional**. Trabalho de Conclusão de Curso - Centro Universitário Eurípedes de Marília, Fundação de Ensino Eurípedes Soares da Rocha, Marília, 2007.

## RESUMO

Este trabalho tem como objetivo utilizar, de forma sistemática, a Engenharia de Software no desenvolvimento de módulos educacionais, proporcionando novas perspectivas de construção e de qualidade de software baseados em conceitos e uso de processos de desenvolvimento, linguagem de padrões e *framework*. A pesquisa objetiva explorar o uso de *frameworks* baseados em linguagens de padrões para apoiar o desenvolvimento de módulos educacionais, em especial, materiais instrucionais para aulas de ambientes virtuais de aprendizado, os quais podem ser voltados para o ambiente *Web*. Para permitir isso, é realizada a análise do domínio de softwares educacionais a fim de possibilitar a geração de uma linguagem de padrões, esta constituída por padrões já existentes e por padrões adicionais desenvolvidos neste trabalho. Por conseguinte, a linguagem de padrões obtida é aplicada na análise e projeto de um *framework* caixa-branca no domínio de softwares educacionais nela baseada.

**Palavras-chave:** Software Educacional, Engenharia de Software, Reúso de Software, Linguagens de Padrões, *Frameworks*.

SCANAVACCA, Aline Cristina. **Análise e Projeto de um Framework Baseado em uma Linguagem de Padrões no Domínio de Software Educacional**. Trabalho de Conclusão de Curso - Centro Universitário Eurípedes de Marília, Fundação de Ensino Eurípedes Soares da Rocha, Marília, 2007.

## ABSTRACT

This work have as objective uses, in a systematic way, the Software Engineering in the educational modules development, providing new construction perspectives and software quality based on concepts and use of development process, pattern languages and framework. The research objectify to explore the frameworks use based on pattern languages to support the educational modules development, especially, instructional material for learning virtual environments classes, which can be gone back to the Web environment. To allow that, it be made the domain analysis of educational softwares to make possible the generation of a pattern language, this constituted by already existent patterns and for additional patterns developed in this work. Consequently, the pattern language obtained to be applied in the analysis and project of a white-box framework in the domain of educational softwares in it based.

**Keywords:** Educational Software, Software Engineering, Software Reuse, Pattern Languages, Frameworks.

## LISTA DE FIGURAS

Figura 2-1 Ilustração da abordagem proposta pelo processo de desenvolvimento de <i>framework</i> baseado em linguagem de padrões de Braga (2002) (Adaptado de Braga (2002)).	29
Figura 2-2 Passos do processo proposto por Braga (2002).	31
Figura 3-1 – A linguagem de padrões “CogLearn”, seus grupos e níveis de abstração (Neto,2005).	34
Figura 3-2 Estrutura da linguagem de padrões “LPCAVirtu” considerando contexto dos padrões que a compõe.	36
Figura 3-3 Padrão ESTENDER CONHECIMENTO e suas variações	39
Figura 3-4 Padrão AVALIAR ALUNO e suas variações.	41
Figura 3-5 Exemplo da aplicação do padrão RELACIONAR CONTEÚDO por meio de questão introdutória do histórico da Engenharia de Software	43
Figura 3-6 Aplicação do padrão ESTENDER CONHECIMENTO com as variantes de EXEMPLIFICAÇÃO e METAFORIZAÇÃO, respectivamente.	44
Figura 4-1 Processo para a construção de um framework baseado em linguagem de padrões (Adaptado de Braga (2002)).	48
Figura 4-2 Arquitetura do framework proposto neste trabalho	52
Figura 4-3 Diagrama de Classes de Projeto do “FCAVirtu”	54
Figura 5-1– Diagrama de Caso de Uso genérico para um software educacional baseado em documento de requisitos (Apêndice A)	70



## LISTA DE QUADROS

Quadro 2-1 Comparativo de Ambientes Virtuais de Aprendizagem.....	23
Quadro 4-1 Identificação dos pontos variáveis do framework “FCAVirtu” considerando as fases do processo proposto por Braga(2002).....	50
Quadro 4-2 Tabela de Mapeamento dos Padrões / Sub-Padrões da linguagem “LPCAVirtu” em relação às classes do <i>framework</i> .....	56

## SUMÁRIO

<b>CAPÍTULO 1. INTRODUÇÃO .....</b>	<b>10</b>
1.1.Contextualização .....	10
1.2. Motivação .....	11
1.3. Objetivos.....	12
1.3.1.Objetivos Gerais .....	12
1.3.2.Objetivos Específicos .....	12
1.4.Organização da Monografia .....	13
<b>CAPÍTULO 2. EMBASAMENTO TEÓRICO .....</b>	<b>14</b>
2.1.Considerações Iniciais .....	14
2.2.Desenvolvimento de Software Educacional .....	14
2.2.1.Conteúdo.....	16
2.2.2.Apresentação .....	17
2.2.3.Avaliação .....	17
2.2.4.Módulos Educacionais.....	18
2.3.Ambientes Virtuais de Aprendizagem.....	18
2.3.1.AulaNet.....	20
2.3.2.Moodle.....	20
2.3.3.AdaptWeb.....	21
2.3.4.Claroline .....	22
2.4.Processos de Software .....	23
2.5.Padrões de Software .....	25
2.5.1.Linguagens de Padrões .....	26
2.6.Frameworks .....	27
2.6.1.Processo de Desenvolvimento de Framework Baseado em Linguagem de Padrões .....	28
2.7.Considerações Finais .....	32
<b>CAPÍTULO 3. LINGUAGEM DE PADRÕES “LPCAVIRTU” .....</b>	<b>33</b>
3.1.Considerações Iniciais .....	33
3.2.Linguagem de Padrões “CogLearn” .....	33
3.3.Linguagem de Padrões “LPCAVirtu” .....	35
3.3.1.Exemplificação de dois Padrões da Linguagem “LPCAVirtu” .....	37

3.4. Uso da linguagem “LPCAVirtu” para planejar e estruturar uma aula virtual de Engenharia de Software.....	41
3.5. Considerações Finais.....	46
<b>CAPÍTULO 4. ANÁLISE E PROJETO DE UM FRAMEWORK NO DOMÍNIO DE SOFTWARES EDUCACIONAIS.....</b>	<b>47</b>
4.1. Considerações Iniciais.....	47
4.2. Processo de Construção do Framework Caixa-Branca “FCAVirtu”.....	47
4.3. Identificação dos Pontos Variáveis.....	48
4.4. Projeto do Framework.....	50
4.4.1. Validação do Projeto do Framework.....	54
4.5. Estrutura do framework “FCAVirtu”.....	51
4.6. Hierarquia de Classes do “FCAVirtu”.....	52
4.7. Considerações Finais.....	57
<b>CAPÍTULO 5. CONCLUSÕES.....</b>	<b>58</b>
5.1. Considerações Iniciais.....	58
5.2. Resultados Obtidos e Contribuições.....	58
5.3. Limitações do Trabalho.....	59
5.4. Trabalhos Futuros.....	59
<b>REFERÊNCIAS.....</b>	<b>61</b>
<b>APÊNDICE A – DOCUMENTO DE REQUISITOS.....</b>	<b>66</b>
<b>APÊNDICE B – DIAGRAMA DE CASOS DE USO.....</b>	<b>70</b>
<b>APÊNDICE C – PADRÕES CONSTITUINTES DA “LPCAVIRTU”.....</b>	<b>71</b>

# CAPÍTULO 1. INTRODUÇÃO

## 1.1. Contextualização

O desenvolvimento de software educacional é uma atividade de fundamental importância no contexto da informática educativa. A forma como estes programas ou aplicativos são desenvolvidos pode implicar diretamente em sua qualidade e finalidade, atingindo os objetivos a serem alcançados quando de sua utilização no processo de ensino-aprendizagem.

Neste trabalho os softwares educacionais serão representados pela especificação de módulos educacionais, os quais segundo Barbosa (2003) são unidades de estudo constituídas de conteúdos teóricos e práticos que podem englobar atividades instrucionais e avaliações e, cuja disponibilização pode ser efetivada por meio de recursos tecnológicos, em especial fazendo uso da *Web*. Esses módulos educacionais apresentam-se relacionados a ambientes virtuais de aprendizado ou ambientes *e-Learning*, os quais, em sua maioria, fazem uso da *Web* para realizar a disponibilização dos módulos em questão visando a proporcionar novas perspectivas de aprendizado.

De uma forma específica, Kemczinsk (2005) apresenta os ambientes *e-Learning* como sendo sistemas disponíveis na Internet, os quais são capazes de auxiliar no processo de ensino-aprendizagem por meio de atividades – estas realizadas com o apoio das Tecnologias de Informação e Comunicação –, e integração de variadas mídias e recursos.

Segundo Gomes e Wanderley (2003), o desenvolvimento de softwares educacionais não ocorre de forma sistemática, sendo que uma grande quantidade ou a maior parte destas produções são criadas a partir de proposições *ad hoc* de especialistas.

Sch lindwein *et al.* (2005) discorrem acerca da importância desse tipo de software exigir uma base pedagógica que fundamente a sua construção englobando questões advindas dos processos de software.

Tomando por base o cenário descrito é necessário que o desenvolvimento de software educacional considerando o âmbito dos módulos educacionais seja sistematizado, ou seja, siga medidas pré-estabelecidas que caracterizem a sua construção. Além disso, ainda é importante destacar o fato de que, não havendo definições para o desenvolvimento de softwares educacionais, vários modelos de desenvolvimento são adotados divergindo entre si

de caso para caso. Nessa situação, quando um desenvolvedor procura soluções para um determinado problema, em geral, não encontra ou é forçado a desenvolver mecanismos próprios de resolução. Esses mecanismos podem, por hora, resolver a questão tratada, porém, eles não têm base fundamentada ou quando têm não podem evoluir em seus aspectos e variações por não estarem documentados e, conseqüentemente, não compartilhados (CAMPOS, 1998).

Para tentar amenizar essa questão, propõem-se a utilização de padrões que são definidos por Braga (2002) como a descrição da solução de um problema que ocorre constantemente em determinado domínio de software, fazendo uso de experiências já adquiridas em outras implementações. Para tanto, os padrões são reconhecidos como técnica de reuso que permite redução de tempo e custo no desenvolvimento e aumento de qualidade uma vez que se baseiam em experiências já conhecidas e testadas.

Estendendo a abrangência dos padrões, estes, de acordo com Coplien (1998), podem ser agrupados em coleções ou linguagens de padrões as quais seriam uma estruturação de padrões interligados e transformadores de requisitos em arquitetura.

Por sua vez as linguagens de padrões juntamente com os padrões, podem ser utilizados na construção de *frameworks* facilitando sua documentação, entendimento e codificação. Os *frameworks*, de acordo com Johnson e Foote (1991), são um conjunto de classes abstratas que incluem soluções para um domínio específico, propiciando reuso maior do que no caso do uso de classes.

Tendo em vista o desenvolvimento de módulos educacionais na atualidade, a utilização de conceitos de reuso é algo ainda pouco pesquisado para a área, sendo que, alguns resultados são produções isoladas que por vezes tratam apenas de um tipo de aplicação e não de uma categoria em geral.

## **1.2. Motivação**

Quando considerado o fato de que grande parte dos softwares educacionais, aqui limitados aos módulos educacionais de aprendizagem, são propostas isoladas de trabalho, ou seja, voltadas a cada aplicação por meio de um desenvolvimento específico e, na maioria das vezes, não sistematizado, percebe-se a carência existente em seu desenvolvimento.

O desenvolvimento de módulos educacionais quando realizado por meio de modelos pré-estabelecidos, assim como ocorre em outros domínios de aplicação, favorece-lhe a

possibilidade de se adequar, de certa forma, a qualquer tipo de conteúdo a ser abordado e, quando não, ao menos podem ser tomadas por base certas variações que o compõem.

Atualmente, a atividade de desenvolvimento de softwares educacionais é considerada quase como artesanal, uma vez que cada desenvolvedor cria sua aplicação educativa fazendo uso de proposições de sua própria experiência. Isso nem sempre contribui para o desenvolvimento de um bom software e ainda, em se tratando de módulos educacionais, os quais exigem requisitos específicos e fundamentais para seu sucesso na composição de aulas virtuais, um bom processo de desenvolvimento pode vir a lhe conferir a qualidade esperada.

De posse dessas afirmações monta-se um cenário de motivação para realização desta pesquisa que busca propor melhorias no contexto do desenvolvimento de softwares educacionais considerando, especificamente, a construção de módulos educacionais, estes necessários para compor aulas virtuais e constituintes da estrutura pressuposta para a implantação do *e-Learning*.

### **1.3. Objetivos**

Os objetivos do trabalho são elencados nas subseções a seguir e estão classificados em objetivos gerais e específicos.

#### **1.3.1. Objetivos Gerais**

O trabalho terá como objetivo principal evidenciar a importância da sistematização do desenvolvimento módulos educacionais, dada a sua importante finalidade na construção de aulas virtuais, por meio da utilização de conceitos pertinentes à Engenharia de Software tais como linguagem de padrões e *frameworks*, uma vez considerada a motivação já tratada.

#### **1.3.2. Objetivos Específicos**

São considerados objetivos específicos desta pesquisa os seguintes itens:

- Conhecer e apresentar as características básicas abrangidas pelo domínio educacional gerando um documento de requisitos;

- Criar ou utilizar, quando de sua existência, uma linguagem de padrões baseada nos requisitos elicitados anteriormente a fim de buscar padronização para o desenvolvimento de módulos educacionais;
- Utilizar um processo de desenvolvimento de *framework* baseado em linguagem de padrões para apoiar o projeto de um *framework* no domínio educacional;
- Projetar um *framework*, ou parte deste, a partir da especificação de uma linguagem de padrões baseada no domínio educacional com o intuito de facilitar a atividade de construção de módulos educacionais e sua instanciação;
- Pesquisar, avaliar e fundamentar idéias que possam concluir a questão presente no objetivo geral deste trabalho.

#### 1.4. Organização da Monografia

A monografia em questão subdivide-se em quatro capítulos, além dos quais deve-se considerar como capítulo inicial a Introdução – Contextualização, Motivação, Objetivos e Referências.

No Capítulo 2, tratam-se os conceitos específicos que abrangem o tema discutido. Nesse capítulo são tratadas questões tais como a do Desenvolvimento de Software Educacional, Ambientes Virtuais de Aprendizagem, Processos de Software, Padrões de Software, Linguagens de Padrões e *Frameworks*. De forma mais específica, na Seção 2.6, será mencionado um processo de desenvolvimento de *frameworks* o qual será utilizado posteriormente para alcançar os objetivos estabelecidos. O Capítulo 3 é um dos capítulos de maior importância desta pesquisa. É apresentada uma linguagem de padrões, a “*CogLearn*”, específica para o domínio de ambientes virtuais de aprendizado por meio do desenvolvimento de materiais instrucionais para ambiente *Web*, da qual extraiu-se a idéia central para o desenvolvimento de uma linguagem particular a esta pesquisa, a “*LPCAVirtu*”, sendo que esta evoluiu da linguagem citada com acréscimo e adaptação de padrões.

A continuidade do capítulo anterior e a finalização da idéia central deste trabalho estão presentes no Capítulo 4, o qual trata da análise e projeto de um *framework* para o domínio educacional, o “*FCAVirtu*”, baseado na linguagem de padrões “*LPCAVirtu*”. Por fim, no quinto e último capítulo é constituído da conclusão da pesquisa realizada, considerando os objetivos definidos, as limitações e trabalhos futuros.

## **CAPÍTULO 2. EMBASAMENTO TEÓRICO**

### **2.1. Considerações Iniciais**

Os tópicos a serem tratados nas seções deste capítulo referem-se ao embasamento teórico necessário para o entendimento e desenvolvimento deste trabalho. Na Seção 2.2 são apresentados o contexto do desenvolvimento de softwares educacionais, considerando os módulos educacionais, e os elementos principais que compõem esse desenvolvimento sendo o conteúdo, apresentação e avaliação. Considerando a Seção 2.3 são apresentados conceitos de Ambientes Virtuais de Aprendizagem e exemplos de ferramentas de geração destes ambientes. Na Seção 2.4 são definidos e caracterizados os processos de software. As questões referentes a padrões e linguagens de padrões de software estão tratadas na Seção 2.5 e, em seqüência, na Seção 2.6 é discutida a questão do reúso de software através da utilização de *frameworks* e de um processo específico para seu desenvolvimento.

### **2.2. Desenvolvimento de Software Educacional**

De acordo com Gomes e Wanderley (2003), o software educacional deve ser entendido como uma classe de interfaces educativas ou um conjunto de artefatos cujo objetivo seja atuar como mediadores em atividades educativas de formação em áreas distintas do conhecimento. A função básica de um software educacional está fundamentada na promoção da aprendizagem para o uso, além da análise da aprendizagem de conceitos específicos que ocorrem com o uso do software, não se tratando apenas de aprender a fazer algo com a interface, mas sim de trabalhar com ela no intuito de aprender algum conceito.

Desenvolver softwares educacionais, limitando-os aos módulos educacionais, na atualidade tem sido um desafio para engenheiros de software e programadores. Segundo Mandel ((1997) apud Gomes e Wanderley (2003)), um dos motivos que dificultam a produção de software educativo pode estar ligado ao fato de existir uma diferença significativa presente no processo de concepção, constituída das representações que projetistas, programadores e professores têm acerca dos processos de ensino-aprendizagem.

Para Campos (1998), o método de desenvolvimento de software educacional no qual programadores trabalhavam sozinhos para a conclusão do produto de software educacional



está se extinguindo. Torna-se necessário recorrer a novos métodos e ferramentas a fim de alcançar a qualidade de software no contexto em questão. Esses softwares devem englobar objetivos educacionais pré-propostos e possuírem um ambiente de aprendizagem no qual seja permitida a criação de situações de desenvolvimento de habilidades buscadas pelos objetivos educacionais.

Com o desenvolvimento dos recursos tecnológicos na área da informática, sobretudo no desenvolvimento de software, nota-se uma crescente utilização dos softwares educacionais, uma vez que sua produção, evidenciada por Cantarelli (2004), pôde contar com avanços tais como a hipermídia<sup>1</sup>, as linguagens de programação orientadas a objeto e a utilização de ferramentas sofisticadas de elaboração de multimídias, tais como softwares de criação e hardware que auxilia este fim (câmeras, scanners, tecnologia laser).

Em conformidade com estes avanços, considerando-se a mesma autora, são elicitadas etapas de desenvolvimento de softwares educacionais a fim de organizá-lo genericamente:

1. Definição dos objetivos instrucionais;
2. Determinação das características do público alvo;
3. Escolha do modelo instrucional a ser adotado;
4. Escolha das estratégias educacionais;
5. Interfaces de comunicação e sua implementação.

Ainda em se tratando de etapas para o desenvolvimento de softwares educacionais, Santos (1999) sugere um modelo um pouco diferenciado do anteriormente citado constituído por fases um pouco mais detalhadas e objetivas baseadas nas fases genéricas de desenvolvimento de softwares, relevando aspectos de projeto tais como o tema, o ambiente de aprendizagem, plataforma, avaliação e validação:

1. Definição do tema a ser abordado no software educacional;
2. Identificação dos objetivos educacionais da aplicação e do público-alvo;
3. Definição do ambiente de aprendizagem;
4. Modelagem da aplicação;
5. Planejamento da interface;

---

<sup>1</sup> Fusão de várias mídias num suporte computacional a partir de meios não lineares, apoiada por sistemas eletrônicos de comunicação (Cantarelli, 2004).

6. Seleção de plataforma de *hardware* e *software*;
7. Implementação;
8. Avaliação do software educacional;
9. Validação.

Seguindo os modelos acima propostos e no contexto dos softwares educativos são considerados três grandes grupos de requisitos: conteúdo, apresentação e avaliação, descritas nas subseções a seguir, as quais, segundo Cantarelli (2004), são os elementos responsáveis pela estruturação da aplicação educativa.

### **2.2.1. Conteúdo**

A seleção do conteúdo é um dos problemas recorrentes em educação que se baseia no planejamento por parte do professor de “como”, “o que”, e “para que” ensinar. Para Cataldi (2000), o professor é o ator principal na caracterização do conteúdo a ser abrangido por um software educacional, é ele que determinará para a equipe de desenvolvimento do software educacional “o que ensinar”, apontando a forma de apresentação, delimitação e subdivisão desse conteúdo seguindo os preceitos pedagógicos.

Por ser o conteúdo um dos três grupos de requisitos fundamentais no desenvolvimento de softwares educacionais – como já citado anteriormente – é importante que as ações de gerenciamento de conteúdos possam ser contempladas no software, isso pode garantir uma maior flexibilidade para o professor e uma maior adaptabilidade ao próprio software, que desta forma, poderá abordar os mais diversos conteúdos.

Vahl (2000), em seu trabalho de avaliação de modelos computacionais para unidades educacionais, cita uma classificação quanto à possibilidade de modificação de conteúdos em software educacionais:

- Softwares educacionais fechados: o conteúdo fornecido previamente com o software não pode ser alterado; e,
- Softwares educacionais abertos: são baseados em uma estrutura básica que pode suportar os mais diversos conteúdos, os quais podem ser incluídos pelos próprios usuários facilitando assim a adequação dos mesmos a diversos contextos.

