

Trabalho de Conclusão de Curso

Padrões Pedagógicos e Aprendizagem Significativa no contexto do framework FA_PorT

Heberth Braga Gonçalves Ribeiro

braga.heberth@gmail.com

Orientador:

Arturo Hernández-Domínguez

Heberth Braga Gonçalves Ribeiro

Padrões Pedagógicos e Aprendizagem Significativa no contexto do framework FA_PorT

Monografia apresentada como requisito parcial para
obtenção do grau de Bacharel em Ciência da Com-
putação do Instituto de Computação da Universidade
Federal de Alagoas.

Orientador:

Arturo Hernández-Domínguez

Maceió, Fevereiro de 2008

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação do Instituto de Computação da Universidade Federal de Alagoas, aprovada pela comissão examinadora que abaixo assina.

Arturo Hernández-Domínguez - Orientador
Instituto de Computação, Campus A. C. Simões
Universidade Federal de Alagoas

Fábio Paraguaçu - Examinador
Instituto de Computação
Universidade Federal de Alagoas

Sílvia Chagas da Silva - Examinador
Instituto de Computação
Universidade Federal de Alagoas

Resumo

O objetivo primordial desta monografia é utilizar padrões pedagógicos com ênfase no processo da aprendizagem significativa no contexto de um sistema portfólio-tutor desenvolvido a partir do framework Fa_PorT (framework para sistemas portfólio-tutor baseado em agentes).

Sistemas portfólio-tutor consistem de um sistema tutor acoplado a um portfólio-eletrônico no contexto do ensino a distância (EAD). Estes sistemas detêm a capacidade de armazenar informações a respeito da aprendizagem do aluno. Um sistema tutor inteligente (STI) é capaz de fornecer um ensino adaptado de acordo com o nível de aprendizagem do aluno.

Um sistema portfólio-tutor construído a partir do FA_PorT oferece meios para o auxílio no processo da aprendizagem significativa, permitindo ao professor criar sessões de ensino a partir de estratégias didáticas, motivando assim, o aluno a explorar e investigar novos conhecimentos.

As estratégias didáticas devem permitir que o conteúdo assimilado pelos alunos seja potencialmente significativo na formação de sua estrutura cognitiva. Sob esta perspectiva, fez-se uma análise de padrões pedagógicos como estratégias de ensino-aprendizagem.

A versão atual do FA_PorT permite a criação de sessões de ensino, onde padrões pedagógicos podem ser utilizados no contexto de uma aula semi-presencial. Os padrões pedagógicos descrevem problemas e soluções relacionados a atividades de ensino e aprendizagem, particularmente caracterizadas pelo ensino do paradigma e desenvolvimento orientado a objetos.

Por meio da especificação dos padrões pedagógicos no contexto da aprendizagem significativa, percebeu-se a necessidade do FA_PorT evoluir. Isto possibilitou a criação de uma nova tática de ensino e inclusão de um novo componente à estrutura do Framework, via reuso de software.

Palavras-chave: Framework, Educação a Distância, Padrões Pedagógicos, Aprendizagem Significativa, Portfólio Eletrônico e Sistema Tutor Inteligente.

Abstract

The primary goal of this monograph is to use pedagogical patterns with emphasis in the process of significant learning in the context of a portfolio-tutoring system developed from the FA_PorT framework (framework for portfolio-tutoring systems based on agents).

Portfolio-tutoring systems consist of a coupled tutoring system to an electronic-portfolio in the context of distance education. These systems have the capacity to store informations about the student learning. An intelligent tutoring system (ITS) is able to provide an adjusted teaching according to the level of the student learning.

A portfolio-tutoring system constructed from the FA_PorT provides the means to the aid in the process of significant learning, allowing the teacher to create on-line teaching sessions from the pedagogics strategies, encouraging, the student to explore and investigating new knowledges.

The pedagogics strategies should allow that content assimilated by the students would be it potentially significant in the formation of their cognitive structure. Under this perspective, it has been made pedagogical patterns analysis as strategies of teaching and learning.

The current version of FA_PorT allows the creation of teaching sessions, where pedagogical patterns can be used in the context of a semi-presential lesson. The pedagogical patterns describes related teaching activities and learning problems and solutions, particularly characterized by the objects orientated development and teaching paradigm.

From the pedagogic patterns specification in the context of significant learning, it was noted the need of FA_PorT to evolve. This enabled the development of a new teaching tactic and the add of a new component to the structure of the Framework by software reuse.