Tendo em vista a caracterização anteriormente descrita, de acordo com o mesmo autor, a segunda classificação de software educacional mostra-se como a opção mais indicada para o futuro da produção dos softwares educacionais.

### **2.2.2. Apresentação**

A apresentação do conteúdo em softwares educacionais desenvolvidos na atualidade, de acordo com Vahl (2000), se efetiva por meio de interfaces gráficas, que representam todas as possíveis formas de interação do aluno com o software por meio de botões, janelas, entre outros elementos.

A área da apresentação de conteúdo em softwares educacionais para Fenner (2000) exige a função de desenvolver uma linguagem visual composta de símbolos e ícones que seja compatível com o conhecimento do aluno e com o conteúdo a ser abordado. Deve se apresentar da forma mais simples possível simulando idéia de programas com os quais este usuário já mantém contato. Em suma, a área de trabalho deverá ser elaborada visando a viabilidade de uso de recursos visuais e instrumentos familiares ao aluno.

Considerando outros aspectos, uma apresentação de conteúdo que possa contar com recursos que reproduzam ou simulem o mundo real, sem dúvida pode gerar uma melhor obtenção de resultados. Conforme os desenvolvedores do projeto MAC-Multimídia (1998) citados por Fischer (2001), a apresentação de uma interface que possa conter som, imagem, vídeo e animação de soluções para problemas propostos, facilita o aprendizado e motiva o aluno a ampliar o conhecimento adquirido em aula.

### **2.2.3. Avaliação**

Tomando por base Leiva (2003), a área da avaliação dentro do desenvolvimento de softwares educacionais contempla os diferentes tipos de elementos de que se pode dispor na obtenção de um indicador quantitativo e qualitativo do conhecimento adquirido pelo aluno durante a utilização do software, além de avaliar a efetividade do processo de ensino-aprendizagem a fim de verificar o alcance dos objetivos pedagógicos.

O mesmo autor indica uma classificação de avaliação em três tipos, comentados a seguir:

- Avaliação diagnóstica: identifica o conhecimento já adquirido pelo aluno em relação a certas atividades de aprendizagem, determinando o seu nível de conhecimento inicial além de poder evidenciar possíveis dificuldades de aprendizagem;
- Avaliação formativa: baseia-se na apresentação inicial de conteúdos ao aluno por meio dos mais diversos recursos e na avaliação paralela dos conhecimentos adquiridos por eles durante a sua navegação pelo sistema;
- Avaliação somativa: também conhecida como “avaliação final”, esse tipo de avaliação ocorre no módulo final de utilização do sistema com o intuito de fornecer ao professor, e até mesmo ao próprio aluno, os níveis de aprendizado alcançados durante o uso do software.

#### **2.2.4. Módulos Educacionais**

Como já apresentado na Seção 1.1, os módulos educacionais são definidos por Barbosa (2003) como sendo unidades de estudo compostas por conteúdos de todos os tipos que podem se relacionar a atividades e avaliações. Estes módulos podem ser disponibilizados fazendo uso de vários recursos tecnológicos. Atualmente, com o rápido crescimento e expansão das redes de computadores e com a necessidade cada vez maior dos profissionais em realizar cursos à distância, a divulgação dos módulos educacionais ocorre frequentemente por meio de recursos da *Web*, nos quais, em geral, esses módulos são encontrados compondo aulas virtuais.

Ainda considerado Barbosa (2003), pode-se afirmar que a produção de módulos educacionais se assemelha ao processo de desenvolvimento de softwares em geral, porém, requer a utilização de processos melhor estruturados e aplicação de determinados conceitos pedagógicos que não a tornam uma tarefa tão simples.

Para efeito desta pesquisa, o estudo de módulos educacionais é feito com base em exemplos de ambientes virtuais de aprendizado que podem gerar módulos educacionais ou aulas virtuais.

### **2.3. Ambientes Virtuais de Aprendizagem**

Segundo Santos (2003), a Internet é contemporaneamente o maior meio de comunicação mundial. Sua utilização tem se estendido para os mais diversos domínios. No

contexto educacional, cita-se a utilização da Internet por meio dos ambientes virtuais de aprendizagem – expressão muito utilizada nos dias atuais por educadores, técnicos em informática e outros profissionais, cujo interesse esteja voltado para a utilização e desenvolvimento de módulos educacionais com mediação tecnológica, de forma específica pelas relações entre humanos e redes telemáticas de informação e comunicação.

Luciano e Vieira (2002) apresentam os ambientes virtuais de aprendizagem como sendo cenários que envolvem módulos instrucionais para a interação de aprendizes, incluindo ferramentas para atuação autônoma e automonitorada, oferecendo recursos pela aprendizagem coletiva e individual.

Segundo o mesmo trabalho estes ambientes devem ter seu foco na aprendizagem e oferecerem situações para que os alunos registrem informações definindo sua caminhada na busca de novas idéias e descobertas.

Santos (2003) afirma que um ambiente virtual é um espaço fecundo de significação em que seres humanos e objetos técnicos interagem de forma a evidenciar a construção de conhecimentos, ou seja, a aprendizagem. Para tanto, todo ambiente virtual é um ambiente de aprendizagem.

Para Oliveira (2001) os ambientes virtuais ajudam na autonomia do aluno desenvolvendo sua criatividade e emancipação; porém, existe, entre muitos educadores, resistência ao uso das tecnologias de informática na educação, incluindo o uso de ambientes virtuais, por diversas causas, sendo que a principal delas é a ausência de contato com esses meios de aprendizagem que leva esses educadores a opiniões equivocadas quanto à sua utilização.

Embora sejam variados os motivos da não aceitação da tecnologia, de acordo com Costa e Franco (2005), o desenvolvimento de ambientes virtuais de aprendizagem tem se tornado uma prática constante nas pesquisas em educação; ainda, a utilização desses ambientes não exige dos professores extenso domínio em conhecimentos de informática, sendo necessárias apenas algumas horas de treinamento no uso do ambiente.

Para o contexto desta pesquisa, os ambientes virtuais de aprendizado constituem o meio de implantação dos módulos educacionais, sendo que, para assim o fazer existem diversos ambientes virtuais de aprendizagem na Internet, dentre esses, vários abrangem as mais distintas aplicabilidades possíveis. Por essa diversidade foram escolhidos quatro ambientes de destaque, os quais caracterizam-se por serem instanciadores de aulas virtuais para *Web*, para serem apresentados e comparados (por meio do Quadro 2.1) como referência neste trabalho, são eles: AulaNet, Moodle, AdaptWeb e Claroline.

### 2.3.1. AulaNet

O AulaNet é descrito por Lucena *et al.* (2005) como um ambiente de ensino-aprendizagem na Web baseado no modelo de colaboração W3C<sup>1</sup>. Seu projeto iniciou-se em junho de 1997 no Laboratório de Engenharia de Software do Departamento de Informática da PUC – Rio, e sua distribuição é feita pela EduWeb<sup>2</sup>.

Os serviços do AulaNet estão classificados em comunicação, coordenação e cooperação. Sua concepção foi realizada de forma a separar o conteúdo da navegação. A navegação dos alunos pelos conteúdos produzidos é tratada pelo ambiente enquanto que os professores se encarregam da produção dos conteúdos didáticos (Lucena, 2003).

Fuks *et al.* (2005) comentam que os serviços AulaNet podem ser vistos como componentes que são plugados ou desplugados conforme a necessidade do grupo. Um *framework* de componentes denominado *Groupware Component Framework* fornece um conjunto de interfaces que devem ser implementadas pelos componentes para que estes possam ser plugados no *framework* e interagir com outros componentes, compondo desta forma a arquitetura deste ambiente.

Os mecanismos oferecidos pelo AulaNet constituem-se de mensagem aos docentes, grupo de discussão, grupo de interesse, debate, contato com os participantes, avisos, plano de aulas, tarefas, avaliação, relatórios de participação, referências, documentação e *download* (Fuks, 2000).

### 2.3.2. Moodle

Neto (2004) apresenta o Moodle como um software para produzir e gerenciar atividades educacionais baseadas na Internet ou em redes locais. Trata-se, ainda, de um projeto de desenvolvimento contínuo projetado para apoiar o social-construtivismo educacional que conjuga um sistema de administração de atividades educacionais com um pacote de software desenvolvido para auxiliar os educadores a obter um alto padrão de qualidade nas atividades educacionais *on-line* criadas. Ainda considerando o mesmo autor, a

---

<sup>1</sup> World Wide Web Consortium – consórcio de empresas fundado para levar a Web ao seu potencial máximo por meio do desenvolvimento de protocolos comuns que promovem sua evolução e interoperabilidade.

<sup>2</sup> Empresa de distribuição de software educacional inicialmente incubada na PUC-Rio por meio do LES (Laboratório de Engenharia de Software) desta mesma instituição.

palavra Moodle provém do termo “*Modular Object-Oriented Dynamic Learning Environment*”.

Em afirmação de Comassetto (2006) tem-se que o projeto e desenvolvimento do Moodle são guiados por uma filosofia de aprendizagem especial, denominada “pedagogia social construcionista”, abordagem esta embasada em quatro conceitos principais da pedagogia sendo: construtivismo, construcionismo, construtivismo social e comportamento conectado e separado.

O conceito geral de funcionamento desse ambiente comentado por Fernandes (2005) apóia-se no fato de consistir numa página, em que os professores disponibilizam recursos e desenvolvem atividades com e para os alunos. A cada usuário registrado está associado um perfil (em geral com foto) que possibilita comunicação com outros usuários. Como sendo um sistema de gestão de ensino-aprendizagem apresenta funcionalidades com forte componente de participação e comunicação entre usuários dos mais diversos perfis, sem contar o componente de avaliação (representado pela atribuição de tarefas e testes durante o processo de aprendizado nas aulas) que é um dos mais importantes perfis em qualquer ambiente semelhante. As funcionalidades principais englobam: fórum, trabalho (desenvolvimento de pesquisas, relatórios, etc), *chat*, diálogo, glossário, aula, teste (forma de avaliação realizada ao final de cada leitura), questionário e *wiki* (estilo de funcionamento do Wikipedia®).

### **2.3.3. AdaptWeb**

O projeto AdaptWeb é um ambiente hipermídia de ensino à distância adaptativo para *Web*, capaz de fazer uso de conceitos de adaptabilidade com o intuito de disponibilizar um mesmo conteúdo para aprendizes de diferentes grupos (Viero, 2003).

O estudo desse ambiente foi motivado pela necessidade de facilitar a tarefa de autoria de material instrucional para *Web* por parte do professor e pelo fato do interesse dos alunos ser diferente, de acordo com a natureza de seus cursos gerando carência de uma abordagem diferenciada. Os resultados constituem um sistema hipermídia adaptativo de Educação à Distância (EAD) para *Web*, cujo conteúdo educacional e o percurso de navegação podem ser modificados com o objetivo de sanar as necessidades de cada aluno (Oliveira, 2003).

Segundo o mesmo autor, a proporção dos aspectos de adaptabilidade de um Sistema Hipermídia Adaptativo<sup>1</sup> requer uma separação das funções dos componentes envolvidos. Nesse âmbito, o ambiente AdaptWeb conta com quatro componentes: autoria, armazenamento em XML, adaptação de conteúdo baseada no modelo do aluno e interface adaptativa.

Para Marçal (2003) acredita-se que a maneira personalizada de apresentar uma instrução possa proporcionar a individualização na aprendizagem, estímulo e motivação para o aluno, e este ambiente é capaz de agregar todas essas ocorrências.

### 2.3.4. Claroline

Em sua apresentação de ambientes virtuais de aprendizagem Malange (2005) descreve Claroline como sendo um sistema desenvolvido pela Universidade Católica de Louvain (UCL – Bélgica) e distribuído por meio de GPL – *Open Source License*<sup>2</sup>. Trata-se de um Sistema de Gerenciamento de Cursos baseado na *Web* que possibilita aos professores criar e administrar cursos para seus alunos na Internet fazendo uso de qualquer navegador, sem a exigência de vastos conhecimentos de informática.

A apresentação de conteúdos nesse ambiente pode constituir-se de hipertextos, apresentações, textos de aulas, vídeos e imagens. A interatividade aluno-professor se efetiva por mecanismos de comunicação como: contato com o professor (*e-mail*), grupo de interesse, grupo de discussão, debate (*chat*).

Sisson (2003) cita como funcionalidades principais do Claroline a descrição de curso (capacidade de criar vários cursos no ambiente), gerenciador de arquivos (sessão para armazenamento de arquivos tais como programas e apresentações), agenda, anúncios, *links*, exercícios, fórum, gerenciamento de usuários e trabalhos de alunos.

Malange (2005) afirma que um dos recursos mais notáveis no Claroline, que talvez seja escasso em outros ambientes, é a escalabilidade que o permite funcionar em qualquer tipo de máquina (desde *palm*s até *mainframes*). Além disso, um outro aspecto importante desse ambiente é o fato do material didático a ser disponibilizado poder ser produzido de acordo com o projeto pedagógico mais adequado.

---

<sup>1</sup> Sistema no qual o objetivo principal é a adaptação de conteúdos e recursos computacionais às necessidades do usuário aumentando, desta forma, a funcionalidade da hipermídia e tornando-a personalizada. (Oliveira,2003)

<sup>2</sup> Designação de licença para software livre baseada em quatro liberdades: execução do programa, estudo de como o programa funciona, redistribuição de cópias e aperfeiçoamento do programa com liberação de novas versões.



<b>Características</b>	<b>Moodle</b>	<b>Claroline</b>	<b>AulaNet</b>	<b>AdaptWeb</b>
<b>Ferramentas de Comunicação, Coordenação e de Cooperação</b>	Fórum de Discussão, Atividades, Busca, Últimas Notícias	Correio Eletrônico, Lista de Discussão, Grupo de Discussão, Fórum, Chat, Descrição do Curso, Agenda, Anúncios, Documentos, <i>Links</i> , Trabalhos do Aluno, Exercícios	Contato com o Professor, Grupo de Interesse, Grupo de Discussão, Debate, Agenda, Notícias do Curso, Provas, Trabalhos, Exercícios, Transparências, Apresentação Gravada, Texto de Aula, Livro Texto, Bibliografia, Demonstração, Download, Tutorial	Curso / Disciplina, Filtro de Mídias, Interface Adaptativa, Modelo do Aluno, Modelo do Professor, Adaptação de Conteúdo, Conceito, Exemplo, Material Complementar
<b>Formas de Avaliação</b>	Avaliação do Curso, Questionário, Tarefa, Trabalho com Revisão	Análise Estatística, Participação em Chats e Fóruns, Análise dos Textos, Exercícios Propostos	Trabalhos, Participação em Chats-Listas-News, Provas	Exercício, Teste
<b>Considerações</b>	Software de implementação livre e código aberto funcionando em qualquer sistema que suporte PHP	Implementação tanto com ferramentas de desenvolvimento e execução GLP como com sistemas proprietários	Baseado em Sistema Operacional Proprietário (Windows)	Implementação e desenvolvimento regida pelo GNU com plataforma aberta

**Quadro 2-1 Comparativo de Ambientes Virtuais de Aprendizagem**

## 2.4. Processos de Software

Segundo definição apresentada por Maia (2004), processos de software podem ser entendidos como o conjunto de atividades a serem realizadas desde a concepção até a implantação do produto de software, sendo definidos formalmente por meio de modelos de processo, estes enumerando a seqüência de atividades que devem ser realizadas, as ferramentas a serem utilizadas, os produtos e documentos de software que serão criados e utilizados e os papéis dos envolvidos no desenvolvimento do software.

Tomando-se as idéias de Reis (2003), o processo de software pode ser constituído de atividades (análise e definição de requisitos, projeto arquitetural e detalhado, codificação, teste de unidade, de integração e de sistema, lançamento e implantação), as quais são categorizadas como essenciais à construção de um software.

Ainda considerando Reis (2003), com base nas atividades essenciais que constituem o processo de software alguns autores sugerem modelos para descrever o ciclo de vida de um projeto de construção de software, os quais podem ser denominados seqüenciais (como o modelo cascata) ou evolucionários / iterativos (como a prototipação).

Para Pressman (1994), a definição do modelo cascata, paradigma mais antigo e mais utilizado na Engenharia de Software, é dada por uma abordagem seqüencial e sistemática de desenvolvimento que engloba as atividades de engenharia de sistemas, análise, projeto, codificação, teste e manutenção.

Já quanto ao modelo de processo por prototipação, o autor descreve como sendo um processo que capacita o desenvolvedor a criar um protótipo do software a ser implementado. Esse protótipo pode ser em papel, de trabalho implementando um subconjunto de funções exigidas ou um programa já existente com outros aspectos os quais serão melhorados no decorrer do desenvolvimento.

Também merece citação o modelo de processo de desenvolvimento conhecido como incremental, o qual está sendo amplamente utilizado em diversos domínios de aplicação. Segundo Sommerville (2003), trata-se de um modelo intermediário que combina vantagens dos dois modelos anteriormente citados. Sua abordagem visa a um meio de reduzir o “retrabalho” gerado nas fases anteriores no processo de desenvolvimento sendo que a implementação trabalha as funções do sistema por meio da prioridade a elas atribuída.

Neste contexto, Reis (2003) cita considerações importantes sendo que o processo de software particular é algo concreto e individualizado quando implementado por uma organização, ou seja, cada equipe de desenvolvimento cria sua própria versão.

O processo de software utilizado no desenvolvimento de um produto de software gera forte dependência na sua qualidade, por isso, é necessário conhecer como os produtos são planejados e produzidos, isto é, o seu processo, requisito básico para o alcance da qualidade do software (Bertollo, 2003).

Para o mesmo autor, processos de software devem ser definidos caso a caso, nos quais se devem considerar aspectos tais como a adequação às tecnologias envolvidas, tipo de software, domínio de aplicação, grau de maturidade da equipe, características da organização e do projeto.

Em questão de qualidade de software com a utilização de processos Fuggetta (2000) ressalta que a qualidade dos produtos de software está estritamente ligada à qualidade do processo utilizado sendo que a qualidade de ambos é importante na garantia de que o software

atenda às necessidades dos seus usuários. Neste trabalho foi utilizado um processo de desenvolvimento de *framework* que será apresentado na Seção 2.6.1.

## 2.5. Padrões de Software

A maioria das pessoas desenvolve idéias por meio da reutilização de outras idéias já pensadas anteriormente. Essas idéias podem levar a técnicas que se desenvolvem gradativamente com o decorrer do tempo fazendo com que áreas como a indústria de software tenha que documentá-las de acordo com sua necessidade (Yoder, 2002).

O reúso de software, conceito da engenharia de software atualmente muito difundido, tem o objetivo de realizar o reaproveitamento de partes do software para o desenvolvimento de outros projetos a fim de alcançar diminuição de custos, aumento de produtividade e evolução do próprio produto. Nesse contexto, pesquisa-se o conceito de padrões de software, os quais são considerados formas de reúso que utilizam aspectos de orientação a objetos (Braga *et al.*, 2001).

Segundo Maldonado *et al.* (2004), um padrão descreve uma solução possível de ocorrer com frequência no desenvolvimento de um software, ou ainda, é um conjunto de instruções que possui um nome e que capta a estrutura essencial e o raciocínio de um grupo de soluções comprovadamente bem sucedidas para um problema recorrente de determinado contexto e repercussão.

Cruz (1998) cita que apesar dos diferentes formatos utilizados na representação de padrões, existem elementos fundamentais que devem existir consensualmente em qualquer documentação de padrões, os quais estão descritos a seguir:

- **Nome:** denominação significativa e sucinta do padrão;
- **Problema:** explanação do problema a ser resolvido pelo padrão;
- **Contexto:** aplicabilidade do padrão, condições sobre as quais o padrão deve atuar sobre o problema;
- **Solução:** resolução para o problema de determinado contexto;
- **Exemplos:** cenários de funcionamento do padrão;
- **Contexto resultante:** pós-aplicação do padrão considerando estado do sistema e conseqüências.

Pelo fato de existirem diversos padrões de diversos contextos (de análise, de projeto, arquitetural, de implementação, entre outros) e, em geral, não existirem padrões isolados, ou

seja, um padrão pode depender de outro, Maldonado *et al.* (2004), apresentam a importância de agrupar os padrões existentes por meio de algum critério a fim de facilitar seu reuso. Com esse aspecto elencam-se as linguagens de padrões cuja explanação segue.

### 2.5.1. Linguagens de Padrões

Coplien (1998) define linguagem de padrões como uma coleção estruturada de padrões que se apóiam uns nos outros para transformar requisitos e restrições numa arquitetura. Ou ainda, é uma forma de subdividir um problema geral e sua solução complexa em determinada quantidade de problemas mais simples e suas soluções. De uma forma mais específica temos que linguagens de padrões de análise são linguagens constituídas de padrões a serem aplicados durante a fase de análise de sistemas de determinado domínio servindo de base para o desenvolvimento de *frameworks* para o desenvolvimento de sistemas implementados com essas linguagens (Braga, 2002).

Uma linguagem de padrões considera a inclusão de diretrizes que explicam a aplicabilidade de seus padrões na resolução de um problema. Ainda, pode fornecer soluções completas para um conjunto de problemas reiterados, requerendo evolução periódica para que haja atualização dessas soluções (Cagnin, 2005).

Neste trabalho o uso de linguagem de padrões estará focado no contexto da construção de módulos educacionais, sob o qual nota-se carência na utilização de padrões e de linguagens de padrões no processo de desenvolvimento de *frameworks*. Essa carência é preocupante principalmente ao considerar-se que o uso de padrões no desenvolvimento de software, fazendo uso de *frameworks* neles baseados por meio de linguagens de padrões, pode garantir determinada qualidade do produto de software desenvolvido e também contribuir de forma significativa na redução de tempo e custo associados ao desenvolvimento.

Em meio às pesquisas realizadas encontra-se como exemplo de uma linguagem de padrões para *e-Learning*, uma linguagem desenvolvida e descrita por Neto (2005) com a finalidade de documentar a prática de desenvolvimento de material instrucional para cursos de Educação à Distância.

A linguagem, denominada “*CogLearn*”, tem por objetivos prover padrões que orientem os professores na elaboração das aulas (no aspecto do conteúdo a ser abordado), auxiliem na concepção de um projeto para elaborar a tomada de decisão dos alunos quando do uso do ambiente que provê essas aulas – considerando a possibilidade destes determinarem o

ritmo em que querem aprender – e durante a realização do curso por meio de estratégias de acesso, além de que, abranjam questões de navegação e *layout*.

Esta linguagem será mais bem apresentada em momento futuro quando do desenvolvimento da linguagem de padrões deste trabalho, sendo que existe a possibilidade de utilização de padrões oriundos dessa linguagem no desenvolvimento da linguagem de padrões aqui proposta.

## 2.6. Frameworks

A utilização de *frameworks* está diretamente associada à questão do reuso de software. Para Johnson e Foote (1991), *framework* pode ser definido como um projeto de um grupo de objetos que colaboram entre si para a execução de um conjunto de responsabilidades, reutilizando análise, projeto e código. Análise por descrever os tipos de objetos importantes e como problemas maiores podem ser minimizados; projeto por conter algoritmos abstratos e descrever a interface a ser implementada e as restrições impostas pela implementação, código pelo fato deste tornar mais fácil o desenvolvimento de uma biblioteca de componentes compatíveis.

Segundo Fayad e Johnson (2000) *frameworks* são mais do que projeto ou arquitetura, tratam-se de uma aplicação semicompleta contendo componentes estáticos e dinâmicos que podem ser adaptados para produzir aplicações específicas do usuário. Ainda, consideram existir, de acordo com a forma de reuso, três tipos de *frameworks*: caixa branca, caixa preta e caixa cinza. No *framework* caixa branca, o reuso é provido por herança, ou seja, o usuário deve criar subclasses das classes abstratas contidas no *framework*. Quanto ao *framework* caixa preta, o reuso é por composição, isto é, o usuário combina diversas classes concretas existentes no *framework*. Já o *framework* caixa cinza consiste da mescla entre os *frameworks* caixa branca e caixa preta sendo que o reuso é obtido por herança, por ligação dinâmica e por interfaces de definição.

Braga (2002) sumariza que um *framework* caixa branca é mais fácil de projetar pelo fato de não ser necessário prever todas as alternativas de implementação, o que ocorre no *framework* caixa preta. No entanto, este último é mais fácil de usar, pois basta escolher a implementação desejada, ao invés de fornecer a implementação completa, como é o caso anterior. Desta forma, os *frameworks* caixa cinza tiram proveito das vantagens dos dois anteriores, sendo assim, um meio termo.

Na visão da mesma autora, a presença de *frameworks* influencia o processo de desenvolvimento de software considerando-se que este não necessita mais ser uma atividade totalmente “artesanal” partindo do “marco zero”; porém, esta praticidade deve levar em conta os processos de manutenção do *framework* o qual não é uma tarefa tão simples quanto possa parecer.

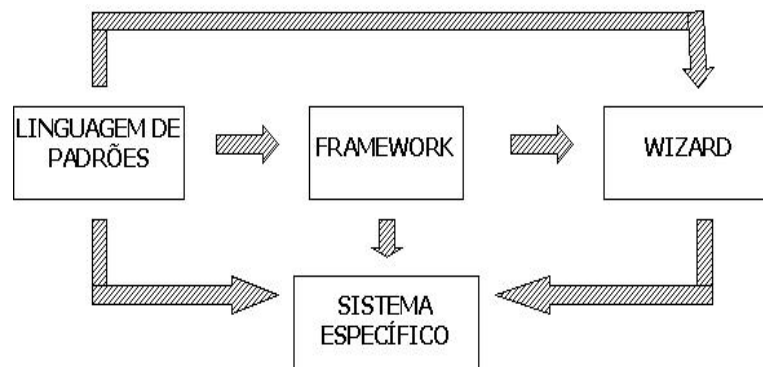
No que tange ao desenvolvimento de softwares educacionais, com base nas pesquisas realizadas, os *frameworks* ainda estão em fase de experimentação e pouca é a sua efetivação neste domínio. Algumas aplicações na área de softwares educacionais adaptativos podem ser destacadas como é o caso do trabalho de Primus (2005) que aborda o desenvolvimento de um *framework* para a avaliação de sistemas de autoria de hipermídia educacional adaptativa com base em três critérios de avaliação sendo estes: tecnológico, educacional / conceitual e relato do usuário final.

### **2.6.1. Processo de Desenvolvimento de *Framework* Baseado em Linguagem de Padrões**

Dada a classificação de *frameworks* na Seção 2.6, tem-se que o seu desenvolvimento baseado em linguagens de padrões de análise – nos quais segundo Cagnin (2005) os sistemas gerados podem herdar padrões de análise e projeto, código fonte na linguagem de implementação de origem, meios de armazenamento, entre outros – pode seguir um processo. Esse processo, definido por Braga (2002), corresponde a quatro passos: construção de uma linguagem de padrões, construção de um *framework* caixa-branca baseado na linguagem de padrões desenvolvida, construção de uma ferramenta (*Wizard*) que instancia automaticamente aplicações do domínio e instanciação do *framework* construído.

Um detalhamento geral do processo de desenvolvimento proposto, conforme Figura 2.1, consiste no fato de construir uma linguagem de padrões de análise, a qual servirá de base para o projeto do *framework*. Na hierarquia de classes do *framework* são implementadas funcionalidades presentes na linguagem de padrões de forma a oferecer opções *default* e variantes ou *hot spots* - que são classes ou métodos abstratos implementados quando da instanciação do sistema -, de acordo com os requisitos nela especificados (Cagnin, 2005). Os *frameworks* desenvolvidos são do tipo caixa-branca pelo fato de ser necessário, para se derivar aplicações, criar subclasses que herdem características de classes abstratas. As classes definidas no escopo dos padrões relacionam-se com classes do *framework* proporcionando

facilidade na instanciação da aplicação. Para favorecer a instanciação por meio do uso de uma forma gráfica e automática uma ferramenta do tipo *Wizard* pode ser desenvolvida.



**Figura 2-1 Ilustração da abordagem proposta pelo processo de desenvolvimento de *framework* baseado em linguagem de padrões de Braga (2002) (Adaptado de Braga (2002)).**

Existe iteratividade no processo proposto por Braga (2002), sendo que pode haver retorno de informações durante a sua execução a fim de alcançar maior detalhamento de atividades dentro de cada uma das fases.

A seguir serão apresentados, de maneira objetiva e sucinta, os quatro passos propostos para o processo definido:

- ***Desenvolvimento da Linguagem de Padrões:*** O passo de construção da linguagem de padrões pode ser dividido em sub-passos, cada qual com seus respectivos requisitos de complexidade e atividade. Destacam-se a Análise de Domínio ou Engenharia Reversa, Determinação dos Padrões, Criação do Grafo de Fluxo de Aplicação dos Padrões, Escrita dos Padrões e Validação da Linguagem de Padrões. Cada um desses sub-passos será rapidamente explanado a seguir:
  - *Análise de Domínio ou Engenharia Reversa:* para caracterizar o domínio com o qual a linguagem será desenvolvida se faz necessário obter informações referentes a este. Para que isto seja feito duas alternativas são propostas: a realização da Análise do Domínio ou a Engenharia Reversa em sistemas desse domínio;
  - *Determinação dos Padrões:* para que a determinação dos padrões seja efetivada, atividades tais como a análise de padrões e linguagens de padrões já existentes na literatura e relacionadas com o domínio em questão devem ser realizadas. É importante definir o modelo de classes do domínio que conterá as classes essenciais e complementares ou variantes de cada padrão e, por fim, atribuir uma

denominação a cada um desses padrões criados, devendo seguir o contexto do padrão e relacionar-se à sua aplicabilidade;

- *Criação do Grafo de Fluxo de Aplicação dos Padrões*: o grafo de fluxo de aplicação expressa a navegabilidade e interação entre os padrões, não em questão de funcionamento do sistema, mas da seqüência de aplicação entre eles para favorecer a modelagem do sistema por meio da linguagem de padrões;

- *Escrita dos Padrões*: a escrita dos padrões corresponde a uma descrição dos mesmos, organizada por um grupo predefinido de quesitos ou aspectos, os quais pelas definições encontradas na literatura temática podem ser, basicamente: nome, contexto, problema, forças, solução, entre outros;

- *Validação da Linguagem de Padrões*: a fase de validação consiste, em suma, da aplicabilidade da linguagem de padrões na modelagem de sistemas do domínio a fim de testar, de forma prática, a sua utilização.

• ***Desenvolvimento de um Framework Caixa-Branca***: Assim como o processo geral proposto, a construção do *framework* caixa-branca baseia-se em quatro passos:

- *Identificação dos Pontos Variáveis*: essa etapa realiza a análise nos padrões da linguagem com o objetivo de identificar possíveis variações de aspectos do sistema que seriam incluídas ao *framework* como opções de implementação, não sendo, portanto, necessárias à instanciação da aplicação, mas gerando flexibilidade à sua estrutura;

- *Projeto do Framework*: o projeto do *framework*, dada a definição de seus pontos variáveis, deve contar com os sub-passos de projeto arquitetural (o qual inclui questões referentes á arquitetura dos sistemas a serem instanciados) e projeto de hierarquia de classes (a partir do qual se obtém o projeto completo do *framework*, uma vez já consideradas definidas as classes que compõem cada aspecto do domínio de aplicação);

- *Implementação do Framework*: neste passo os sub-passos existentes correspondem à implementação das classes obtidas na fase anterior em determinada linguagem de programação seguida da documentação do mesmo, desde a geração da linguagem de padrões;

- *Validação do Framework*: consiste simplesmente do mapeamento da aplicação respectiva para o modelo já implementado do *framework* realizando as atividades de implementação e testes específicos.



- **Desenvolvimento do Wizard:** a intenção de desenvolver uma ferramenta de automatização do *framework* se efetiva pelo grande trabalho envolvido em questão da sua implementação. Braga (2002) apresenta duas alternativas no desenvolvimento de uma ferramenta *Wizard*. A primeira consiste no desenvolvimento de uma ferramenta voltada para a implementação específica de um *framework* baseado em linguagem de padrões de determinado domínio, sendo que esta alternativa abrange maior facilidade no desenvolvimento e maior eficiência no uso do *Wizard*, porém, encontra-se desvantagem no fato da aplicação se efetivar somente em *frameworks* específicos. A segunda alternativa consiste no desenvolvimento de um *Wizard* genérico, o qual pode ser adaptado para diversos domínios de *frameworks* baseados em linguagens de padrões, isto fornecendo facilidade quanto à variação dos *frameworks* e linguagens de padrões, porém, conferindo uma menor eficiência na automatização dos mesmos.
- **Definição do Processo de Instanciação do Framework:** o processo de instanciação do *framework* obtido pode ser definido de duas formas: manualmente e por meio da utilização de ferramentas. O processo manual é a implementação e teste na linguagem de programação escolhida resultando no código-fonte das classes de aplicação associadas às classes do *framework*. Já quando da utilização de ferramentas como o *Wizard*, o mesmo processo que é realizado manualmente pode ser nelas realizado de forma automática.

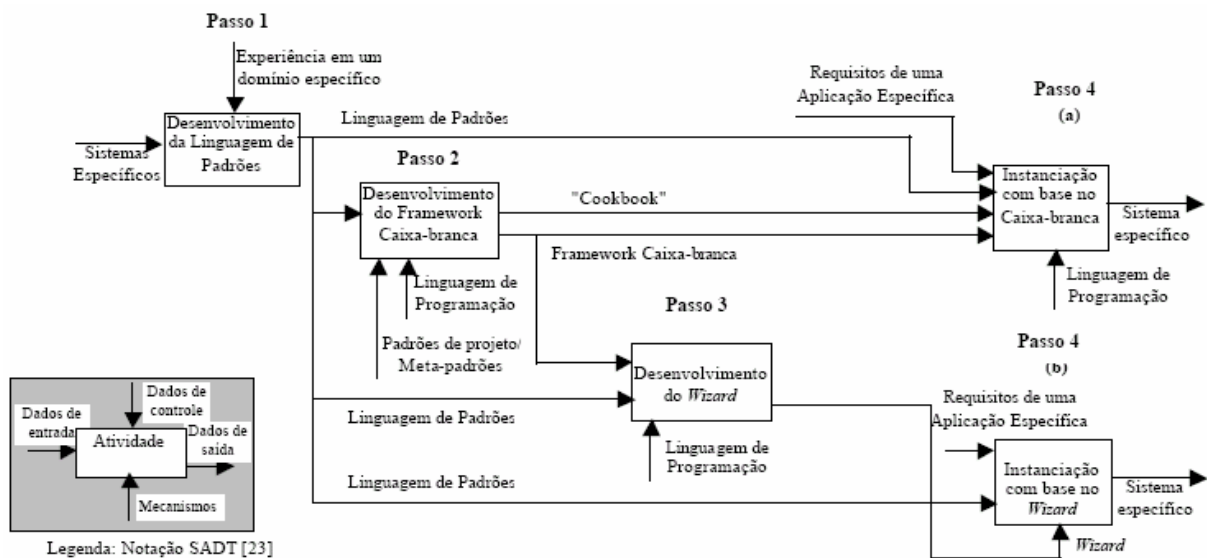


Figura 2-2 Passos do processo proposto por Braga (2002).

Considerando-se as etapas descritas, o processo proposto por Braga (2002) abrange o desenvolvimento de *frameworks* para domínio específico como um todo. Além do que, utilize recursos essenciais da Engenharia de Software para assim fazê-lo. Assim, a aplicação deste processo no desenvolvimento do framework aqui proposto sistematiza a construção do mesmo, fazendo com que o objetivo de utilizar recursos oriundos da Engenharia de Software para o desenvolvimento de módulos educacionais seja efetivamente alcançado.

## 2.7. Considerações Finais

No capítulo em questão foi introduzido o contexto atual do desenvolvimento de software educacional considerando-se os seus elementos principais (conteúdo, apresentação e avaliação) e a sua abrangência em questão da representatividade dos módulos educacionais, os quais constituem o foco desta pesquisa. A seguir houve a explanação acerca de processos de software e modelos de processos de desenvolvimento os quais têm por finalidade a padronização e a qualidade da atividade de desenvolvimento de software. A seção referente aos ambientes virtuais de aprendizagem teve por objetivo a caracterização destes ambientes e a apresentação de alguns deles, com o intuito de conhecê-los de uma forma mais aprofundada, isto favorecendo a sua contextualização com este trabalho – a ser dada pelo fato desses serem o ambiente no qual os módulos educacionais são disponibilizados – e a comparação entre as variedades existentes. Os conceitos sobre padrões de software e linguagens de padrões de software também são importantes neste trabalho principalmente em se tratando de sua utilização para desenvolvimento de módulos educacionais. Por fim, as definições de *frameworks* e suas aplicações, além do seu processo de desenvolvimento, puderam, de certa forma, reforçar o contexto de reuso tratado nesta pesquisa.

Por meio dos aspectos levantados nos tópicos anteriores este trabalho pode ser fundamentado e alicerçado tendo em vista os conceitos do desenvolvimento de softwares educacionais, considerando os módulos educacionais, fazendo uso de formas de reutilização de software oriundas da Engenharia de Software tais como padrões, linguagens de padrões e *frameworks* na construção de módulos educacionais.

## **CAPÍTULO 3. LINGUAGEM DE PADRÕES “LPCAVirtu”**

### **3.1. Considerações Iniciais**

As seções deste capítulo tratam da existência e desenvolvimento de uma linguagem de padrões no contexto do *e-Learning*. É apresentada uma linguagem de padrões já existente na literatura para o contexto em questão, a “*CogLearn*”, da qual constam as características e estrutura. A partir dos aspectos dessa linguagem de padrões, a Seção 3.3 apresenta a linguagem de padrões “*LPCAVirtu*”, a qual foi baseada na linguagem “*CogLearn*”, e que figura resultado desta pesquisa e embasando a análise e projeto de um *framework* no domínio de software educacionais.

### **3.2. Linguagem de Padrões “CogLearn”**

A utilização de padrões e linguagens de padrões no contexto do desenvolvimento de módulos educacionais é um fator que ainda se encontra pouco explorado. As pesquisas que tangem esse assunto são isoladas e, por vezes, incompletas.

Estudo de destaque pode se conferir à pesquisa realizada no Departamento de Computação da UFSCAR - Universidade Federal de São Carlos, referente ao trabalho de Neto (2005), intitulada “Linguagem de Padrões para Apoiar o Projeto de Material Instrucional para EAD”.

Segundo o autor, desenvolver material instrucional para *e-Learning* fazendo uso de ambiente *Web* por parte dos professores, torna-se, na maioria das vezes, um processo enfadonho que gera aulas deficientes em recursos, em geral com excessiva quantidade de textos, que não despertam o interesse e não favorecem o aprendizado do aluno.

Os conceitos fundamentais da pesquisa estão no entendimento e no uso dos padrões como abordagem para captura e apresentação de conhecimento de projeto na solução de problemas e, na importância de uma linguagem de padrões para EAD conter padrões que orientem os professores no planejamento das aulas, considerando aspectos fundamentais como conteúdo, apresentação e navegação.

A estrutura da linguagem proposta por Neto (2005) é mostrada na Figura 3.1. A construção da linguagem de padrões “*CogLearn*”, apresentada no mesmo trabalho, baseou-se

na exploração de padrões previamente definidos provenientes de estudos de caso. Os padrões utilizados se caracterizam como pedagógicos, de IHC (Interface Homem-Computador) e híbridos (pedagógicos e IHC juntamente).

Um ponto interessantes a ser considerado na linguagem em questão seria o fato de poder incluir, alterar e substituir novos padrões à sua estrutura, uma vez que esta é dividida em níveis de abstração, de acordo com as necessidades de cada professor. Outro aspecto é a possibilidade de redução significativa do número de referências entre padrões pela organização em níveis, grupos e sub-grupos, além da representação de especializações.

Aspectos negativos levantados na linguagem de padrões “CogLearn” consistem na ausência de um nível de abstração que compreenda a avaliação do aluno. Afinal, segundo Leiva (2003), esta serve para verificar o ponto de alcance dos vários objetivos educacionais. Além do que, o uso de exemplificação dos padrões, que embora não constitua fator obrigatório na documentação de padrões, como comentado por Maldonado (2004), não é considerado na descrição de todos os padrões que constituem a linguagem. Isso dificulta o entendimento e aplicabilidade de certos padrões. Outro aspecto a ser citado refere-se à sequencialização de padrões que não se torna clara, a primeira vista, na estrutura da linguagem. Ainda, é necessário considerar a inexistência de sub-padrões como forma de refinamento das soluções adotadas em cada padrão.