Keywords: Framework, Distance Education, Pedagogical Patterns, Significant Learning, Electronic Portfolio and Intelligent Tutoring System.

Agradecimentos

Primeiramente, agradeço a minha família, minha mãe Ítala e meu Pai Hélio (*in memoriam*), meus irmãos Helynne, Hélio e Helyzabeth, pelo incentivo em todos os momentos de minha vida e por sempre terem acreditado no meu potencial para alcançar este momento.

Agradeço a mim mesmo por sempre ir atrás dos meus objetivos, independente se fracassarei ou terei êxito.

Ao meu orientador, professor doutor Arturo Hernández-Domínguez pelos ensinamentos em sala de aula, esclarecimento de dúvidas e o acompanhamento passo-a-passo para que este trabalho de conclusão de curso pudesse ter sido realizado.

Aos amigos de longa data Marcinkus, Thiago Papel, Wagner, Rômulo, Thiago Chada e Daniel, pelo incentivo, paciência nos momentos em que tive que me ausentar para estudar e pôr em prática os longos trabalhos do curso, e mais do que tudo, a amizade.

Ao meus colegas da UFAL, em especial aos amigos Mário, Diogo, Ícaro e Gustavo pelas discussões sobre assuntos relevantes à graduação e extra graduação, as noites perdidas elaborando projetos, e a amizade que surgiu durante o curso.

Ao colega e mestrando Alex Melo, pela ajuda na monografia.

Ao Pink Floyd e suas músicas inspiradoras, que ajudaram nos momentos que exigiam concentração e calma.

Aos filmes do David Lynch, pelas idéias abstratas e a criatividade em lidar com os pensamentos conscientes e inconscientes.

Às minhas chefes Renata e Wilma, na época de estágio na Empresa Brasileira de Correios e Telégrafos, pelos ensinamentos sobre a vida profissional, desenvolvimento de sistemas e banco de dados.

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Contexto	1
1.2.1	Framework FA_PorT	2
1.2.2	Sistemas Portfólio-Tutor	2
1.3	Objetivos	2
1.4	Estruturação	2
2	Fundamentação Teórica	3
2.1	Reuso de Software	3
2.1.1	Componentes de Software	3
2.1.2	Framework	4
2.1.3	Padrões de Projeto	6
2.2	Aprendizagem Significativa	8
2.2.1	Predisposição para aprendizagem significativa	8
2.2.2	Material de aprendizagem potencialmente significativo	8
2.2.3	Elementos da aprendizagem significativa	9
2.3	Padrões Pedagógicos	10
2.3.1	Formato de um Padrão Pedagógico	11
2.3.2	Exemplo de Padrões Pedagógicos	11
2.3.3	Utilização dos Padrões Pedagógicos	16
3	Framework FA_PorT	17
3.1	Arquitetura do FA_PorT	17
3.2	Componentes do FA_PorT	18
3.3	Especificação de um novo Componente e sua Interface na Camada Serviços	18
3.3.1	Componente Fórum	18
3.4	Especificação de uma sessão de ensino on-line no FA_PorT	21
3.5	Funcionamento de um novo Sistema Portfólio-Tutor	24
3.5.1	Camadas de uma nova aplicação	24
3.6	Construção de Aplicações Portfólio-Tutor a partir do FA_PorT	25
4	Padrões Pedagógicos no contexto de uma sessão de ensino no FA_PorT	28
4.1	Ferramentas Utilizadas	28
4.1.1	Java	28

4.1.2	Webwork	29
4.1.3	MySQL	29
4.1.4	Eclipse	30
4.2	Especificação do padrão Leitura, Crítica, Exposição Teórica, Atividade e Apresentação com discussão (RCLAP)	30
4.2.1	Conceitos e Recursos didáticos	30
4.2.2	Definição da Estratégia de Ensino	31
4.3	Especificação do padrão Exposição Teórica, Exemplos, Atividades, Apresentação e Avaliação (LEASPE)	32
4.3.1	Conceitos e Recursos didáticos	33
4.3.2	Definição da Estratégia de Ensino	33
4.4	Utilização dos Padrões Pedagógicos RCLAP e LEASPE no contexto de sessões de ensino no FA_PorT	35
4.4.1	Sessão de ensino baseada no Padrão Pedagógico RCLAP	35
4.4.2	Sessão de ensino baseada no Padrão Pedagógico LEASPE	39
5	Considerações Finais	44
5.1	Resultados Obtidos	44
5.2	Sugestões Para Trabalhos Futuros	45