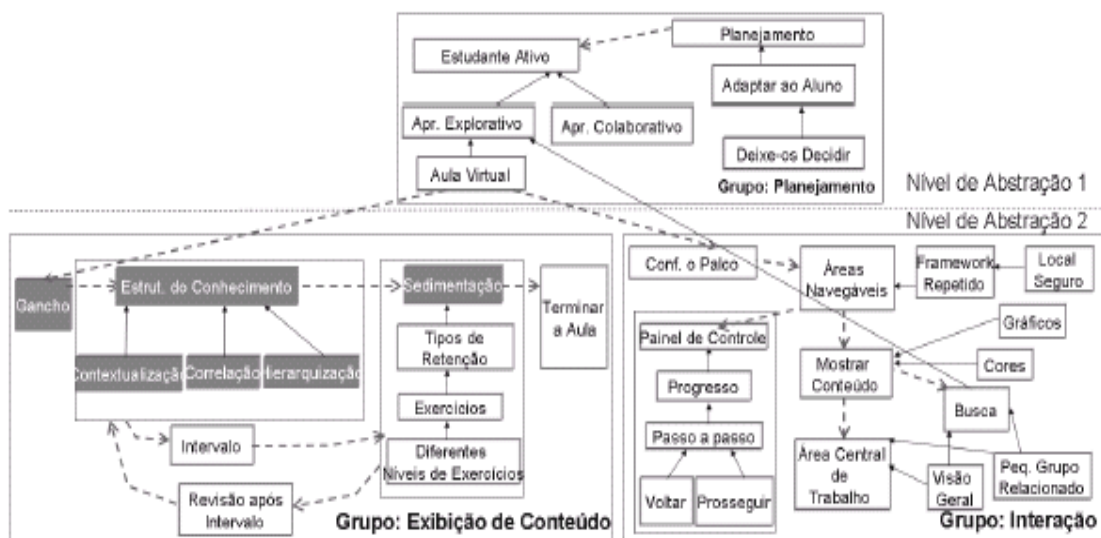


Figura 3-1 – A linguagem de padrões “CogLearn”, seus grupos e níveis de abstração (Neto,2005)

Atualmente, a contar desde a sua apresentação, em 2005, o trabalho de mestrado produzido na UFSCAR tem se aperfeiçoado e conta com a implementação de uma ferramenta, denominada “Cognitor”. Segundo o autor, a ferramenta consiste de um *framework* baseado na

linguagem de padrões definida que busca auxiliar no desenvolvimento de materiais instrucionais para EAD por meio da *Web* (Neto, 2006). Porém, em análise realizada nas respectivas referências, o “Cognitor” não constitui um *framework* e sim um gerador de aulas virtuais uma vez que não abrange os conceitos fundamentais da estrutura de *frameworks*, tais como implementação de variabilidades entre outros. Desta forma, essa ferramenta não será extensamente comentada ou apresentada neste trabalho pelo fato de o *framework* aqui tratado ser uma criação própria que segue processo definido de desenvolvimento e envolver somente as fases de análise e projeto.

### 3.3. Linguagem de Padrões “*LPCAVirtu*”

Por meio de estudos acerca da linguagem de padrões apresentada anteriormente, a “*CogLearn*”, e tomando por base a construção de linguagens de padrões num contexto geral, foi realizado o desenvolvimento da linguagem de padrões proposta nesta pesquisa, a “*LPCAVirtu*” – “Linguagem de Padrões para Construção de Aulas Virtuais”.

O contexto da linguagem *LPCAVirtu*, assim como o da linguagem “*CogLearn*”, envolve os padrões para produção de aulas virtuais com enfoque no ambiente *Web*. Basicamente, a linguagem expõe os padrões essenciais, com suas respectivas variações, para que o professor, mesmo não possuindo conhecimento sobre definição de padrões ou linguagens de padrões, possa produzir uma aula com a melhor qualidade possível alcançando os objetivos pré-estabelecidos no processo de ensino-aprendizagem por parte do aluno.

Considerando a expressividade dos padrões existentes em ambas as linguagens, pode-se evidenciar o fato de sua utilização não somente no desenvolvimento de aulas virtuais, como foram tomados neste trabalho, mas também de uma forma específica, na constituição e orientação de aulas presenciais.

Na Figura 3.2 tem-se a estrutura da linguagem *LPCAVirtu* representada por seus padrões. Estes estão agrupados conforme a sua originalidade: os de cor cinza representam os padrões cujo contexto é advindo da linguagem “*CogLearn*”, mas que foram modificados para mais bem expressar a sua aplicabilidade no contexto de criação de módulos educacionais – essas modificações correspondem às alterações de sua nomeação e de sua aplicabilidade ; os padrões na cor verde são aqueles que apenas foram transportados para a nova linguagem – ou seja, não passaram por alterações ou adaptações; os amarelos são padrões cujo contexto foi adaptado para a visão desta pesquisa – essa adaptação corresponde à alteração somente de sua

nomeação e não de seu contexto – e, por fim, os padrões azuis correspondem aos padrões que foram desenvolvidos e incluídos durante o processo de desenvolvimento da linguagem proposta, a fim de acrescentar e completar aspectos deficientes encontrados na linguagem anterior.

A estruturação da linguagem considerou conceitos advindos das técnicas da UML (*Unified Modeling Language*). Procurou-se demonstrar a forma de navegação entre os padrões na possível seqüência na qual estes são aplicados para desenvolver uma aula virtual, objetivo da linguagem neste trabalho. As bordas espessas existentes na Figura 3.2 correspondem à delimitação da linguagem em três grupos: Conteúdo, Navegação e Avaliação. Estes grupos são os itens básicos da composição de softwares educacionais, conforme já visto na Seção 2.2.

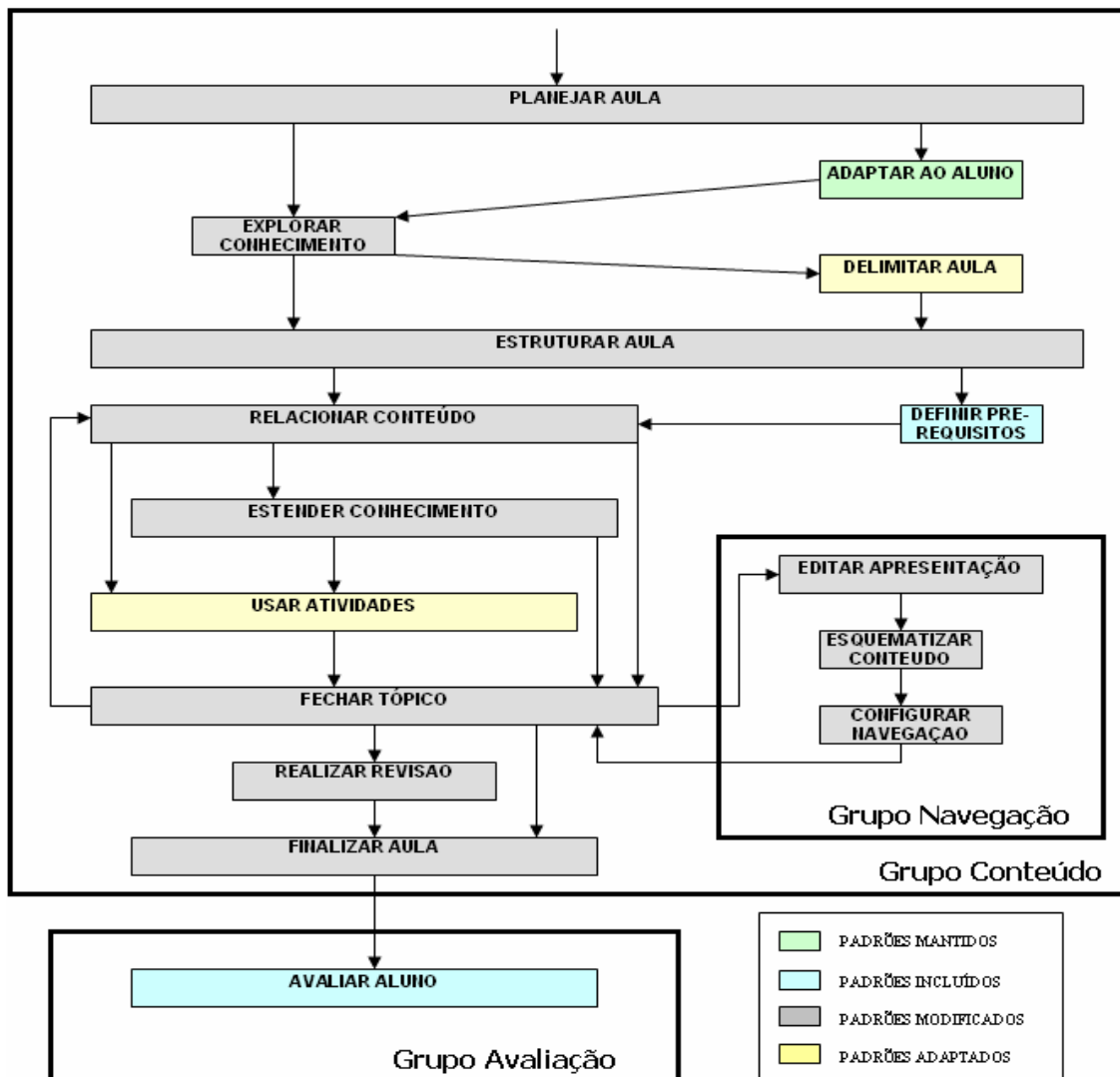


Figura 3-2 Estrutura da linguagem de padrões “LPCAVirtu” considerando contexto dos padrões que a compõe.

O Grupo Conteúdo contém padrões que orientam o professor em como escolher determinado conteúdo a ser disponibilizado na aula, como estruturá-lo de forma a incluir recursos que auxiliem no processo de ensino-aprendizagem e em como aperfeiçoar de forma efetiva este conteúdo para que o mesmo seja aprendido pelo aluno de forma eficiente. Neste grupo, padrões como DELIMITAR AULA, USAR ATIVIDADES e DEFINIR PRÉ-REQUISITOS são padrões totalmente voltados para a qualidade da aula. Padrões como ESTENDER CONHECIMENTO e REALIZAR REVISÃO foram reestruturados considerando a linguagem anterior para representarem além de suas configurações originais, variações de solução abordadas respectivas à identificação de sub-padrões.

Em relação ao Grupo Navegação, as questões de edição e formatação da aula estruturada no Grupo Conteúdo conferem a importância deste agrupamento de padrões. Existem, por conseguinte, variações dos padrões deste grupo, as quais flexibilizam a forma de uso da linguagem neste aspecto. O uso de padrões como ESQUEMATIZAR CONTEÚDO e CONFIGURAR NAVEGAÇÃO pode auxiliar de forma muito significativa no aprendizado por parte do aluno devido a se tratarem, conforme Neto (2005), de padrões de IHC (Interface Humano-Computador) que solucionam os problemas de interfaceamento da aula com o aluno.

O Grupo Avaliação, por sua vez, é identificado como o grupo de relevância notória na linguagem proposta, tendo em vista ser constituído por padrões que foram desenvolvidos no próprio processo de construção da linguagem considerando aspectos que não se apresentavam na linguagem “*CogLearn*”. O padrão AVALIAR ALUNO contido neste grupo é necessário para que o professor possa identificar os níveis de conhecimento e desempenho do aluno em relação aos conteúdos apresentados. Por meio da utilização desse padrão, em especial de suas variações, o professor conta com recursos amplos para avaliar os conhecimentos do aluno da forma como lhe convier.

### **3.3.1. Exemplificação de Dois Padrões da Linguagem “*LPCAVirtu*”**

Devido à quantidade de padrões identificados para a linguagem “*LPCAVirtu*” e a efetiva extensão desta pesquisa, esta seção apresenta a exemplificação de alguns padrões da linguagem em detalhes. Todos os demais padrões constituintes da linguagem possuem seu problema, contexto, solução e contexto resultante especificados no Apêndice C.

A título de exemplificação e para abranger o contexto de desenvolvimento dos padrões, dois destes foram tomados como referência. O primeiro, ESTENDER

CONHECIMENTO, foi escolhido pelo fato de ser um padrão adaptado e modificado em relação à linguagem utilizada como base para o desenvolvimento da linguagem proposta. E, o segundo, o padrão AVALIAR ALUNO, foi utilizado nesta exemplificação pelo fato de ser um padrão novo, criado neste trabalho, e incluído à linguagem proposta em seu processo de construção.

### **PADRÃO: ESTENDER CONHECIMENTO**

**PROBLEMA:** Como o aluno pode ter noção acerca dos conhecimentos que deve adquirir durante uma aula e como o professor pode dispor maiores informações sobre o conhecimento a ser aprendido?

**CONTEXTO:** O aluno, mesmo quando em uma aula presencial, tem a necessidade de absorver o conteúdo por meio dos mais diversos recursos. Uma vez estando definido o assunto da aula, o professor deve pensar em formas de estender este conteúdo para que ele seja apresentado de forma compreensível e completa.

**SOLUÇÃO:** Como estratégias para poder apresentar o conteúdo, lançando mão de recursos com os quais o aluno possa compreender a matéria explorada, o professor pode basicamente aplicar o padrão ESTENDER CONHECIMENTO, tomando por base três sub-padrões: EXEMPLIFICAR NO DOMÍNIO, METAFORIZAR e SUB-APRESENTAR O CONTEÚDO. A solução EXEMPLIFICAR NO DOMÍNIO consiste na alternativa do professor utilizar exemplos de aplicação do tema tratado dentro do próprio contexto do mesmo. Quanto ao sub-padrão METAFORIZAR, o mesmo consiste na exemplificação do tema por meio da utilização de recursos tais como figuras ou esquemas, os quais têm a finalidade de relacionar o conteúdo apresentado à situações cotidianas já conhecidas pelo aluno. Por fim, o sub-padrão SUB-APRESENTAR O CONTEÚDO trata da questão de organizar um conteúdo extenso de forma a exibi-lo em apresentações menores para o seu melhor entendimento por parte do aluno. Todas as variações do padrão em questão estão apresentadas na Figura 3.3.

**CONTEXTO RESULTANTE:** A utilização do padrão Estender Conhecimento faz com que o aprendizado dos alunos possa, por meio dos recursos apresentados nas diferentes instâncias da solução, ser intensificado, uma vez que contará com a possibilidade deste conteúdo ser exemplificado e desta forma tornar-se mais compreensível.

**PRÓXIMOS PADRÕES:** Ao passar pelo padrão ESTENDER CONHECIMENTO o professor tem a opção de realizar a aplicação do padrão USAR ATIVIDADES, quando



preferir que o conteúdo do tópico seja ainda trabalhado por mais algum tempo pelo aluno, ou avançar para a utilização do padrão FECHAR TÓPICO, o qual deve ser aplicado quando a quantidade de informação e exploração sobre o assunto de determinado tópico já foi totalmente explorada.

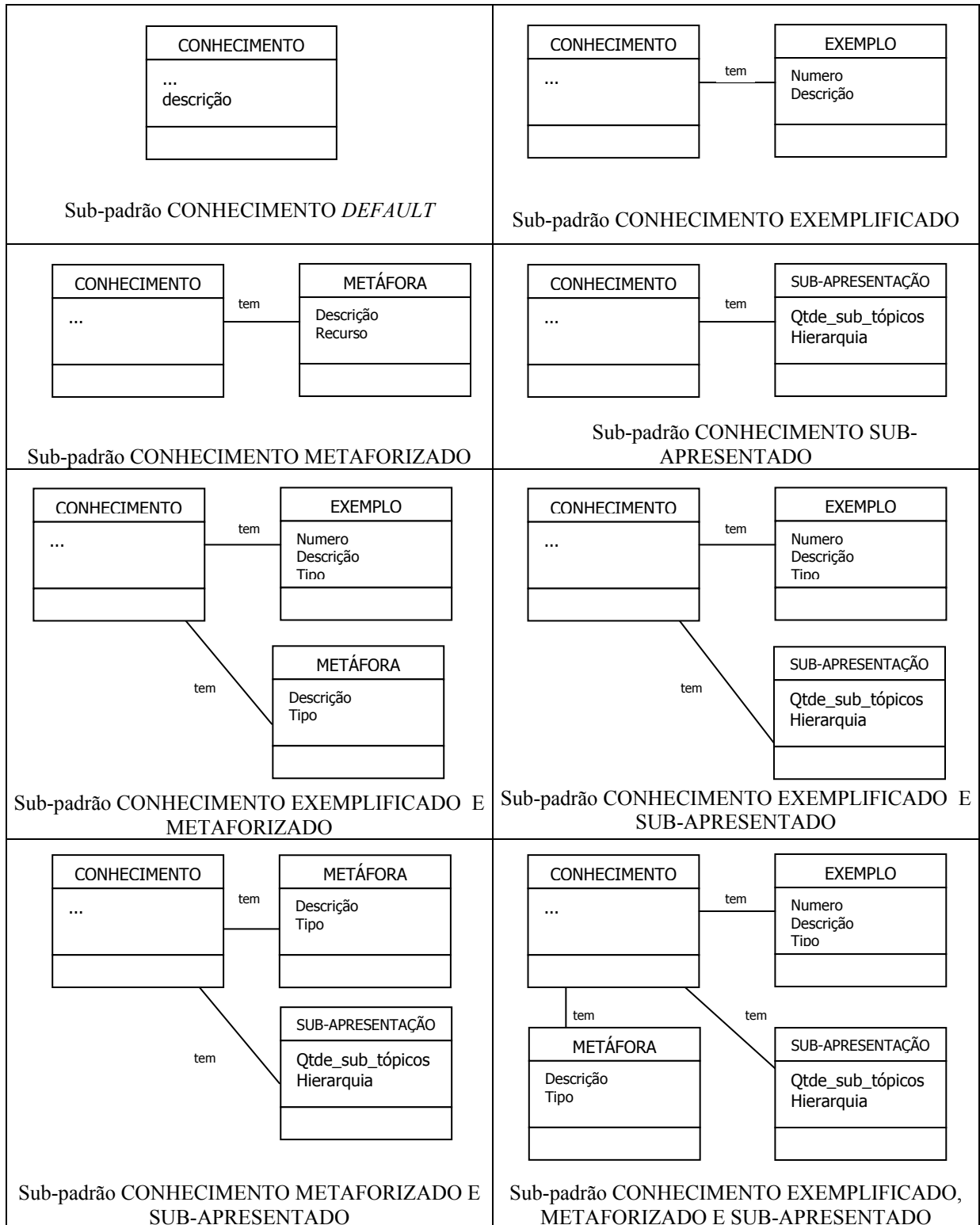


Figura 3-3 Padrão ESTENDER CONHECIMENTO e suas variações

## **PADRÃO: AVALIAR ALUNO**

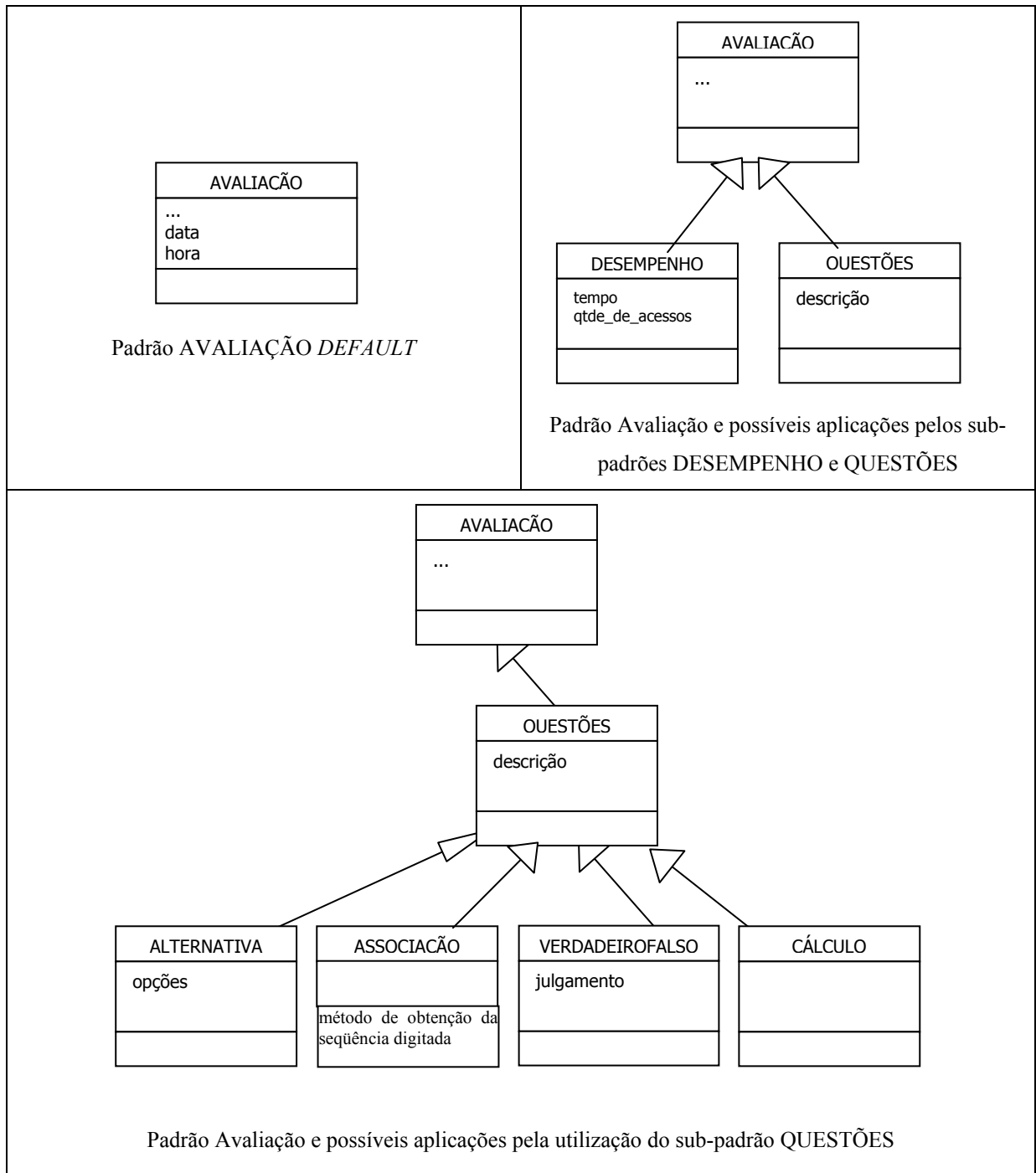
**PROBLEMA:** Como aplicar o processo de avaliação do aluno?

**CONTEXTO:** Em qualquer aula, mesmo presencial, é importante tanto para o aluno quanto para o professor que um processo de avaliação seja realizado. Esse processo de avaliação pode revelar o desempenho do aluno quanto à aula realizada, o nível de dificuldade do material disponibilizado e o nível de conhecimento dos alunos de determinado perfil.

**SOLUÇÃO:** O processo de avaliação em uma aula virtual pode contar com o mais diversos recursos: questões (alternativas, dissertativas, lacunas, associação, jogos, entre outros), controle de acesso, tempo de execução da aula, desempenho nas atividades. Considerando essas possíveis soluções como variações da aplicação do padrão AVALIAR ALUNO, pode-se identificar dois sub-padrões desse padrão: AVALIAR\_POR\_DESEMPENHO – no qual o professor pode envolver aspectos tais como o perfil do aluno durante a realização da aula - e AVALIAR\_POR\_QUESTÕES – no qual o professor pode utilizar os mais diversos tipos de questões para avaliar o conteúdo apresentado e aprendido pelo aluno na aula virtual. O sub-padrão AVALIAR\_POR\_QUESTÕES, nesta exemplificação passa a ser identificado por possíveis variações de implementação em relação ao sub-padrão QUESTÕES, essas variações, juntamente com a representação do padrão AVALIAR ALUNO, encontram-se nos esquemas apresentados pela Figura 3.4.

**CONTEXTO RESULTANTE:** O padrão AVALIAR ALUNO é um padrão extenso que requer envolvimento de aspectos pedagógicos. Sua utilização inclui mais uma alternativa de motivação ao aluno quanto à exploração da aula e ao professor uma ferramenta para diagnosticar o efeito de suas aulas. Pela extensa quantidade de soluções que este padrão pode envolver, de acordo com a literatura referente, este trabalho restringe a utilização destas variações ao contexto já apresentado na Figura 3.4.

**PRÓXIMOS PADRÕES:** Considerando a aplicação do padrão AVALIAR ALUNO, o professor pode contar com a possibilidade de aplicar o padrão AVALIAR\_POR\_DESEMPENHO ou AVALIAR\_POR\_QUESTÕES. No caso da utilização da segunda opção de padrão, a mesma apresenta a possibilidade de migração para diversos outros sub-padrões que podem constituir-la. Estes padrões referem-se a alguns tipos de questões que podem ser implementadas no processo de avaliação da aula virtual. Portanto, como apresentado na Figura 3.4, a aplicação do sub-padrão AVALIAR\_POR\_QUESTÕES leva à aplicação de outros sub-padrões: ALTERNATIVA, ASSOCIAÇÃO, VERDADEIRO/FALSO e CÁLCULO.



**Figura 3-4 Padrão AVALIAR ALUNO e suas variações**

### 3.4. Uso da linguagem “LPCAVirtu” para planejar e estruturar uma aula virtual de Engenharia de Software

Para ilustrar o processo de aplicação da linguagem de padrões proposta, esta seção sugere o uso da mesma na atividade de planejar e estruturar uma aula virtual da disciplina de Engenharia de Software. Os passos e a utilização de cada padrão serão apresentados

juntamente com os detalhes da aplicação de variabilidades dos padrões quando da sua necessidade. Recorreu-se, quanto ao conteúdo, à utilização de uma aula apresentada ao curso de Ciência da Computação, na disciplina de Engenharia de Software, no formato de aula presencial para servir como base para a aplicação da linguagem de padrões na construção da aula virtual. Os slides da mesma correspondem à produções do Grupo de Engenharia de Software do ICMC (Instituto de Ciências Matemáticas e Computação) da USP de São Carlos.

O primeiro passo a ser realizado pelo professor consiste em planejar a aula determinando o tema a ser tratado e delimitando a abrangência do mesmo. Além disso, ainda faz parte desta fase a definição dos objetivos e resultados a serem alcançados pelo aluno a partir da realização da aula. Todas estas atividades são orientadas pelo padrão PLANEJAR AULA.

Ex: PLANEJAR AULA

DISCIPLINA: *“Engenharia de Software”*

ASSUNTO: *“Definição de Conceitos Introdutórios sobre Engenharia de Software”*

Para que o aluno possa familiarizar-se completamente com o material apresentado é necessário que o professor utilize o padrão ADAPTAR AO ALUNO.

Ex: ADAPTAR AO ALUNO

*Utilização de linguagem simples e adaptada. Termos técnicos, quando possível, podem ser substituídos ou exemplificados para melhor entendimento.*

O próximo passo a ser executado trata do planejamento dos recursos que o professor utilizará para que o aluno entre em contato com o conhecimento apresentado. A própria aula virtual constitui uma forma de aplicação do padrão EXPLORAR CONHECIMENTO.

Ex: EXPLORAR CONHECIMENTO

*A aula virtual introdutória sobre Engenharia de Software conta com recursos multimídia como som e imagem.*

Pode haver ainda a possibilidade do professor deixar a delimitação do tema da aula a cargo do aluno por meio do padrão DELIMITAR AULA, onde conjuntos de tópicos podem ser planejados de acordo com o conhecimento prévio do aluno que pode ser detectado pelo seu perfil.

Ex: DELIMITAR AULA

*O professor delimita dois conjuntos de aula, de acordo com dois perfis do aluno. Um conjunto relaciona-se aos alunos cujo conhecimento em Engenharia de Software é limitado e outro é voltado para os alunos que já possuem conhecimento na disciplina e apenas querem rever conceitos.*

## Estruturação

Uma vez planejado, o conteúdo passa a ser efetivamente estruturado. Antes, porém, de iniciar a apresentação da aula o professor pode determinar os pré-requisitos para que o aluno possa realizar a aula com completude por meio do padrão DEFINIR PRÉ-REQUISITOS.

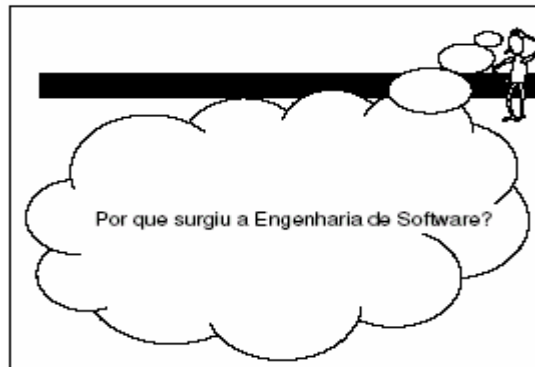
Ex: DEFINIR PRÉ-REQUISITOS

*Como requisitos para realizar esta aula o aluno deve possuir conhecimentos básicos em informática e programação.*

Para estruturar o início da apresentação de determinada aula ou tópico se faz necessária a aplicação do padrão RELACIONAR CONTEÚDO.

Ex: RELACIONAR CONTEÚDO

*Para a aula aqui estruturada pode-se utilizar uma tela de abertura que por meio de uma questão ou definição do tema central possa dar a idéia do assunto tratado.*



Fonte: ICMC - USP

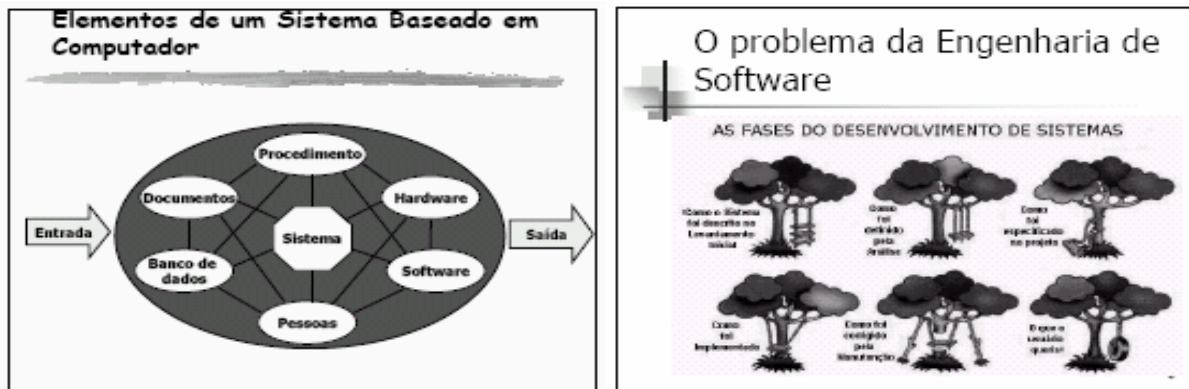
**Figura 3-5 Exemplo da aplicação do padrão RELACIONAR CONTEÚDO por meio de questão introdutória do histórico da Engenharia de Software**

A aplicação do padrão ESTENDER CONHECIMENTO é apresentada como alternativa sendo que da apresentação do conteúdo o professor pode migrar para o USAR ATIVIDADES ou FECHAR TÓPICO. Para estender certo conteúdo podem ser utilizadas as variabilidades do padrão constituídas por EXEMPLOS, METÁFORAS e SUB-APRESENTAÇÕES.

Ex: ESTENDER CONHECIMENTO

*Para melhor apresentar os conceitos de Engenharia de Software para os alunos, ainda mais considerando aqueles que nunca tiveram envolvimento com a disciplina,*

recursos de exemplificação e metáfora das definições serão utilizadas, sempre que possível.



Fonte: ICMC - USP

**Figura 3-6** Aplicação do padrão **ESTENDER CONHECIMENTO** com as variantes de **EXEMPLIFICAÇÃO** e **METAFORIZAÇÃO**, respectivamente.

Com a aplicação do padrão apresentado anteriormente o professor alcança condição de poder dispor aos alunos meios de fixar o conteúdo aprendido. Ele tem para isso o padrão **USAR ATIVIDADES**.

Ex: **USAR ATIVIDADES**

*Nesta aula de Engenharia de Software, depois da exploração de cada conteúdo serão utilizadas atividades como questões interativas e links para acesso a sites pedagógicos relacionados ao assunto estudado.*

Uma vez chegado ao fim de determinado tópico referente à aula em questão, o professor pode fazer uso do padrão **FECHAR TÓPICO** o qual dará suporte e recursos adequados para expressar o encerramento do módulo.

### *Edição*

Definida a estrutura da aula o professor tem a disponibilidade de editar o conteúdo estruturado. Por meio do padrão **EDITAR APRESENTAÇÃO** a formatação de textos e adição de recursos multimídia ao mesmo podem ser aplicadas fazendo com que as telas da aula sejam mais bem apresentadas.

Ex: **EDITAR APRESENTAÇÃO**

*Na aula sobre Engenharia de Software a fonte padrão para os textos será a Tahoma, no tamanho 14. Os títulos terão cor de fonte azul, tamanho 16, serão sublinhados e em estilo negrito. Serão incluídos sons a cada mudança de página e no acesso aos tópicos mais importantes para chamar a atenção dos alunos.*

O conteúdo já estruturado e editado passará, por sua vez, pela aplicação do padrão ESQUEMATIZAR CONTEÚDO, no qual será definida a melhor forma de organização do mesmo.

Ex: ESQUEMATIZAR CONTEÚDO

*O acesso aos tópicos da aula será disponibilizado por meio de um índice que considerará os tópicos pela sua ordem de aprendizado.*

Com o conteúdo já esquematizado em tópicos pode-se configurar formas de navegação pela aula parcialmente construída. O padrão CONFIGURAR NAVEGAÇÃO dispõe ao professor recursos editar a navegação pelos tópicos e páginas da aula de acordo com a preferência do mesmo.

Ex: CONFIGURAR NAVEGAÇÃO

*A navegação pelas páginas da aula será estruturada por meio de botões existentes na parte inferior das mesmas. O aluno poderá avançar ou retroceder de página em página, mas, no caso dos alunos iniciantes, estes nunca poderão acessar páginas de tópicos posteriores sem ter visitado todas as páginas de tópicos anteriores. A visita completa do tópico implicará na alteração da cor de sua fonte no índice.*

Como um recurso adicional, o professor pode utilizar o padrão REALIZAR REVISÃO, o qual quando aplicado favorece o aprendizado por parte do aluno.

Ex: REALIZAR REVISÃO

*Ao fim da aula introdutória sobre Engenharia de Software uma breve revisão dos conceitos mais importantes de cada tópico será disponibilizada ao aluno como reforço adicional de aprendizado.*

Para encerrar a aula o professor pode aplicar o padrão FINALIZAR AULA, no qual, por base, uma página de finalização é construída.

Como uma última e fundamental aplicação, o padrão AVALIAR ALUNO constitui recursos e alternativas para que o professor possa, por meio da construção de um conjunto de avaliação, medir os resultados alcançados por parte do aluno em relação à aula realizada.

Ex: AVALIAR ALUNO

*O conjunto de avaliação desta aula será composto por questões alternativas, de ordenação e associação relacionadas ao conteúdo apresentado nos tópicos. Ao final da realização das questões o aluno receberá uma pontuação sobre a qual será indicada a sua aprovação ou reprovação na realização da aula.*

### 3.5. Considerações Finais

Neste capítulo foi apresentada a linguagem de padrões “*CogLearn*” e as suas características, as quais serviram como base na elaboração de uma nova linguagem de padrões para o contexto do *e-Learning*, a linguagem de padrões “*LPCAVirtu*”, apresentada na Seção 3.3. Constitui foco na mesma seção a apresentação da forma de desenvolvimento e dos padrões que constituem a linguagem, esta com diferenciais em relação à primeira, principalmente considerando a introdução de novos padrões e adaptação dos já existentes, uma vez que esses foram modificados com o intuito de aplicar de melhor forma soluções que evidenciam os três grupos de requisitos básicos dos sistemas educacionais citadas na Seção 2.2. Como complemento, os padrões ESTENDER CONHECIMENTO e AVALIAR ALUNO, constituintes da linguagem, juntamente com suas variabilidades de aplicação, foram detalhados na Seção 3.3.1. Na Seção 3.4, uma forma de aplicação da linguagem proposta foi apresentada, de forma a esboçar a real utilização da “*LPCAVirtu*” na construção de aulas virtuais, embora em outros contextos a linguagem também possa ser aplicada à construção de aulas presenciais. Por fim, vale ressaltar que a linguagem de padrões considerada como base na construção da linguagem proposta neste trabalho possui limitações, as quais a linguagem “*LPCAVirtu*” procura suprimir. Dentre as principais limitações foram apresentadas: a não utilização de sub-padrões que contribuem com o refinamento das possíveis soluções implementadas por cada padrão, a ausência de um módulo para avaliação do aluno e a dificuldade na sequencialização e / ou navegação entre os padrões.



## **CAPÍTULO 4. ANÁLISE E PROJETO DE UM *FRAMEWORK* NO DOMÍNIO DE SOFTWARES EDUCACIONAIS**

### **4.1. Considerações Iniciais**

A partir dos resultados obtidos neste trabalho e, com base em um processo de construção de *frameworks* existente na literatura pesquisada, este capítulo trata da análise e projeto de um *framework* caixa-branca baseado na linguagem de padrões “*LPCAVirtu*”, apresentada no capítulo anterior.

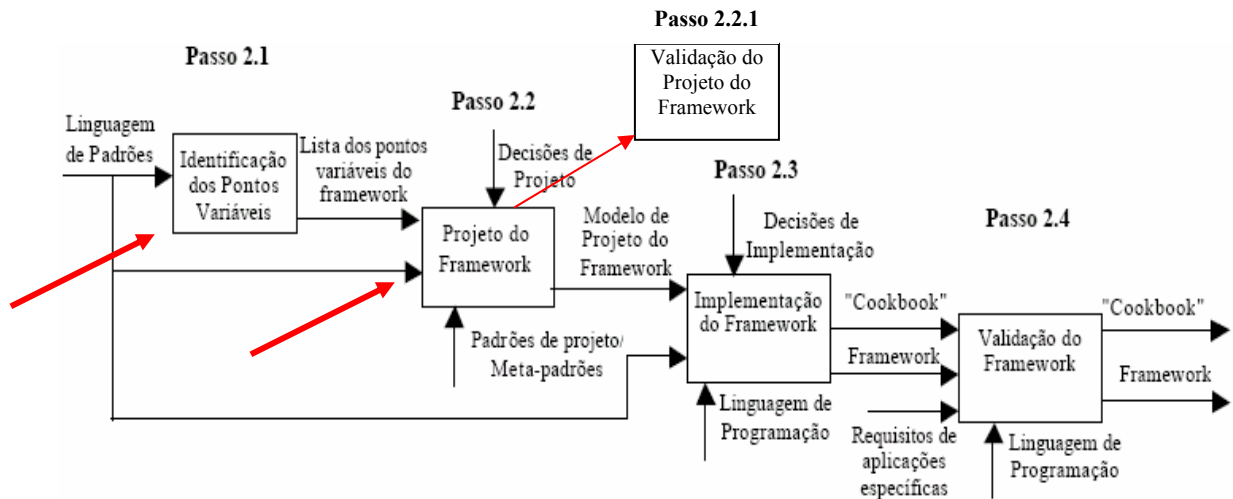
Na Seção 4.2 é comentada a estrutura do *framework* a ser desenvolvido juntamente com o processo utilizado para assim fazê-lo. Na Seção 4.3 apresenta-se o processo de análise do *framework* por meio do levantamento dos seus pontos variáveis. Já quanto a Seção 4.4 esta trata especificamente da atividade de projeto do *framework* proposto neste trabalho.

### **4.2. Processo de Construção do *Framework* Caixa-Branca “*FCAVirtu*”**

O processo de construção de *frameworks* utilizado neste trabalho é apresentado na Seção 2.6.1. O mesmo constitui-se de quatro fases, conforme já explicitado anteriormente, as quais são apresentadas na Figura 4.1.

Para a construção do *framework* caixa-branca proposto neste trabalho, o “*FCAVirtu*” – “*Framework para Construção de Aulas Virtuais*”, o qual será baseado na linguagem de padrões para o domínio educacional, a “*LPCAVirtu*”, a utilização do processo proposto por Braga (2002) será primordial uma vez que este processo tem sua fundamentação para *frameworks* de domínio específico.

A utilização do processo proposto, no entanto, sugeriu a adaptação do mesmo para o contexto deste trabalho. Essa adaptação foi realizada por meio da inclusão do Passo 2.2.1, identificado como “Validação do Projeto do Framework”, conforme mostrado na Figura 4.1. A necessidade de incluir este sub-passo na definição do processo surge da limitação do escopo da pesquisa aqui realizada, a qual, por ser restrita, não poderá contar com o processo de validação do *framework* após a sua implementação, uma vez que o *framework* não será implementado, mas apenas projetado.



**Figura 4-1** Processo para a construção de um framework baseado em linguagem de padrões (Adaptado de Braga (2002)).

Assim, a aplicação do processo definido se dará apenas pela realização das atividades indicadas pelas setas nas cores vermelhas na Figura 4.1, considerando o passo 2.2.1, o qual foi incluído como complementação para atender ao contexto desta pesquisa, sendo que nas próximas seções a aplicação destas atividades na construção do *framework* proposto será detalhada conforme a sua utilização.

### 4.3. Identificação dos Pontos Variáveis

Segundo Braga (2004), a identificação dos pontos variáveis do *framework* no processo proposto constitui atividade fundamental, pois levanta os aspectos de flexibilidade do mesmo na instanciação dos softwares por ele gerados. Cada software requer particularidades ou generalidades que devem ser previstas e atendidas pelo *framework*.

Esta atividade do processo de construção é sub-dividida em cinco passos. A seguir, cada um desses passos será apresentado:

- *1ª passo*: Análise do grafo da linguagem de padrões – Esta análise considera a estrutura da linguagem de padrões, em especial, tomando por base o item PRÓXIMOS PADRÕES. Por meio deste item variabilidades podem ser verificadas em relação à opção de utilizar ou não determinado padrão no caminho de aplicação, de acordo com o contexto requerido pelo software.
- *2ª passo*: Análise de cada padrão – Na análise individual dos padrões pode-se verificar flexibilidades relacionadas à forma de aplicação do padrão em cada

situação. O padrão pode vir a assumir várias soluções para o problema exposto e essas várias soluções também constituem pontos variáveis para o *framework*.

- 3ª *passo*: Refinamento da lista de pontos variáveis – Com as variações já obtidas pode-se refiná-las, a fim de que a implementação posterior do *framework* seja dada de forma mais completa. Esse passo também é importante pelo fato da possibilidade de inclusão de novas variações não definidas na linguagem de padrões, estas, portanto, sendo oriundas de conhecimento de implementação do *framework*.
- 4ª *passo*: Referência cruzada da lista de pontos variáveis – Incluído como mais uma possibilidade de aperfeiçoar a linguagem de padrões, este passo envolve conhecimento direto do domínio abordado, a fim de que possa ser realizada uma referência cruzada entre os pontos já identificados e, esta ser capaz de identificar inconsistências e, talvez, novos pontos variáveis.
- 5ª *passo*: Análise de requisitos não funcionais – Com base na análise de requisitos a consideração dos requisitos não funcionais também pode constituir atividade para identificação dos pontos variáveis, uma vez que esses requisitos, mesmo não pertencendo diretamente à linguagem de padrões, possam gerar variações na instanciação de cada software por meio do *framework* criado.