Lista de Figuras

2.1	Diagrama de um componente e sua interface com as operações definidas. . . .	4
2.2	Modelo de colaborações em uma biblioteca de classes e em um framework. . .	5
2.3	Esquema do “Princípio Hollywood”.	5
2.4	Processo de assimilação.	10
3.1	Arquitetura do framework FA_PorT para sistemas portfólio-tutor.	18
3.2	Diagrama de Componentes do Framework FA_PorT	19
3.3	Diagrama de classes do Componente Fórum.	22
3.4	Representação de uma sessão de ensino on-line no FA_PorT.	22
3.5	Camadas de um aplicação construída a partir do framework FA_PorT.	24
4.1	Tela inicial da aplicação portfólio-tutor.	35
4.2	Tela da estratégia didática baseada no padrão pedagógico RCLAP.	36
4.3	Tela da criação da sessão de ensino baseada na estratégia didática RCLAP. . .	37
4.4	Tela da tática de ensino “Reuso” para Evolução da UML.	37
4.5	Tela exibindo o recurso didático em anexo.	38
4.6	Tela da tática de ensino “Exercício” para o exercício de verdadeiro ou falso. . .	38
4.7	Tela da tática de ensino “Geração de relatório”.	39
4.8	Tela da tática de ensino debate síncrono.	39
4.9	Tela da estratégia didática baseada no padrão pedagógico LEASPE.	40
4.10	Tela da tática de ensino “Reuso” para Introdução sobre Framework.	41
4.11	Tela exibindo o recurso didático em anexo.	42
4.12	Tela exibindo os tópicos disponíveis durante a tática de Fórum numa sessão de ensino.	42
4.13	Tela exibindo as mensagens enviadas.	43
4.14	Tela exibindo as mensagens enviadas e um anexo.	43

Lista de Tabelas

4.1	Táticas de ensino da estratégia que representa o padrão pedagógico RCLAP . .	36
4.2	Táticas de ensino da estratégia que representa o padrão pedagógico LEASPE .	40

Lista de Códigos

3.1	Interface do Componente Fórum	19
3.2	Nova aplicação gerada a partir do Framework	25
3.3	Nova aplicação gerada e customizada a partir do Framework	25

Capítulo 1

Introdução

1.1 Motivação

Com o surgimento de novas tecnologias computacionais, em especial a Internet, possibilitou-se o crescimento de meios para o auxílio no processo da aprendizagem significativa, permitindo ao aluno construir seu conhecimento através de ações ativas e investigativas (Nunes & Santos, 2006).

O professor como indutor da aprendizagem significativa, deve ir além da simples transmissão de conteúdos. Ele deve ser capaz de traçar estratégias e situações que motivem o aluno na perspectiva de aquisição de conhecimentos relevantes a sua estrutura cognitiva (Nunes & Santos, 2006).

Objetivando auxiliar os professores na procura de estratégias para obter um melhor aprendizado dos alunos, Silva (2007) cita duas possibilidades:

1. A análise de Padrões Pedagógicos (Lilly, 1996), para servir como uma estratégia de ensino e aprendizagem a partir de uma proposta de aprendizagem significativa.
2. Analisar sistemas existentes na Web com uma abordagem que possibilite a utilização de padrões pedagógicos com ênfase numa aprendizagem significativa e que permita construir interações a partir de conhecimentos pré-existent na estrutura cognitiva do aluno.

1.2 Contexto

Neste trabalho de conclusão de curso, apresenta-se a evolução do framework FA_PorT (Medeiros 2007) a partir do uso de padrões pedagógicos (Lilly, 1996) em sessões de ensino no contexto de um curso semi-presencial com ênfase numa aprendizagem significativa.

1.2.1 Framework FA_PorT

O FA_PorT, framework para sistemas Portfólio-tutor baseado em agentes (Medeiros, 2006) fornece a estrutura necessária para a criação de sistemas Portfólio-tutor.

1.2.2 Sistemas Portfólio-Tutor

Segundo Nascimento et al. (2002), um sistema portfólio-tutor consiste de um sistema tutor acoplado a um portfólio-eletrônico. O portfólio-tutor, além de permitir, através da criação de sessões de ensino, a exibição de conteúdos e exercícios para os alunos, permite também o acompanhamento e o registro do que foi aprendido pelo aluno.

O Portfólio-tutor é uma ferramenta que dá suporte a sessões de ensino on-line (Medeiros, 2007). As sessões de ensino são representadas através de estratégias didáticas que são definidas através de um conjunto de táticas de ensino (Medeiros, 2007).