A identificação dos pontos variáveis para o *framework* “FCAVirtu”, utilizou os aspectos da linguagem de padrões como base na aplicação do segundo, terceiro e quarto passos do processo de identificação de pontos variáveis proposta por Braga (2002).

Para apoiar o refinamento das variabilidades e, posteriormente, o passo de projeto do *framework*, um documento de requisitos com base no domínio de softwares educacionais de apoio ao processo de ensino-aprendizagem, já construído, foi utilizado. O mesmo se encontra apresentado no Apêndice A, enquanto que sua representação em diagrama de caso de uso está expressa no Apêndice B.

As variabilidades encontradas por meio da execução dos cinco passos propostos no processo de Braga (2002) estão dispostas no Quadro 4.1. Neste quadro cada variabilidade está representada por meio do nome do padrão de origem, possibilidades de variação, descrição e passo pertencente ao processo utilizado na qual a variabilidade foi detectada.

<b>Padrão / Sub-Padrão</b>	<b>Variações</b>	<b>Descrição</b>	<b>Passo Utilizado</b>
PLANEJAR AULA	- ADAPTAR AO ALUNO	O(s) padrão(ões) identificado(s) como variante(s) pode(m) constituir opção(ões) para o professor quando da transição do padrão origem.	1º passo
EXPLORAR CONHECIMENTO	- DELIMITAR AULA		
ESTRUTURAR AULA	- DEFINIR PRÉ-REQUISITOS		
RELACIONAR CONTEÚDO	- USAR ATIVIDADES - FECHAR TÓPICO		
FECHAR TÓPICO	- RELACIONAR CONTEÚDO - EDITAR APRESENTAÇÃO - FINALIZAR AULA		
ESTENDER CONHECIMENTO	- EXEMPLIFICAR NO DOMÍNIO - METAFORIZAR - SUB-APRESENTAR CONTEÚDO	As variantes identificadas para o padrão em questão constituem sub-padrões extraídos por meio das várias soluções existente na aplicação do padrão em questão a um problema específico do domínio tratado.	2º passo
REALIZAR REVISÃO	- EXERCITAR CONTEÚDO		
EDITAR APRESENTAÇÃO	- MANTER LAYOUT - MODIFICAR LAYOUT		
CONFIGURAR NAVEGAÇÃO	- PADONIZAR ACESSO		
AVALIAR ALUNO	- AVALIAR_POR_QUESTÕES -AVALIAR_POR_DESEMPENHO		
QUESTÃO	- ALTERNATIVA - ASSOCIAÇÃO - VERDADEIRO/FALSO - CÁLCULO	A identificação das variações corresponde ao refinamento de sub-padrões de forma a envolver o conhecimento de implementação do software do domínio tratado.	3º passo
METAFORIZAR	- EXEMPLIFICAR NO DOMÍNIO - SUB-APRESENTAR	Por meio de análise e referência cruzada entre os padrões e sub-padrões podem existir inconsistências em questão de determinados sub-padrões poderem coexistir em comum com outros sub-padrões	4º passo

**Quadro 4-1 Identificação dos pontos variáveis do framework “FCAVirtu” considerando as fases do processo proposto por Braga(2002).**

#### 4.4. Projeto do Framework

Seguindo a seqüência do processo proposto por Braga (2002), o passo de projeto do *framework*, uma vez tendo sido identificados os pontos variáveis para o mesmo, refere-se à construção da estrutura para o *framework* na qual possa existir a máxima flexibilidade na abrangência das variabilidades definidas na seção anterior.

Assim como na etapa anterior, a atividade de projeto do *framework* também pode ser sub-dividida em sub-passos, a citar: a elaboração do projeto arquitetural e do projeto das

classes. Ambos os sub-passos serão tratados a seguir considerando a sua utilização no contexto do projeto do *framework* proposto nesta pesquisa.

- *1ª sub-passo*: Projeto Arquitetural – Neste sub-passo constam as especificações referentes à definição estrutural do *framework*, a sua arquitetura. São considerados os aspectos de persistência dos objetos, interface gráfica com o usuário (GUI), questões de segurança, acesso e distribuição.
- *2ª sub-passo*: Projeto das Classes – Neste momento é definida a hierarquia de classes do *framework* considerando a identificação dos *hot spots* e os padrões de projeto. Constituem atividades desse sub-passo: projeto geral do *framework*, generalização / especialização, uso de padrões de projeto, projeto de outras camadas. A aplicação das atividades desse sub-passo leva a um diagrama que representa a cobertura da linguagem de padrões em seu todo.

A partir da identificação dos pontos variáveis existentes para o *framework* “*FCAVirtu*”, mostrados no Quadro 4.1, foi possível, considerando o passo inicial de análise e o passo de projeto, projetar a arquitetura do *framework*, a qual é mostrada e comentada na Seção 4.5, e construir um diagrama das classes que compõem o *framework*, este apresentado e também comentado na Seção 4.6.

#### **4.5. Arquitetura do *framework* “*FCAVirtu*”**

A atividade de projeto da arquitetura do *framework* “*FCAVirtu*” seguiu as definições apresentadas por Braga (2002) no 1º sub-passo do passo de projeto do *framework*, constante na Seção 4.4. Na Figura 4.2 a estrutura originária da realização dessa atividade é apresentada.

A arquitetura do *framework* aqui proposto considera as camadas de persistência – responsável pelo armazenamento e recuperação de informações, estabelecimento de conexões com bases de dados e persistência dos objetos – e interface gráfica com o usuário (GUI) – referente à implementação de classes para implantação, gerenciamento e controle das interações do usuário com o sistema. Porém, pela abrangência desta pesquisa, estas camadas não serão desenvolvidas, seguindo esta atividade como sugestão de trabalho futuro.

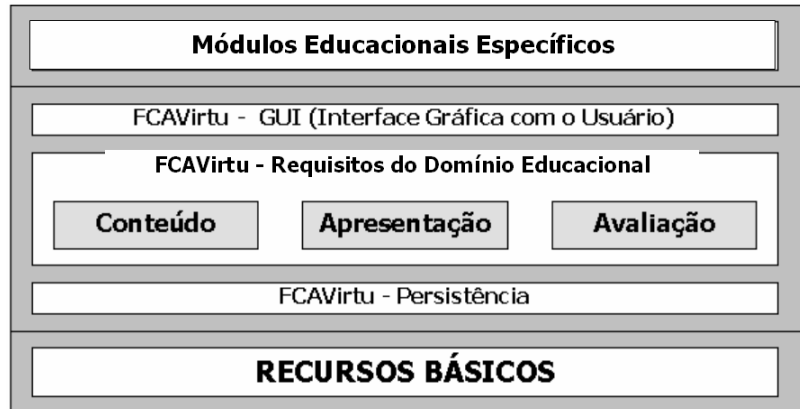


Figura 4-2 Arquitetura do framework proposto neste trabalho

A camada trabalhada nesta pesquisa, a de Requisitos do Domínio Educacional, está representada por delimitações mais específicas existentes na camada dos Módulos Educacionais Específicos. Essas delimitações correspondem aos grupos de requisitos identificados na Seção 2.2.

Também está representada nesta estrutura a possibilidade da instanciação considerando apenas os Recursos Básicos, Persistência e Requisitos do Domínio Educacional, excluindo-se a obrigatoriedade da utilização da camada da GUI para efetuar a instanciação do *framework* na aplicação específica. Isso confere maior flexibilidade na utilização do *framework* proposto, uma vez que se possa utilizar a estrutura do mesmo juntamente com uma interface gráfica disponibilizada por outros meios.

#### 4.6. Hierarquia de Classes do “FCAVirtu”

A construção da hierarquia das classes do *framework* proposto neste trabalho baseou-se no segundo sub-passo do passo de projeto do *framework*, definido por Braga (2002) e apresentado na Seção 4.4.

De acordo com as definições de variabilidade e arquitetura do *framework*, a elaboração das classes de projeto foi realizada a partir da consideração do mapeamento dos padrões em classes do *framework*, e por meio da utilização do documento de requisitos e diagrama de caso de uso do domínio, tratados respectivamente, nos Apêndices A e B.

Para que o desenvolvimento da hierarquia das classes do *framework* pudesse ser realizado, foi utilizado em determinadas situações o padrão de projeto STRATEGY (Gamma *et al.*, 1994), por meio do qual o *framework* pode implementar diversas soluções a um mesmo

problema. A aplicação deste padrão, conforme a Figura 4.4 que representa a hierarquia de classes do *framework*, se efetiva na implementação das classes *Conhecimento* e *Layout*.

A classe *Conhecimento* foi mapeada a partir dos padrões RELACIONAR CONTEÚDO e ESTENDER CONHECIMENTO. Para aplicar a solução ou estratégia ao problema identificado por meio do padrão ESTENDER CONHECIMENTO, foi utilizado o padrão STRATEGY na construção da classe *ConhecimentoStrategy*, pela qual o *framework* pode implementar a solução utilizando a classe *Exemplo*, *Metáfora* e *SubApresentação*, respectivamente únicas ou associadas. Cabe esclarecer que, as classes *Exemplo* e *Metáfora* compõem-se da classe *Recurso*, a qual foi identificada a partir da análise do documento de requisitos do Apêndice A.

Ainda considerando a classe *Conhecimento*, a mesma pode relacionar-se à classe *Atividade*, a qual constitui uma estratégia do padrão USAR ATIVIDADES.

Mesma situação da classe *Conhecimento* pode ser identificada na classe *Layout*, a qual por meio do padrão *LayoutStrategy*, pode implementar as variações a partir das classes *Fonte* e *Fundo*.

A classe *Tópico*, unidade da classe *Conteúdo*, relaciona-se às classes *Layout* e *Navegação* e é composta pela classe *Conhecimento*. Estes relacionamentos envolvem a possibilidade de serem configuradas a navegação e o layout de cada tópico dentro da aula virtual.

Em se tratando da classe *Avaliação*, originária do padrão AVALIAR ALUNO, podem ser escolhidas duas formas de aplicação: a avaliação por desempenho e avaliação por questões. A avaliação por desempenho considera alguns aspectos do perfil do aluno, os quais são obtidos a partir do relacionamento entre as classes *AulaVirtual* e *Aluno*. A avaliação por questões, representada pela classe *AvalFormativa*, é composta pela classe *Questão*, a qual pode implementar diversos tipos de questões para uma avaliação por meio das classes *Alternativa*, *Ordenação*, *VerdadeiroFalso* e *Cálculo*.

A maior e mais importante unidade de implementação do modelo de classes é representada pela classe *AulaVirtual*, a qual é composta pela classe *Conteúdo* e mantém relacionamento com as classes *Aluno*, *Pré-Requisitos* e *Disciplina*. Esta classe constitui o foco do desenvolvimento do *framework* aqui proposto.

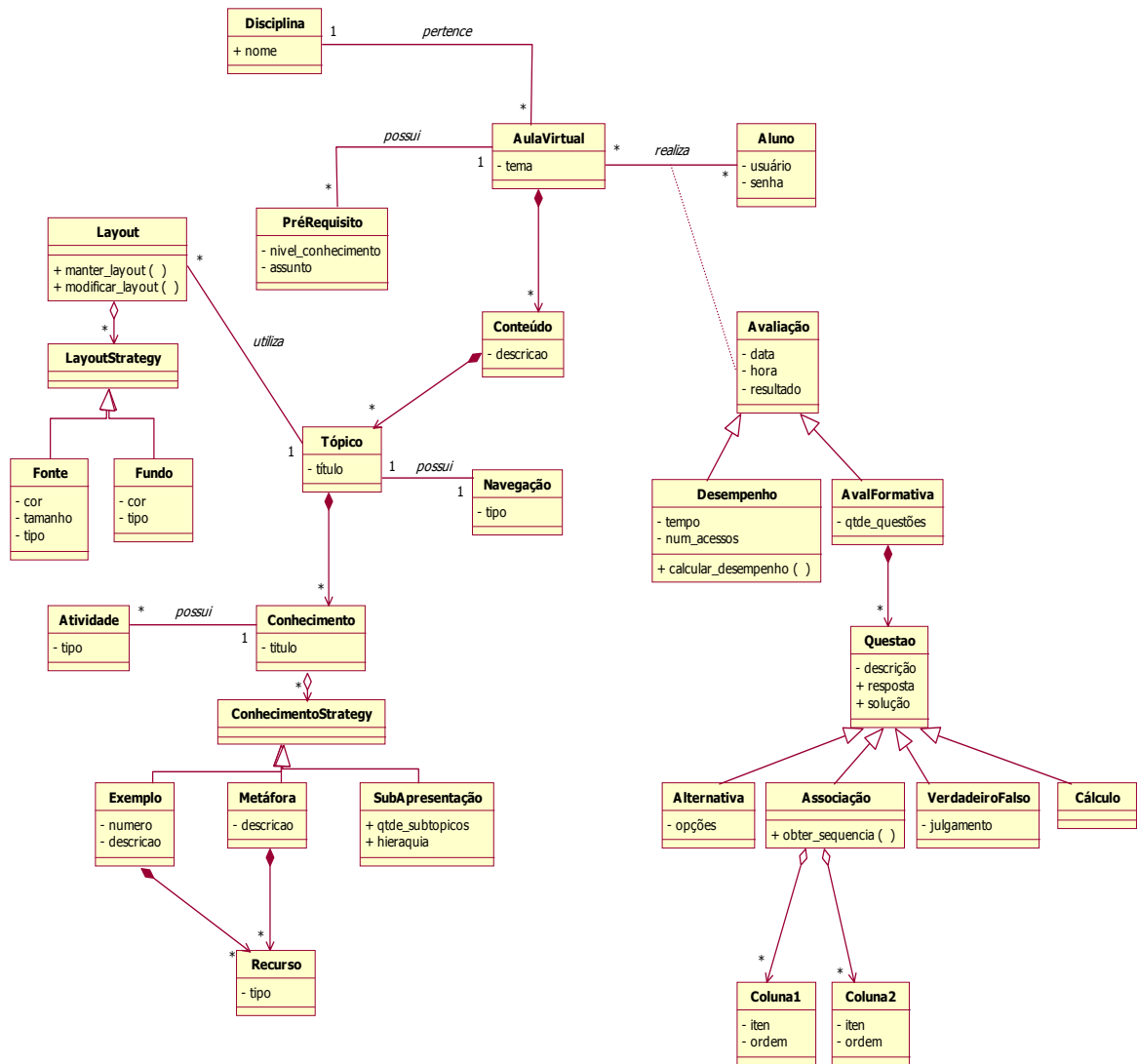


Figura 4-3 Diagrama de Classes de Projeto do “FCAVirtu”

Algumas das classes pertencentes à hierarquia do *framework* “FCAVirtu”, tais como a classe *Aluno*, *Disciplina* e *Recurso*, são provenientes de análise e identificação de pontos variáveis e de implementação do *framework*. Essas classes, mesmo não sendo originárias da linguagem de padrões na qual o *framework* foi baseado, constituem importância em sua hierarquia em questão da geração de aulas virtuais e foram abstraídas de identificações realizadas por análises no documento de requisitos e diagrama de caso de uso do domínio, estes respectivamente apresentados nos Apêndice A e B.

#### 4.7. Validação do Projeto do Framework

Em análise ao processo de desenvolvimento de *framework* proposto por Braga (2002), este trabalho realiza a inclusão de um sub-passo ao passo de projeto do *framework*.



Esse sub-passo, identificado como “Validação do Projeto do *Framework*”, consiste na realização de uma validação parcial do *framework* que está sendo construído a partir do mapeamento das classes definidas no projeto em relação aos padrões existentes na linguagem de padrões.

O passo consiste, basicamente, na análise dos padrões e mapeamento dos mesmos a fim de que sejam verificadas todas as possibilidades de aplicação desses padrões, existentes na linguagem de padrões, às classes definidas no processo de projeto.

Para auxiliar no desenvolvimento deste sub-passo recursos como tabelas ou afins podem ser utilizados para efetuar o mapeamento.

A intenção em incluir esse sub-passo ao processo em questão se justifica, principalmente, pelo fato do processo normal de validação proposto por Braga (2002) ocorrer posteriormente à fase de implementação do *framework*, a qual não será aqui aplicada considerando o escopo que foi delimitado para este trabalho.

<b>Padrão / Sub-Padrão da “LPCAVirtu”</b>	<b>Classe</b>	<b>Comentários</b>
PLANEJAR AULA	Classe AulaVirtual	O planejamento da aula, a possibilidade de adaptar a aula ao contexto do aluno e de delimitá-la de acordo com o nível de conhecimento do mesmo em relação ao tema tratado, são definições abrangidas pela classe Aula Virtual.
ADAPTAR AO ALUNO	Classe AulaVirtual	
DELIMITAR AULA	Classe Aula Virtual	
EXPLORAR CONHECIMENTO	Classe Conteúdo	As formas de estruturação da aula, considerando o Conhecimento como unidade básica, serão abrangidas pela classe Conteúdo.
ESTRUTURAR AULA	Classe Conteúdo	
DEFINIR PRÉ-REQUISITOS	Classe PréRequisito	Cada pré-requisito identificado pelo professor como necessário à aula poderá ser descrito por meio da respectiva classe.
RELACIONAR CONTEÚDO	Classe Conhecimento	Os padrões deste item estão expressamente abrangidos pela classe Conhecimento, a qual utiliza estratégias de solução principalmente para o padrão ESTENDER CONHECIMENTO.
ESTENDER CONHECIMENTO	Classe Conhecimento	
USAR ATIVIDADES	Classe Atividade	A classe atividade implementa o padrão em questão para que o professor utilize, em cada unidade de conhecimento, atividades de desenvolvimento do aprendizado.
FECHAR TÓPICO	Classe Tópico	O fechamento de tópicos é implementado na classe respectiva.
REALIZAR REVISÃO	Classe Atividade	A utilização de revisões, considerando os conhecimentos apresentados, pode ser tratada como

		uma atividade.
FINALIZAR AULA	Classe Aula Virtual	Recursos para finalização da aula podem ser abrangidos pela classe de implementação da aula virtual.
EDITAR APRESENTAÇÃO	Classes Layout e Navegação	O padrão Editar Apresentação, por ser conceitual, está implementado nas classes de Layout e Navegação.
ESQUEMATIZAR CONTEÚDO	Classe Conteúdo	Considerando-se o conteúdo como um conjunto de tópicos, esse padrão apresenta formas de esquematizá-los por meio da classe Conteúdo.
CONFIGURAR NAVEGAÇÃO	Classe Navegação	Trata-se da implementação da flexibilidade na apresentação e decorrer da aula virtual, a qual é representada pela classe Navegação.
AVALIAR ALUNO	Classe Avaliação	Sub-dividida em duas possíveis formas de implementação essa classe apresenta padrão incluso na linguagem nesta pesquisa.
EXEMPLIFICAR	Classe Exemplo	Esses padrões possuem em seu mapeamento como extensão a classe Recurso
METAFORIZAR	Classe Metáfora	
SUB-APRESENTAR	Classe Sub-Apresentação	A estratégia de sub-dividir unidades de conhecimento em níveis menores é mapeada para a classe Sub-Apresentação.
AVALIAR_POR_QUESTÕES	Classe AvalFormativa	O sub-padrão aqui apresentado tem como mapeamento principal a ramificação pela classe Questão
AVALIAR_POR_DESEMPENHO	Classe Desempenho	Sem indicações de especializações ou composições esse mapeamento considera a classe Desempenho como em relação ao perfil do aluno.

**Quadro 4-2 Tabela de Mapeamento dos Padrões / Sub-Padrões da linguagem “LPCAVirtu” em relação às classes do *framework***

Para o contexto da validação da análise e projeto do *framework* “FCAVirtu” é utilizada a estrutura de tabela para mapeamento dos padrões em relação às classes conforme pode ser acompanhado pelo Quadro 4.2.

A partir da utilização do mapeamento dos padrões para as classes do *framework* foi possível reconhecer situações distintas as quais envolvem padrões provenientes da análise de requisitos do domínio obtidos a partir do documento de requisitos (Apêndice A). A interligação de diversos padrões a uma mesma classe, como é o caso dos padrões PLANEJAR AULA, ADAPTAR AO ALUNO e DELIMITAR AULA mapeados para a classe *Aula Virtual*, constitui fato a ser denotado no mapeamento, assim como ressalta-se o fato de padrões como o EDITAR APRESENTAÇÃO, os quais relacionam-se a mais de uma classe, neste caso as classes *Layout* e *Navegação*, constituir um padrão de abrangência variada.

## 4.8. Considerações Finais

Neste capítulo foi apresentada a análise e o projeto do *framework* “*FCAVirtu*”, baseado na linguagem de padrões definida na Seção 3.3., a “*LPCAVirtu*”. As atividades para o desenvolvimento do *framework* proposto seguiram um processo de desenvolvimento de *framework* sistematizado e validado, definido por Braga (2002), e apresentado na Seção 2.6.1.

Na Seção 4.2 discorreu-se acerca da aplicação do processo no desenvolvimento do *framework* baseado na linguagem de padrões “*LPCAVirtu*” e da abrangência desta pesquisa em relação à aplicação do processo definido por Braga (2002). Nas Seções 4.3 e 4.4 respectivamente, foram comentados os passos de análise e projeto do *framework* pertencentes ao processo de desenvolvimento utilizado. Ressalta-se ainda que a Seção 4.4.1 tratou da inclusão de um sub-passo ao processo de desenvolvimento de *framework* definido por Braga (2002), o qual refere-se à validação do *framework* considerando somente a execução dos dois passos iniciais desse processo, como é o caso deste trabalho.

Por fim, foi apresentada a estrutura do *framework* “*FCAVirtu*”, juntamente com a sua hierarquia de classes, respectivamente nas Seções 4.5 e 4.6, os quais constituíram os resultados obtidos com a realização dos passos de análise e projeto do processo de Braga (2002).

O contexto de desenvolvimento do *framework* aqui proposto constitui, por si só, uma das dificuldades em construí-lo. Isso, inicialmente, se dá pelo fato do desenvolvimento ocorrer somente até à fase de projeto considerando a abrangência dessa pesquisa. Outro fato surge da questão da identificação e apresentação das variabilidades existentes para o *framework*, as quais não poderão ser identificadas em seu todo considerando a limitação imposta em seu desenvolvimento somente até a fase de projeto.

## **CAPÍTULO 5. CONCLUSÕES**

### **5.1. Considerações Iniciais**

Neste capítulo são apresentados os resultados obtidos a partir das pesquisas realizadas neste trabalho, juntamente com as contribuições fornecidas à área científica mediante a realização do mesmo. Complementarmente, são denotadas nas Seções 5.3 e 5.4, respectivamente, as limitações encontradas neste trabalho e os possíveis trabalhos futuros dele decorrentes.

### **5.2. Resultados Obtidos e Contribuições**

Considerando todas as pesquisas realizadas durante a realização deste trabalho os resultados e contribuições obtidas com o mesmo podem ser claramente explanados, sendo que os objetivos pré-definidos foram todos abordados.

Primeiramente, revela-se como produção desta pesquisa a extensão e adaptação da linguagem de padrões para o contexto do *e-Learning*, a “*CogLearn*” (apresentada na Seção 3.2), por meio da construção da linguagem “*LPCAVirtu*” (Seção 3.3), levando em consideração a adaptação dos padrões já existentes na linguagem pesquisada e a inclusão de novos padrões, em especial referentes à avaliação do aluno, ainda inexistente em sua estrutura. A identificação de padrões, sub-padrões e a navegabilidade entre os mesmos por meio da construção de um grafo de fluxo de aplicação dos padrões dentro da linguagem proposta facilitou a análise e o projeto de um *framework* nela baseado, o qual também constitui produto deste trabalho.

Uma vez definida a linguagem de padrões aqui proposta e a sua estrutura de padrões prosseguiu-se com a análise e desenvolvimento de um *framework* caixa-branca, o “*FCAVirtu*” (apresentado na Seção 4.2), por meio de um processo de desenvolvimento de *framework* sistematizado e validado definido pelo trabalho de Braga (2002).

Com a utilização do processo de desenvolvimento de *framework* definido por Braga (2002), esta pesquisa ainda permitiu detectar a necessidade da inclusão de um sub-passo a esse processo de desenvolvimento que trata da validação da fase de projeto do *framework* sendo construído. A inclusão deste sub-passo se fez pela necessidade de realizar uma

validação durante o processo de construção do framework, o qual neste trabalho só abrange os passos de análise e projeto, passos estes que não relacionam validação alguma ao se considerar o processo proposto por Braga (2002).

Considerando o contexto de desenvolvimento do framework “*FCAVirtu*” foram ainda explicitados os pontos variáveis do mesmo a fim de apresentar flexibilidade na sua instanciação em relação às aplicações com ele produzidas.

Outro objetivo alçando que também constitui uma contribuição deste trabalho à área científica corresponde à utilização de linguagens de padrões e *frameworks* para a construção de módulos educacionais, a qual, como introduzido na Seção 2.2, ainda é pouco aplicada.

### 5.3. Limitações do Trabalho

Assim como foram levantados os resultados obtidos com o decorrer deste trabalho, limitações também foram encontradas, sendo que as mesmas também são aqui descritas.

Em questão da linguagem de padrões produzida, não foram considerados todos os processos propostos na literatura para a avaliação do aluno, devido à extensão deste trabalho e ao envolvimento direto de questões pedagógicas, as quais não constituíam foco ao tema.

Além disso, ainda considerando a produção da linguagem de padrões, não fica garantido que a abrangência da mesma seja completa, uma vez que possam existir, de acordo com o contexto tratado, novos padrões ou variações não identificadas neste trabalho.

Considerando as fases de análise e projeto do *framework* proposto, ressalta-se que as mesmas foram baseadas somente nos grupos de funcionalidades identificados a partir da linguagem de padrões da qual este foi baseado, não considerando o desenvolvimento das camadas de persistência e interface gráfica com o usuário presentes na arquitetura do *framework* (conforme apresentado na Seção 4.5).

Outro aspecto limitador desta pesquisa refere-se à não implementação do *framework* projetado, não sendo, portanto, possível realizar a sua validação.

### 5.4. Trabalhos Futuros

A partir do desenvolvimento desta pesquisa, novos trabalhos podem ser realizados considerando os resultados obtidos e as limitações levantadas.

Como possíveis trabalhos futuros podem ser citados:

- Refinamento da linguagem de padrões proposta, por meio da identificação de novos padrões ou inclusão de sub-padrões e variantes de aplicação.
- Refinamento do projeto do *framework* desenvolvido, considerando a inclusão das camadas de persistência e interface gráfica com o usuário.
- Implementação e validação do *framework* para complementar de forma significativa a extensão do trabalho aqui realizado.
- Realização do teste de aceitação do usuário, neste caso o professor, em relação à interface gráfica e à qualidade das aulas virtuais geradas a partir da utilização do *framework*. Isto abrangendo como alvo a facilidade de instanciação do *framework*, ou seja, de criar as aulas a partir dele.

## REFERÊNCIAS

BARBOSA, E. F.; MALDONADO, J.C. **Padronização de processo para o desenvolvimento de Módulos Educacionais**. ICMC – USP, 2003.

BERTOLLO, G.; FALBO, R. A. **Apoio Automatizado à Definição de Processos de Software em Níveis**. In: Simpósio Brasileiro de Qualidade de Software. UFES – Universidade Federal do Espírito Santo. Vitória, 2003.

BRAGA, R.T.V; GERMANO, F.S.R.; MASIERO, P.C.; MALDONADO, J. C. **Introdução aos padrões de software**. Notas Didáticas. Instituto de Ciências Matemáticas e de Computação. USP – Universidade Federal do São Paulo. São Carlos, 2001.

BRAGA, R. T. V. **Um Processo para Construção e Instanciação de Frameworks baseados em uma Linguagem de Padrões para um Domínio Específico**. Tese de Doutorado. Instituto de Ciências Matemáticas e de Computação. USP – Universidade Federal do São Paulo, 2002.

BRAGA, R. T. V.; MASIERO, P.C. **Finding Framework Hot Spots in Pattern Languages**. Journal of Object Technology. vol. 3, nº 1, janeiro-fevereiro 2004. p. 123-142.

CAGNIN, M. I. **PARFAIT**: uma contribuição para a reengenharia de software baseada em linguagens de padrões e frameworks. Tese de Doutorado. Instituto de Ciências Matemáticas e de Computação. USP – Universidade de São Paulo. São Carlos, 2005.

CAMPOS, F., CAMPOS, G., ROCHA, A. R. **Dez etapas para o desenvolvimento de software educacional do tipo hipermídia**. In: III Congresso Ibero-americano de Informática Educativa. RIBIE, 1998.

CANTARELLI, E. M. P. **Software Educacional**. Monografia de Pós-Graduação. URI – Universidade Regional Integrada do Alto Uruguai e das Missões. RS, 2004.

CATALDI, Z. **Metodología de diseño, desarrollo y evaluación de software educativo**. Tese de Mestrado. UNLP. UBA, Argentina, 2000.

COMASSETTO, L. S. **Novos espaços virtuais para o ensino e a aprendizagem à distância**. Tese de Doutorado. UFSC, 2006.

COPLIEN, J.O. **Software Design Patterns: Common Questions and Answers**. The Patterns Hand Book: Techniques, Strategies and Applications, p 311-320. Cambridge University Press, New York, 1998.

COSTA, L.A.C; FRANCO, S.R.K. **Ambientes Virtuais de Aprendizagem e suas possibilidades construtivistas.** CINTED. UFRGS, 2005.

CRUZ, S.O. **Um Catálogo de Padrões para a Construção de Sistemas O.O.** Dissertação de Mestrado. PUC, São Paulo, 1998.

DUTRA, R.L.S; TAROUCO, L.M.R. **Objetos de Aprendizagem:** Uma Comparação entre SCORM e IMS – Learning Designing. CINTED. UFRGS, 2006.

FAYAD, M. E.; JOHNSON, R. E. **Domain-specific application frameworks:** Frameworks experience by industry. First ed. John Wiley & Sons, 2000.

FENNER, R. C. **Contribuições Do Design Na Produção De Software Educacional.** Tese de Mestrado. PPG-EP. UFSC, 2000.

FERNANDES, J.; LEGOINHA, P.; PAIS, J. **O Moodle e as comunidades virtuais de aprendizagem.** VII Congresso Nacional de Geologia. Universidade de Évora, Portugal, 2005.

FISCHER, M. C. B. O. **Estudo de Requisitos para um Software Educativo de Apoio ao Ensino da Introdução à Computação.** Dissertação (Mestrado em Ciência da Computação) – Curso de Pós-Graduação em Computação, Universidade de São Paulo, São Paulo, 2001. Disponível em: <[http://fesppr.br/~lecheta/Projeto%20Final/Requisitos/ estudo.pdf](http://fesppr.br/~lecheta/Projeto%20Final/Requisitos/estudo.pdf)>. Acesso em: 01/05/2006.

FUGGUETA, A. **Software process:** a Roadmap. In: 22<sup>nd</sup> International Conference on the Future of Software Engineering. Limerick, Ireland, ACM, 2000.

FUKS, H. **Aprendizagem e Trabalho Cooperativo no Ambiente AulaNet.** Monografias em Ciência da Computação. PUC, Rio de Janeiro, 2000.

FUKS, H.; GOÑI, J.L. **Comparação de ambientes de ensino na Web baseados em plataformas IMS a partir dos papéis dos atores envolvidos.** PUC-RioInf.MCC14/02. PUC, Rio de Janeiro, 2002.

FUKS, H.; RAPOSO, A. B., GEROSA, M. A.; LUCENA, C. J. P. **Applying the 3C Model to Groupware Development,** International Journal of Cooperative Information Systems (IJCIS), v.14, World Scientific, 2005.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. **Design Patterns:** Elements of Reusable Object-Oriented Software. Addison Wesley, 1994.

GOMES, A. S.; WANDERLEY, E. G. **Elicitando requisitos em projetos de Software Educativo.** In: WIE'2003, Campinas - SP.



GOMES, F. D. **PARFAIT/EA – Processo Ágil de Desenvolvimento Baseado em Framework**. Dissertação de Mestrado. PPG-CC – Univem, Marília, 2007.

ICMC – USP. Grupo de Engenharia de Software do Instituto de Ciências Matemáticas e da Computação da Universidade de São Paulo. **Notas Didáticas**. ICMC, 2007.

JOHNSON, R.E.; FOOTE, B. **Designing Reusable Classes**. Journal of Object Oriented Programming - JOOP, jun/jul, 1991.

KEMCZINSK, A. **Métodos de Avaliação para Ambientes e-Learning**. Tese de Doutorado. UFSC, 2005.

KRATZ, R. A. **Fábrica de adequação de conteúdo de ensino para Objetos de Aprendizagem Reutilizáveis (RLOs) respeitando a norma SCORM**. Dissertação de Mestrado. UNISINOS, São Leopoldo, 2006.

LEIVA, W. D. **Um modelo de hipertexto para apoio ao ensino mediado pela Web**. Tese de Doutorado. ICMC. USP, 2003.

LUCENA, C. J. P.; FUKS, H.; CUNHA, M. L.; GEROSA, M. A. **Participação e Avaliação no Ambiente Virtual AulaNet da PUC-Rio**, in: Silva, M.; Educação Online: Teorias, Práticas, Legislação e Formação Corporativa; Edições Loyola, Rio de Janeiro, 2003. Disponível em < <http://www.les.inf.puc-rio.br/groupware> >

LUCENA, C.J.P.; FUKS, H. BARRETO, C.G. **Agregando Frameworks em uma Arquitetura Baseada em Componentes: Um Estudo de Caso no Ambiente AulaNet**. WDBC 2005, Juiz de Fora-MG. Disponível em < <http://www.les.inf.puc-rio.br/groupware> >

LUCIANO, N.A.; VIEIRA, M.B. **Construção e Reconstrução de um Ambiente de Aprendizagem para Educação à Distância**. Coletânea de Textos, ABED, 2002.

MAIA, A.B.; FREITAS, A.V.P.; NUNES, D.J. **Um modelo para interação entre processos de software**. Artigo. PROSOFT. UFRGS, 2004.

MALANGE, F.C.V. **Uma proposta de aprendizado da lógica utilizando um ambiente virtual de aprendizagem colaborativa**. Dissertação de Mestrado. PPG-CC. UFSC, 2005.

MALDONADO, J. C; BRAGA, R.T.V; GERMANO, F.S.R.; MASIERO, P.C.;. **Padrões e Frameworks de Software**. Notas Didáticas. ICMC. USP, 2004.

MARÇAL, V.S.P. **Adaptação de Conteúdos Educacionais Baseada no Modelo de Aluno do Ambiente Adaptweb**. Artigo. UFRGS, 2003.

NETO, W. O. **Estudo e análise de um modelo de referência para conteúdos e ambientes de ensino e aprendizagem à distância: SCORM**. Trabalho de Conclusão de Curso. UFL, MG, 2004.

NETO, A. T. **Linguagem de Padrões para Apoiar o Projeto de Material Instrucional para EAD**. Dissertação de Mestrado. PPG-CC. UFSCAR, 2005.

NETO, A. T.; SILVA, J. C. A.; NERIS, V.P.A. **Padrões para Apoiar o Projeto de Material Instrucional para EAD**. Anais do SugarLoafPLOp 2005, Campos do Jordão, 2005. v.1. p. 1-16.

NETO, A. T.; SILVA, J. C. A.; NERIS, V.P.A. NETO, A. T. **CogLearn: Uma Linguagem de Padrões para e-Learning**. Revista Brasileira de Informática na Educação. Rio de Janeiro, 2006. v.3., p. 33-50.

NETO, A. T.; SILVA, J. C. A.; NERIS, V.P.A. **Framework baseado na Linguagem de Padrões Cog-Learn para apoio a criação de objetos de aprendizagem**. TIDIA-Ae, FAPESP, 2006.

OLIVEIRA, A.V. **Construção de Ambientes Virtuais de Aprendizagem baseados na Internet: utilizando recursos gratuitos**. Dissertação de Mestrado. PPG-EP. UFSC, 2001.

OLIVEIRA, J.P.M.; BRUNETTO, M.A.O.C.; PROENÇA JR. M. L.; PIMENTA, M.S; RIBEIRO, C.H.F.P.; LIMA, J.V. **AdaptWeb: um ambiente para ensino-aprendizagem adaptativo na Web**. Revista Educar - Especial, Editora UFPR, 2003. p. 175-197.

PEREIRA, L. A. M. **Objetos de Aprendizagem Reutilizáveis (RLOs): Conceitos, Padronização, Uso e Armazenamento**. PUC, Rio de Janeiro, 2003.

PRESSMAN, R.S. **Engenharia de Software**. São Paulo: Makron Books, 1994.

PRIMUS, N.J.C. **A generic framework for evaluating Adaptive Educational Hypermedia authoring systems**. Tese de Mestrado. University of Twente, 2005.

REIS, C.R. **Caracterização de um processo de software para Projetos de Software Livre**. Dissertação de Mestrado. ICMC – USP, 2003.

SANTOS, N. **Desenvolvimento de Software Educacional**. Notas de Aula. IME, UERJ, 1999. Disponível em < [www.ime.uerj.br/~neide/](http://www.ime.uerj.br/~neide/) >

SANTOS. E. O. **Ambientes virtuais de aprendizagem: por autorias livre, plurais e gratuitas**. In: Revista FAEBA, v.12, 2003.

SCHLINDWEIN, L.M.; SEARA, E.F.R.; BENITTI, F. B.V. **Processo de Desenvolvimento de Software Educacional**: proposta e experimentação. CINTED, UFRGS, 2005.

SISSON, G. A. **Gerenciador de Alertas para um Ambiente de Ensino a Distância na Internet**. Trabalho de Diplomação. UFRGS, 2003.

SOMMERVILLE, I. **Engenharia de Software**. 6ª ed. São Paulo: Pearson Education do Brasil, 2003.

VAHL Jr., J.C. **Avaliação de um Modelo Computacional para Unidades Educacionais Multimídia**. Trabalho de Conclusão de Curso. Universidade Federal de Pelotas, 2000.

VAZ, D.M. **Construção de um sistema de workflow em um ambiente de ensino à distância utilizando padrões abertos e software livre**. UFRGS, 2003.

YODER, J. W. **Software Patterns Tutorial**. Software Patterns for SugarLoafPlop, Itaipava, RJ, 2002.

VAZ, M.F.R.; BRESSAN, G. **Padrões Emergentes de Ensino à Distância**. Intertech,2002.

VIERO, D. M. **Construção de um sistema de *workflow* em um ambiente de ensino à distância utilizando padrões abertos e software livre**. UFRGS, 2003.

# APÊNDICE A – DOCUMENTO DE REQUISITOS

## Softwares Educacionais de Apoio ao Processo de Ensino-Aprendizagem

### 1. Visão Geral do Sistema

Os softwares educacionais de apoio ao ensino-aprendizagem devem abordar conteúdos específicos da disciplina a que forem destinados, seja ela qual for, de forma que, ao serem aplicados em atividades de aprendizagem possam levar à produção do conhecimento na área envolvida. Os principais usuários destes softwares são, em geral, alunos de ensino fundamental e médio, graduação e até mesmo de cursos tecnológicos mediante a disciplina e o seu contato com a aprendizagem de conceitos a ela pertinentes. Esses softwares têm que englobar, quase de forma obrigatória, as atividades de aquisição de conhecimento, prática e avaliação de aprendizado, as quais garantem a sua eficiência quando aplicados ao processo de ensino-aprendizagem em qualquer que seja o seu domínio de aplicação.

### 2. Requisitos Funcionais

#### A) Aluno

- 1) O sistema deve ser capaz de identificar, por meio de dados de *login* ou semelhante, o conteúdo a ser acessado pelo aluno;
- 2) A interface pode permitir ou não ao aluno a escolha do tema a ser tratado na utilização do software. No caso de permissão, o usuário-aluno entrará com o termo respectivo à área de conhecimento ou terá a possibilidade de selecionar este termo e o sistema lhe trará todas as aulas /informações necessárias para executar os módulos posteriores;
- 3) Serão disponibilizadas as funções básicas de busca e seleção de conteúdos, realização de exercícios e consulta de desempenho individual;
- 4) Na interface de exercícios pode ainda ser possível a divisão por níveis, os quais estarão relacionados diretamente ao estágio de conhecimento do aluno;

## B) Professor

- 5) O sistema deve ser capaz de identificar, por meio de dados de *login* ou semelhante, o conteúdo a ser acessado pelo professor;
- 6) As funções de área de professores estarão baseadas na inclusão de exercícios e conteúdos que “alimentarão” o sistema. Esta inclusão ocorrerá em telas específicas.
- 7) O software contará com as funções básicas de inclusão de material e exercícios e consulta de desempenho de alunos;
- 8) O sistema deve fornecer o relatório de aproveitamento do aluno ou de uma turma contendo, de alguma forma, a quantidade de exercícios realizados, o nº de acertos e o nº de erros de cada exercício ou de cada aluno, dependendo do contexto individual ou em grupo;
- 9) Pode-se ainda, quando tomado o aspecto de geração de relatórios em grupo fornecer ao professor o desempenho dos alunos através de medidas estatísticas que o caracterizem (nº de usuários, nº de acertos, média,...);
- 10) O editor de exercícios deve disponibilizar ao professor campos para inserção de pergunta e de suas possíveis respostas;
- 11) O sistema deve fornecer recursos que proporcionam liberdade de projeto dos aspectos navegacionais da aplicação além da possibilidade de mapeamento direto do modelo conceitual para o projeto da navegação, o professor deve poder utilizar o modelo conceitual como uma orientação para projetar a aplicação, ou seja, os aspectos navegacionais da aplicação devem poder ser modelados de acordo com as preferências do professor, não estando vinculados somente à fase de modelagem do domínio de conhecimento;
- 12) O sistema deve fornecer recursos que permitam o estabelecimento do tipo de controle sobre a navegação: o professor deve poder determinar "quanto" de controle sob a navegação da aplicação deve ser atribuído ao aluno, no sentido de perseguir um objetivo em relação a uma proposta didática subjacente. Ele deve poder estabelecer se o aluno terá acesso a todos os nós da aplicação ou somente a alguns deles. Isso pode ser feito definindo-se o grau de liberdade (total, parcial ou nenhum) associado a cada nó;
- 13) O sistema deve fornecer recursos que permitam fixação de material específico: o professor - autor da aplicação - deve poder designar material específico para estudantes ou grupos de estudantes. Esses recursos possibilitam que questões

relacionadas à prática pedagógica possam ser conservadas ao utilizar-se um sistema de apoio ao ensino-aprendizagem;

- 14) O sistema deve fornecer recursos que possibilitam inclusão de material essencial e complementar: no caso de aplicações para ensino, é desejável uma estrutura que inclua o material essencial ou relevante, sem eliminar o secundário, desde que este não seja conflitante com o anterior e que possa contribuir para os objetivos do aluno. A implementação desses recursos é interessante em aplicações para ensino, porque possibilita que a flexibilidade das informações contidas na aplicação seja projetada para fornecer material adicional, que será apresentado somente se o aluno requisitá-lo.