1.3 Objetivos

O objetivo deste trabalho é utilizar padrões pedagógicos com ênfase numa aprendizagem significativa. Para isto, alguns objetivos específicos devem ser alcançados:

- Identificar características que permitem uma aprendizagem significativa;
- Pesquisar núcleo de padrões pedagógicos, analisar e identificar no contexto da aprendizagem significativa os elementos necessários para padrões pedagógicos;
- Analisar o framework FA_Port que será utilizado na implementação de aplicações baseadas em padrões pedagógicos.

1.4 Estruturação

Este trabalho está organizado da seguinte maneira:

No **Capítulo 2 (Fundamentação Teórica)** apresentam-se os conceitos sobre reuso de software, componentes de software, framework, aprendizagem significativa e padrões pedagógicos.

No **Capítulo 3 (Framework FA_PorT)** apresenta-se a estrutura do FA_PorT, o detalhamento de sua arquitetura, a inclusão de um novo componente e, por fim, a construção de aplicações a partir do FA_PorT.

No **Capítulo 4 (Padrões Pedagógicos no contexto de uma sessão de ensino no FA_PorT)** apresenta-se a utilização de dois padrões pedagógicos no contexto de uma sessão de ensino no FA_PorT.

No **Capítulo 5 (Considerações Finais)** apresenta-se as considerações finais sobre o trabalho.

Capítulo 2

Fundamentação Teórica

Este capítulo tem como objetivo descrever alguns conceitos utilizados durante a elaboração deste trabalho. Na seção 2.1 será abordado o reuso de software (Componentes, Framework e Padrões de Projeto). Na seção 2.2 serão abordados conceitos no contexto da aprendizagem significativa, por fim, na seção 2.3, conceitos sobre padrões pedagógicos.

2.1 Reuso de Software

O reuso de software, segundo Sommerville (2005), tem como objetivo desenvolver sistemas de software utilizando componentes já existentes, não somente na fase de codificação, mas nas diversas etapas do processo de desenvolvimento (especificação, modelagem, etc.).

Para Sommerville (2005) as vantagens do reuso de software são os custos mais baixos, o desenvolvimento mais rápido de software e diminuição de riscos.

Sommerville (2005) cita algumas técnicas de reuso de software, tais como: Componentes de Software, Frameworks, Padrões de Projeto, etc.

2.1.1 Componentes de Software

Szyperski (1998) define um componente de software como uma unidade de composição com interfaces contratualmente especificadas com dependências explícitas de contexto, e que pode ser instalada independentemente e estar sujeito à composição por outros componentes.

De acordo com Sommerville (2005), os componentes são mais abstratos que as classes de objetos e podem ser considerados provedores de serviços. Assim, quando um sistema precisa de um serviço específico, ele chama um componente para fornecer esse serviço, sem precisar ter algum conhecimento sobre o funcionamento dele. A Figura 2.1 representa um componente e suas interfaces.

No contexto de reusabilidade de um componente de software, Katz (1993) define que um componente pode ser usado em mais de um sistema de software, ou na construção de outros

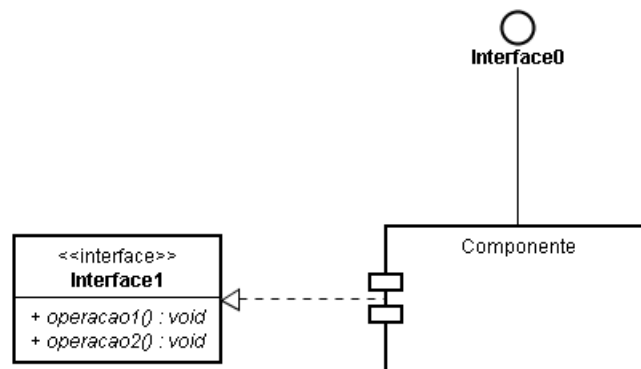


Figura 2.1: Diagrama de um componente e sua interface com as operações definidas.

componentes, com pouca ou nenhuma adaptação. Definindo assim, o nível que um componente possui para ser reutilizado.

2.1.2 Framework

Os primeiros entusiastas do desenvolvimento orientado a objetos sugeriam que objetos eram a abstração mais apropriada para reuso. Porém, os objetos são muitas vezes pequenos e especializados demais para uma determinada aplicação. Em vez disso, tornou-se claro que o reuso orientado a objetos é mais bem apoiado por um processo de desenvolvimento orientado a objetos por meio de abstrações maiores chamadas *frameworks* (Sommerville, 2005).

De acordo com Fayad e Schmidt (1997), frameworks de aplicações orientada a objetos são tecnologias que visam a diminuição do custo e qualidade do software durante a fase de projeto e implementação.

Definição

Fayad e Schmidt (1997) definem *framework* como um projeto reutilizável de subsistema constituído de um conjunto de classes abstratas e concretas. As aplicações, normalmente, são construídas a partir dos frameworks.

Segundo Fayad e Schmidt (1997), um framework funciona como um esqueleto, permitindo que aplicações sejam construídas podendo ser customizadas de acordo com a necessidade dos desenvolvedores de aplicações. Isto é possível, devido ao fato do framework possuir métodos “hook” (ganchos), que permitem as aplicações estenderem sua interface estável.

Ainda, segundo Fayad e Schmidt (1997), o framework objetiva o reuso de sua estrutura interna, isto é, o núcleo do esqueleto de uma aplicação, e a diminuição do tempo de desenvolvimento de uma nova aplicação.

Características

A arquitetura de um framework é caracterizada pela inversão de controle, onde aplicações são customizadas com os métodos definidos pelo usuário e o próprio framework será o responsável por chamar estes métodos (Fayad e Schmidt, 1997).

Diferenças entre um Framework e biblioteca de classes orientada a objetos

Segundo Sauv  (2000), cada classe   nica, independente das outras e as aplica  es criam as colabora  es. Em um framework, as depend  ncias (colabora  es) est  o embutidas. A Figura 2.2 mostra as colabora  es em uma biblioteca e em um framework.

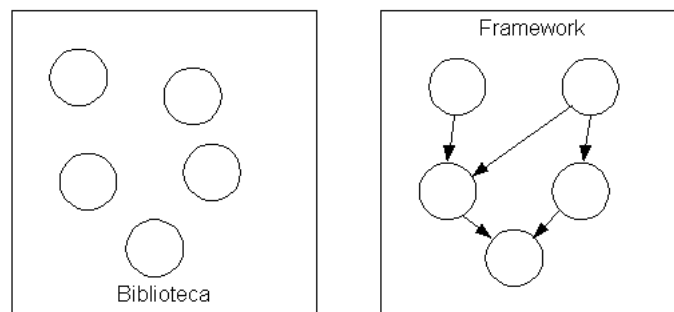


Figura 2.2: Modelo de colabora  es em uma biblioteca de classes e em um framework.

De acordo com Sauv  (2000), o framework    utilizado de acordo com o ‘‘Princ pio Hollywood’’(N o nos chame, n s   que lhe chamaremos). O framework    quem ir  chamar o c digo de aplica  o. A Figura 2.3 mostra o esquema do ‘‘Princ pio Hollywood’’.

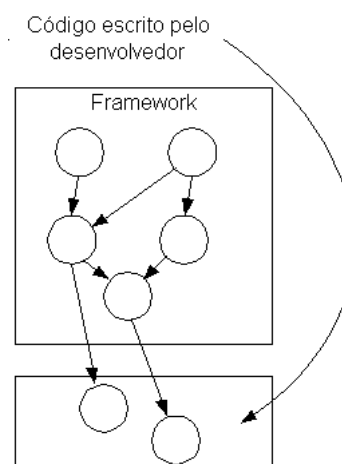


Figura 2.3: Esquema do ‘‘Princ pio Hollywood’’.

Classificação

Segundo Fayad & Schmidt (1997) e Sommerville (2005), podemos classificar um framework em três classes:

1. **Frameworks de infra-estrutura de sistema:** São compatíveis com o desenvolvimento das infra-estruturas de sistemas (comunicações, interfaces com o usuário, compiladores, etc.).
2. **Frameworks de integração com middleware:** Comumente utilizados na integração de aplicações distribuídas e componentes. São designados para permitir o reuso, modularização e estender a infra-estrutura do software em ambientes distribuídos.
3. **Frameworks de aplicações corporativos:** Incluem o conhecimento do domínio de aplicações, sendo compatíveis com o desenvolvimento de aplicações para os usuários finais.

2.1.3 Padrões de Projeto

Definição

Um padrão de projeto nomeia, abstrai e identifica os aspectos essenciais de uma estrutura de projeto comum, de modo que esta possa ser utilizada na criação de um projeto orientado a objetos reutilizável. O padrão de projeto identifica as classes participantes, seus papéis, colaborações e a distribuição de responsabilidades. Cada padrão de projeto está relacionado a um determinado tipo de problema de projeto orientado a objetos. Ele descreve quando é possível aplicá-lo, se é possível ou não aplicá-lo em função de outras restrições de projeto e as consequências, custos e benefícios de sua utilização (Gamma & Helm & Johnson & Vlissides, 2005).