### **3. Requisitos não-funcionais**

#### *15) Facilidade de aprendizagem*

O sistema deve prover ao usuário-aluno garantias de sucesso no aprendizado de conteúdos trabalhados de forma subjetiva. Ao utilizar o sistema, de forma implícita, o aluno já deve estar interado no processo de ensino-aprendizagem;

#### *16) Baixa ocorrência de erros*

Por ser um sistema que visa o aprendizado, deve-se evitar ao máximo a ocorrência de erros que podem levar o aluno à dúvida ou à aquisição de conteúdo de forma errada;

#### *17) Eficiência*

O sistema deve mostrar-se eficiente no processo de ensino-aprendizagem de qualquer conteúdo educacional, seja ele qual for, obtendo os mesmos resultados de processamento e qualidade;

#### *18) Compatibilidade com o mundo real*

O sistema deve tentar apresentar o conteúdo ao aluno de forma clara e objetiva, numa linguagem semelhante à utilizada em sala de aula a fim de que o aluno possa situar-se com maior facilidade na sua utilização;

### *19) Help e Documentação*

Por mais fácil que seja a interface do sistema este obrigatoriamente deve apresentar alguma forma de documentação e acesso a tutoriais ou *helps* de ajuda ao usuário em todas as suas funcionalidades;

### *20) Navegabilidade*

É de fundamental importância que o sistema apresente formas avançadas de navegabilidade e interconexão de telas em todas as suas funcionalidades, isso facilita a interconectividade e inter-relacionamento do usuário com o próprio sistema;

### *21) Usabilidade*

O sistema deve caracterizar-se por ser de fácil utilização e correto de tal modo que sua utilização não culmine a erros.

### *22) Tempo de resposta*

Por ser um sistema interativo e de cunho educacional o mesmo não pode dispor de grande tempo de resposta, deve-se procurar minimizar o tempo de resposta de todas as ações que envolvem o usuário em especial, o usuário-aluno.

## **4. Glossário de termos do domínio**

***Login:*** forma de acesso a um programa através da identificação composta, em geral, por usuário e senha;

***Desempenho dos alunos:*** quantidade de aprendizado obtido pelos alunos com a utilização do software;

***Projeto de navegação:*** esquemas que favorecem a disponibilidade de navegação pelos conteúdos presentes em cada tela do programa;

***Grau de liberdade:*** refere-se à “quantidade” de acesso que pode ser disponibilizada ao aluno através das telas do sistema;

## APÊNDICE B – DIAGRAMA DE CASOS DE USO

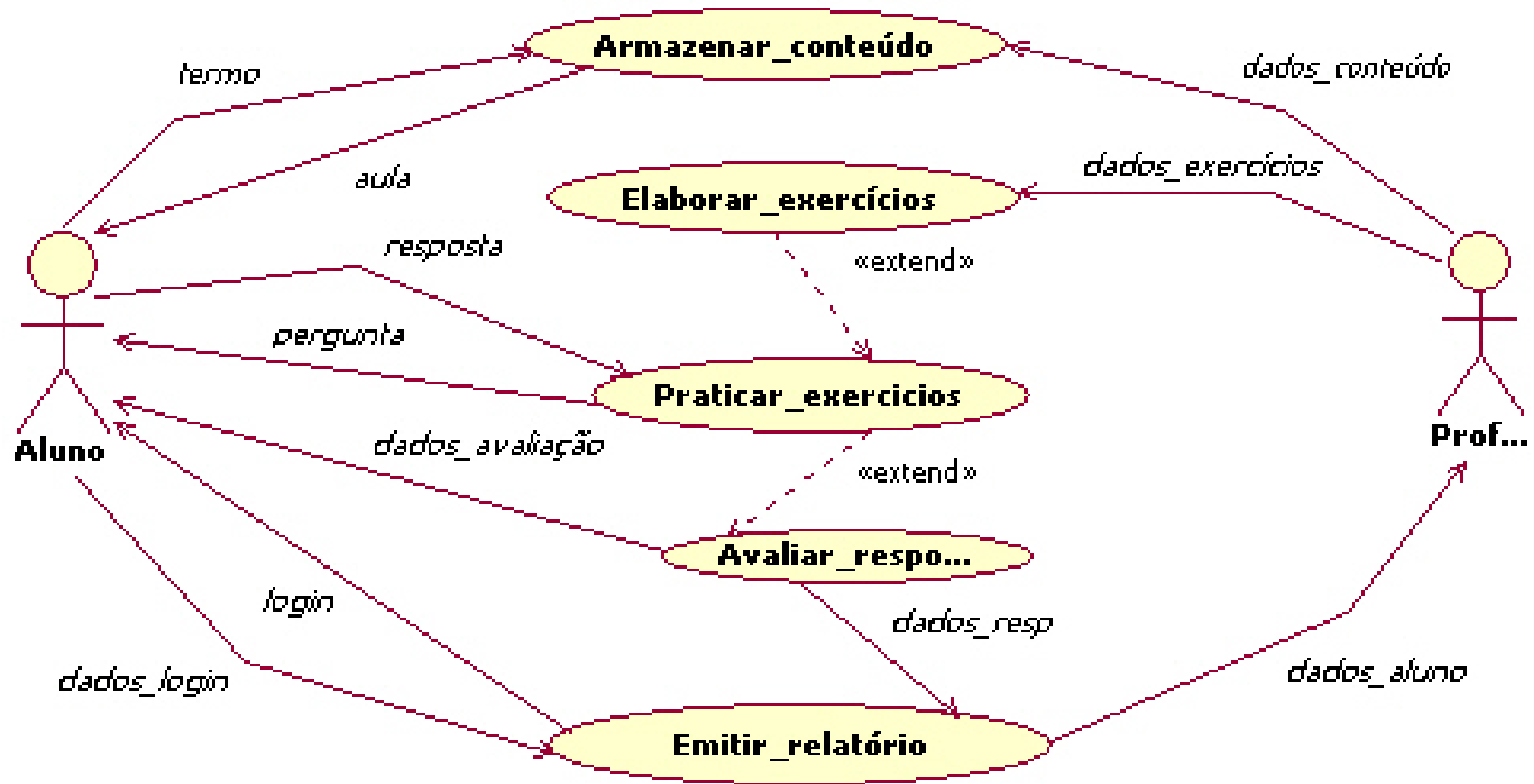


Figura 5-1– Diagrama de Caso de Uso genérico para um software educacional baseado em documento de requisitos (Apêndice A)



## APÊNDICE C – PADRÕES CONSTITUINTES DA “LPCAVIRTU”

Os padrões abaixo descritos constituem uma reunião de diversos padrões: alguns já existentes no âmbito do *e-Learning*, outros sendo uma adaptação dos padrões utilizados por Neto (2005) em seu trabalho sobre a linguagem de padrões “*CogLearn*”, apresentada nesta pesquisa, além de padrões inexistentes na literatura pesquisada, os quais estão sendo incluídos neste trabalho como contribuição ao desenvolvimento da EAD. Alguns padrões da linguagem “*LPCAVirtu*” já foram descritos na Seção 3.3.1 e por este motivo não estão relacionados neste Apêndice.

### 1. PLANEJAR AULA

**Problema:** Qual a melhor forma de estruturar a aula para que todos os conceitos necessários sejam transmitidos ao aluno?

**Contexto:** Os conceitos que o professor deseja transmitir ao aluno sobre determinado tema em uma aula virtual devem estar estruturados. A estruturação permite que o aluno compreenda o tema estudado e o motive, além de que, até mesmo em uma aula presencial o planejamento caracteriza uma fase crucial para a condução de uma boa aula.

**Solução:** Uma vez definido o tema e os sub-temas a serem tratados durante uma aula, o professor deve utilizar algum recurso que permita estruturar a aula seqüenciando o conteúdo a ser apresentado e /ou explorado pelo aluno. Isto pode ser realizado a partir de um Mapa de Conceitos ou de uma estrutura semelhante.

**Contexto Resultante:** o padrão Planejar Aula auxilia na organização da aula e na transmissão de conceitos ao aluno seja pela sua própria exploração ou mesmo pela apresentação tópico a tópico.

### 2. ADAPTAR AO ALUNO

**Problema:** De que forma uma aula deve ser apresentada para que o aluno possa motivar-se a compreendê-la?

**Contexto:** Durante uma aula, mesmo sendo esta uma aula virtual, o aluno busca relacionar os conceitos apresentados com aplicações efetivamente práticas que possam auxiliá-lo nos problemas reais.

**Solução:** Na elaboração do material de uma aula, o professor deve evitar o uso de recursos que não sejam familiares ao aluno. Desde a linguagem utilizada até os recursos multimídia, tudo deve estar adaptado à realidade e contexto do aluno.

**Contexto Resultante:** O padrão adaptar ao aluno se faz importante pelo fato de constituir-se como ponte entre o conhecimento e o aluno. Esta ponte é formada pelos recursos comuns já reconhecidos pelo aluno e presentes em seu contexto de aprendizagem.

### ***3. EXPLORAR CONHECIMENTO***

**Problema:** Quais os recursos que podem ser utilizados para efetivar a exploração de conhecimento por parte do aluno?

**Contexto:** Para que o aluno explore o conhecimento que vem adquirindo durante a apresentação da aula ele necessita de motivações e recursos que o notifiquem para a importância de entender seus conhecimentos.

**Solução:** Os recursos *on-line* disponibilizados ao aluno para que ele possa explorar o conhecimento em questão devem ser os mais variados, provir de várias fontes e, principalmente, utilizarem notações que atraiam o aluno e o motivem a estender o seu conhecimento.

**Contexto Resultante:** A utilização do padrão Explorar Conhecimento faz com que o aluno, durante a aula virtual, em meio a conceitos que está aprendendo ou que terminou de aprender, procure utilizar recursos disponíveis na própria aula para enriquecer o seu aprendizado.

### ***4. DELIMITAR AULA***

**Problema:** Como adaptar uma aula ao ritmo de cada aluno?

**Contexto:** Ao iniciar uma aula o aluno poderia desenvolver melhor o seu aprendizado sobre o conteúdo se pudesse escolher a forma como aprender, ou seja, dentro do mesmo tema escolher conjuntos de tópicos diferentes de apresentação.

**Solução:** Como solução para este problema o professor pode utilizar o recurso de produzir, para o mesmo assunto, variados tipos de aula, os quais considerando os tópicos essenciais a

serem utilizados, abordassem com estratégias e organizações diferentes o conteúdo a ser explorado pelo aluno.

**Contexto Resultante:** Com a utilização do padrão delimitação o professor fornece ao aluno meios para escolher o tipo de aula que este quer realizar. Por exemplo, determinados alunos preferem aulas com tópicos resumidos e notas explicativas enxutas, enquanto outros, necessitam de textos inteiros para efetivarem seu aprendizado.

## ***5. ESTRUTURAR AULA***

**Problema:** Como estruturar uma aula virtual tendo em vista que os alunos estão acostumados com as aulas presenciais?

**Contexto:** Numa aula presencial o aluno não conta com o fator de estar perdido ou confuso com a aquisição de informações, pois conta com a figura do professor para orientá-lo. Numa aula virtual a exposição de conceitos deve ser cuidadosa e o professor deve planejá-la como se estivesse, de fato em uma aula presencial.

**Solução:** Para que os alunos não se percam durante a realização da aula virtual é importante que o professor a estruture de uma forma simples, em pequenas partes e com relacionamento bem definido.

**Contexto Resultante:** Com o padrão Estruturar Aula o professor pode determinar o conteúdo a ser aprendido em toda a aula delimitado-o em partes menores que facilitam o entendimento e a realização do acesso por parte do aluno.

## ***6. DEFINIR PRÉ-REQUISITOS***

**Problema:** Como disponibilizar ao aluno meios para que ele resolva por si próprio possíveis problemas de navegação ou conheça a interface com a qual vai trabalhar durante a realização da aula?

**Contexto:** Todo software, programa ou equivalente necessita de um tutorial, índice ou documento texto que discorra acerca de suas funcionalidades, erros e diretivas para que os usuários possam sanar dúvidas ou problemas decorrentes do uso do ambiente.

**Solução:** Ao criar a aula é importante o professor criar uma página com o título “Pré-requisitos” ou afim, que contenha informações sobre a forma de navegação, considerações necessárias para efetivar a aula e requisitos mínimos para que o aluno possa realizar a aula por completo.

**Contexto Resultante:** Por meio do padrão Definir Pré-requisitos, a disponibilização de uma aula se torna mais completa uma vez que o aluno pode conhecer os requisitos básicos para efetuar a aula com completude.

## **7. RELACIONAR CONTEÚDO**

**Problema:** De que forma, ao iniciar um novo tópico ou aula, o professor pode introduzir o conteúdo a ser apresentado ao aluno?

**Contexto:** Uma vez tendo preparado o material a ser disponibilizado para o aprendizado, o professor deve, de alguma forma, fazer com que o aluno relacione o novo conceito a ser aprendido com conceitos previamente conhecidos. Assim, de posse desses relacionamentos, o aluno pode mais facilmente familiarizar-se com os novos tópicos que irá conhecer.

**Solução:** Uma das possíveis soluções apresentadas por Neto (2005), trata do uso da Estratégia Cognitiva Organizadores de Avanço, os quais são reconhecidos como tópicos ou categorias que apresentam uma organização e posterior aprendizado dos fatos e detalhes planejados pelo professor em seu material.

**Contexto Resultante:** Espera-se que com o uso desse padrão o professor possa gerar um relacionamento entre o novo material aprendido pelo aluno e os conceitos já dominados por este, facilitando desta forma, o processo de ensino-aprendizagem.

## **8. USAR ATIVIDADES**

**Problema:** Como testar os conhecimentos do aluno durante o aprendizado?

**Contexto:** Enquanto estão realizando uma aula, os alunos necessitam de recursos que forcem a sua memória a exercitar conceitos pertinentes ao assunto tratado para que o aprendizado se intensifique.

**Solução:** Para fazer com que o aluno exercite os conhecimentos que vêm adquirindo desde o começo da aula é recomendado que seja realizada a inclusão de atividades logo após cada apresentação de conteúdo. Estas atividades não se resumem em exercícios onde o aluno deve fornecer respostas a fim de ser avaliado, elas podem se estender ao acesso à *links* de estudo, leituras adicionais, experiências práticas, entre outros.

**Contexto Resultante:** Quando o professor utiliza o padrão Usar Atividades no desenvolvimento de uma aula ele está ativando uma forma de sedimentação dos conteúdos apresentados que pode aprimorar cada vez mais o aprendizado por parte do aluno.

## ***9. FECHAR TÓPICO***

**Problema:** Como proceder para que os alunos possam aprender em intervalos periódicos dispostos pelo cérebro, tendo conhecimento de que alcançou o fim de cada tópico?

**Contexto:** Para que o aprendizado se efetive o cérebro necessita de determinado tempo após cada apresentação de informação disposta em tópicos. Nesse tempo ele realiza o processamento necessário para armazenar as novas informações e disponibilizar a entrada de novos dados dispostos em tópicos seguintes.

**Solução:** Para que os conteúdos sejam efetivamente aprendidos pelo aluno é fundamental que exista um fechamento parcial de cada um deles. Isto por meio de intervalos nos quais as informações aprendidas possam ser processadas. Esses intervalos podem apresentar Atividades ou conteúdos adicionais, ou mesmo um indicativo de finalização. Também se torna importante demarcar os tópicos já finalizados pelo aluno por meio de alterações de cores de fonte ou de fundo do link relacionado a este tópico.

**Contexto Resultante:** Com a aplicação do padrão Fechar Tópico, durante a realização de uma aula o aluno pode ter intervalos regulares para processar as informações apresentadas para, de fato, aprendê-las, uma vez que sabe quando chegou ao fim da unidade ou módulo.

## ***10. EDITAR APRESENTAÇÃO***

**Problema:** Como representar as informações fazendo uso de recursos multimídia?

**Contexto:** Em cada página de apresentação do conteúdo é interessante utilizar recursos multimídia que possam expressar com maior fidelidade a informação a ser aprendida.

**Solução:** Sempre que possível utilizar imagens, sons, vídeos, tabelas, gráficos, entre outros recursos multimídia, enriquece a aula e faz com que o conteúdo se torne mais fácil de ser explorado. Evitar seções contínuas de texto utilizando uma formatação com cores, marcadores e espaços também constitui uma boa solução para apresentar o conteúdo da aula. Além disso, é importante que o professor estabeleça um plano de fundo e layout para que a apresentação do conteúdo seja realizada uniformemente durante toda a aula.

**Contexto Resultante:** Recursos diferentes de textos simples, como os utilizados em aulas presenciais, são eficientes na aquisição de conteúdos por facilitar a exploração e entendimento destes.

## ***11. ESQUEMATIZAR CONTEÚDO***

**Problema:** Em que proporção a disponibilização do conteúdo deve ocorrer durante a exploração da aula pelo aluno?

**Contexto:** Em determinada aula é interessante para o processo de aprendizado que o aluno tenha uma disponibilização de conteúdo total, ou seja, livre, podendo realizar acesso a todos os tópicos do assunto sem uma ordem predefinida. Porém, existem aulas em que o professor pode restringir o acesso por parte do aluno a determinados tópicos, deixando-o condicionado a finalizar certos tópicos ou atividades consideradas essenciais para o aprendizado posterior.

**Solução:** Quando o professor definir uma aula de livre acesso aos tópicos tratados ele pode se utilizar um índice amplamente navegável, enquanto que, quando a aula requerer disponibilização gradativa de tópicos, pode-se mostrar o conjunto de todos os tópicos existentes na aula, porém fica predeterminada a ativação de um segundo módulo somente quando um primeiro for completado.

**Contexto Resultante:** De posse das opções de poder restringir ou não o acesso aos tópicos da aula o professor pode conduzir o aprendizado do aluno condicionando-o a adquirir conhecimentos básicos que serão necessários para aprendizados futuros.

## ***12. CONFIGURAR NAVEGAÇÃO***

**Problema:** Disponibilização da exploração do conteúdo de maneira compreensível ao aluno.

**Contexto:** Para que o aluno possa explorar a aula navegando pelas páginas existentes de forma prática, o professor deve disponibilizar meios e recursos que garantam a fluidez e continuidade do aprendizado durante a aula.

**Solução:** A utilização de recursos como texto sublinhado, botões, imagens, organizadores, entre outros, pode ser eficaz para ajudar o aluno em sua localização na aula.

**Contexto Resultante:** Quando da utilização do padrão de configuração da navegação, o professor passa a apresentar o conteúdo ao aluno para que este mesmo possa realizar a aula conforme seu ritmo e motivação de forma compreensível a si mesmo.

## ***13. REALIZAR REVISÃO***

**Problema:** Há propensão do aluno esquecer-se do que já foi aprendido.

**Contexto:** Em determinados temas, principalmente em se tratando dos mais longos, a divisão em tópicos com intervalos posteriores pode fazer com que o aluno se esqueça ou perca alguns detalhes do conteúdo inicial apresentado.

**Solução:** Para que isso não prejudique o aprendizado é necessário que o professor estruture revisões após cada intervalo utilizando, se necessário, *links* para os tópicos apresentados para que o aluno possa retornar a eles caso tenha dúvidas.

**Contexto Resultante:** Com a inclusão de revisões após o intervalo entre tópicos o aluno tem a oportunidade de relembrar conceitos já aprendidos, ou mesmo retornar ao tópico para sedimentar conceitos não totalmente aprendidos.

#### ***14. FINALIZAR AULA***

**Problema:** Como o aluno saberá quando finalizou cada parte do aprendizado e chegou ao fim da aula?

**Contexto:** Ao realizar a navegação pelos tópicos que constituem a aula o é importante para o aluno saber quando chegou ao fim de determinada exploração de conhecimento.

**Solução:** A indicação de visitação de tópicos por meio da alteração de cores nos links onde o estudo do tópico já foi realizado é uma das estratégias que podem ser utilizadas juntamente com a presença de uma última página que indique a finalização da aula.

**Contexto Resultante:** Quando o aluno finaliza uma aula de determinado tema a ser estudado é como se ele tivesse alcançado o fim de um trajeto parcial. Isso também é importante para delimitar aprendizado e o início de parte do processo de avaliação.