Estrutura de um Padrão de Projeto

Conforme Gamma et al. (2005), um padrão pode ser descrito através de quatro elementos:

1. **Nome:** É uma referência que pode-se usar para descrever um problema de projeto, suas soluções e consequências em uma única expressão. Dar nome a um padrão aumenta o vocabulário do projeto, permitindo um nível mais alto de abstração, ajudando no desenvolvimento sobre padrões.
2. **Problema:** Descreve quando aplicar o padrão. Mostra também o contexto em que o padrão pode ser aplicado, podendo descrever problemas de projetos específicos. Algumas vezes o problema incluirá uma lista de condições que deve ser satisfeita para que o padrão possa ser aplicado.

3. **Solução:** Descreve como resolver o problema ou como equilibrar todas as restrições e pré-condições especificadas com clareza. A solução não descreve um projeto concreto específico ou uma implementação, e sim, descreve uma estrutura que define os elementos participantes (objetos, classes, etc.).
4. **Consequências:** Descrevem os resultados e análises dos custos da aplicação do padrão. Podem impactar também sobre a flexibilidade e a extensibilidade no uso do padrão.

Classificação

Segundo Gamma et al. (2005), pode-se classificar os padrões de projeto de acordo com um critério, chamado *finalidade*. Este critério reflete o que o padrão faz. Os padrões podem ter finalidade de criação, estrutural ou comportamental.

- **Padrões de Criação:** Os padrões de criação abstraem o processo de instanciação. Eles ajudam um sistema independente de como seus objetos são criados, compostos e representados. Um padrão de criação de classe usa a herança para variar a classe que é instanciada, enquanto que um padrão de criação de objeto delegará a instanciação para outro objeto. Os padrões de criação encapsulam conhecimento sobre suas classes concretas e ocultam como as instâncias destas classes são criadas e compostas. Exemplos de padrões de criação: Abstract Factory, Factory Method, Singleton, Prototype e Builder.
- **Padrões Estruturais:** Os padrões estruturais se preocupam com a forma como classes e objetos são compostos para formar estruturas mais complexas. Os padrões estruturais de classes utilizam herança para compor (composição estática) interfaces ou implementações. Os padrões estruturais de objetos descrevem maneiras de compor (composição dinâmica) objetos para obter novas funcionalidades, tornando possível a composição em tempo de execução. Exemplos de padrões estruturais: Adapter, Bridge, Decorator, Façade, Proxy, etc.
- **Padrões Comportamentais:** Os padrões comportamentais, não descrevem apenas padrões de objetos e classes, mas também os padrões de comunicação entre eles. Estes padrões se preocupam com a atribuição de responsabilidades entre objetos e utilização de algoritmos, além de caracterizar fluxos de controle difíceis de seguir em tempo de execução. Eles se preocupam apenas como os objetos são interconectados. Exemplos de padrões comportamentais: Chain of Responsibility, Command, Interpreter, Template Method, Strategy, Visitor, Observer, etc.

2.2 Aprendizagem Significativa

De acordo com Farias (1982), a teoria de Ausubel prioriza a aprendizagem cognitiva, propondo uma explicação teórica do processo de aprendizagem segundo o ponto de vista do cognitivismo. Para ele, a aprendizagem é a integração do conteúdo aprendido numa edificação mental ordenada, chamada de estrutura cognitiva.

A idéia central da teoria é a de que o fator isolado que mais influencia a aprendizagem é aquilo que o aprendiz já sabe e, conseqüentemente, isso deveria ser determinado para servir de base ao ensino (Farias, 1982). Ainda, segundo Farias (1982), a partir da abordagem cognitiva, Ausubel tomou como conceito central de sua teoria o que ele chama de *aprendizagem significativa*.

Para Ausubel & Novak & Hanesian (1980), a aprendizagem significativa ocorre quando a tarefa de aprendizagem implica relacionar, de forma não arbitrária e substantiva, uma nova informação a outras com as quais o aluno já esteja familiarizado. Segundo Farias (1982), para Ausubel o processo de aprendizagem significativa envolve a interação da nova informação com uma estrutura de conhecimento específica, já existente na estrutura cognitiva do aluno.

Nunes & Santos (2006) dizem que são necessárias duas condições para que se ocorra a aprendizagem significativa, a predisposição para a aprendizagem significativa e o material da aprendizagem significativa ser potencialmente significativo.

2.2.1 Predisposição para aprendizagem significativa

Para Ausubel et al. (1980), as idéias expressas simbolicamente são relacionadas às informações previamente adquiridas pelo aluno através de uma relação não arbitrária e substantiva. Isto significa que, as idéias são relacionadas a algum aspecto relevante na estrutura cognitiva do aluno.

De acordo com Ausubel et al. (1980), a aprendizagem significativa pressupõe que o aluno esteja disposto para a aprendizagem significativa, ou seja, uma disposição para relacionar, de forma não arbitrária e substantiva, novos conceitos à sua estrutura cognitiva.

2.2.2 Material de aprendizagem potencialmente significativo

A aprendizagem significativa não deve ser interpretada simplesmente como a aprendizagem de material significativo. Estes materiais são apenas potencialmente significativos, incorporados a estrutura cognitiva do aluno através de uma relação não arbitrária e substantiva (não literal) (Ausubel et al., 1980).

A relação não arbitrária implica que, se o material propriamente dito exibe um caráter suficientemente não arbitrário (não aleatório), é porque existe uma base adequada para relacioná-lo de forma não arbitrária aos tipos de idéias correspondentemente relevantes que os seres humanos são capazes de aprender (Ausubel et al., 1980).

A relação substantiva permite que um símbolo ou grupo de símbolos equivalentes se relacionem à estrutura cognitiva sem qualquer alteração resultante no significado (Ausubel et al., 1980).

2.2.3 Elementos da aprendizagem significativa

Subsunçores

De acordo com Farias (1982), na Teoria de Ausubel, a aprendizagem significativa ocorre quando a nova informação parte de conceitos ou proposições relevantes (subsunçores) pré-existentes na estrutura cognitiva do aluno. O subsunçor é uma estrutura específica ao qual uma nova informação pode interagir ao cérebro, que é uma estrutura organizada seguindo uma hierarquia responsável pelo armazenamento de experiências prévias do aprendiz.

Ausubel acredita que os subsunçores surgem em duas etapas: na formação de conceitos no desenvolvimento cognitivo de uma criança até a fase escolar (pequena quantidade de conceitos formados) e a partir de conceitos inteiramente novos para o aprendiz (transformação da aprendizagem mecânica em significativa).

Organizadores Prévios

Segundo Ausubel, não existindo subsunçores à aprendizagem significativa de um novo conceito, recomenda-se o uso de organizadores prévios (Farias, 1982).

Os organizadores prévios, âncoras a fim de manipular a Estrutura Cognitiva para facilitar a aprendizagem significativa, são materiais instrucionais elaborados para interagir com subsunçores já existentes. Para Ausubel, isto irá suprir a deficiência de subsunçores adequados ao ensino de uma nova informação (Farias, 1982).

A utilização de organizadores prévios pode acontecer com a apresentação de matérias introdutórias antes do conteúdo a ser aprendido, porém em um nível mais alto de abstração (Farias, 1982).

Assimilação

Na teoria da assimilação proposta por Ausubel, o novo conceito irá interagir com subsunçores relevantes existentes da estrutura cognitiva, ocorrendo assim, a assimilação entre o novo conceito recebido e conhecimentos já existentes (Farias, 1982).

A partir desta interação irão surgir subsunçores modificados (aprimorados os já existentes e até novos subsunçores). A Figura 2.4 mostra esquematicamente o processo de assimilação (Farias, 1982).

A assimilação irá ocorrer quando um conceito **a** é assimilado sob uma idéia ou conceito mais inclusivo já existente na estrutura cognitiva, resultando significado para a nova informação e elaboração de um novo conceito subsunçor. Tanto o conceito que passa a **a'** como o conceito

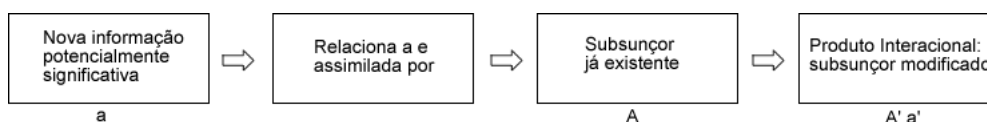


Figura 2.4: Processo de assimilação.

subsunção que passa a A' são modificados pela interação, resultando o produto da aprendizagem significativa $a'A'$ (Farias, 1982).

Diferenciação Progressiva e Reconciliação Integrativa

De acordo com Farias (1982), quando um novo conceito ou proposição é aprendido por subordinação (processo de interação e ancoragem em um conceito subsunção), este também se modifica. A ocorrência deste processo uma ou mais vezes, segundo Ausubel et al. (1980), leva a *diferenciação progressiva* do conceito ou proposição que engloba novas informações.

Em outros tipos de aprendizagem, ocorre muitas vezes que idéias já existentes na estrutura cognitiva podem, no curso de novas aprendizagens, ser reconhecidas como relacionadas, o que antes não acontecia (Ausubel et al., 1980). Assim, segundo Farias (1982), ao mesmo tempo em que novas informações são adquiridas, elementos isolados existentes na estrutura cognitiva podem reorganizar-se e adquirir novos significados. Esta recombinação dos elementos é denominada *reconciliação integrativa* (Ausubel et al., 1980).

A reconciliação integrativa é mais completa quando as possíveis fontes de confusão são eliminadas pelo professor ou recursos didáticos, ajudando assim, o aluno a resolver as inconsistências ou conflitos aparentes entre conceitos ou proposições. Tanto a diferenciação progressiva, quanto a reconciliação integrativa podem ser implementadas através do uso de organizadores prévios adequados (Ausubel et al., 1980).

2.3 Padrões Pedagógicos

Os Padrões Pedagógicos irão tentar capturar conhecimentos avançados na prática de ensino e aprendizagem, particularmente pelo ensino do paradigma da orientação a objetos. (Pedagogical Patterns Project, 2007).

Foi pensando na dificuldade de ensinar o paradigma de orientação a objetos que Susan Lilly deu início ao Projeto Padrões Pedagógicos (Lilly, 1996). Este projeto tem como objetivo catalogar os diversos tipos de padrões em contextos e assuntos bem definidos, cujas descrições seguem um formato estruturado e padronizado (Neto, 2000).

De acordo com Oliveira (2005), em vez de catalogar soluções eficazes para problemas inerentes à elaboração de um projeto de software, os padrões pedagógicos descrevem soluções para problemas relacionados ao processo de ensino-aprendizagem.

A partir dos trabalhos iniciados por Susan Lilly criou-se o Projeto Padrões Pedagógicos registrado em <http://www.pedagogicalpatterns.org/>, e nele encontra-se um vasto catálogo de padrões já registrados e que continua em expansão.

2.3.1 Formato de um Padrão Pedagógico

Logo abaixo é descrita a estrutura de um Padrão Pedagógico (Neto, 2000):

- **Nome:** Apresenta o nome do Padrão Pedagógico (RCLAP, EPIC, etc.).
- **Objetivo:** O que se deseja ensinar, ou que dificuldades minimizar, ou que habilidade se deseja desenvolver, etc.
- **Idéia:** Como o Padrão Pedagógico irá atingir o objetivo.
- **Motivação:** O que leva a crer que a idéia atingirá o objetivo.
- **Aplicabilidade:** Contexto no qual o padrão será aplicado.
- **Contra-indicações:** Situações onde o Padrão Pedagógico gera problemas.
- **Estrutura:** Descrição dos formatos de aula e dos procedimentos a serem adotados e o papel dos tutores e alunos neste procedimento.
- **Conseqüências:** Resultados obtidos após a aplicação do Padrão Pedagógico.
- **Pontos a considerar:** Detalhes a serem providenciados e evitados na implementação do Padrão Pedagógico.
- **Dependências culturais:** Aspectos que tornam o Padrão Pedagógico menos ou mais recomendável para uma determinada cultura.
- **Recursos necessários:** Materiais que precisam ser providenciados para a prática do Padrão Pedagógico.
- **Exemplos de utilização:** Experiências onde o Padrão Pedagógico já foi adotado.
- **Padrões relacionados:** Outros padrões que apresentam alguma relação ao que está sendo adotado.

2.3.2 Exemplo de Padrões Pedagógicos

Esta seção irá mostrar alguns padrões pedagógicos aplicados no ensino de orientação a objetos, porém, podem ser aplicados em outras áreas de ensino.

Padrão Pedagógico *Lecture - Activity - Student Presentation - Discussion* (LASD)

Padrão catalogado por Helen Sharp (The Open University - UK). Abaixo segue a estrutura especificada.

- **Objetivo:** Permitir que os alunos tenham uma compreensão de um determinado conceito ou conjunto de conceitos.
- **Motivação:** Permitir que os alunos aprofundem seus conhecimentos através de apresentações de experiências e perspectivas de outros alunos.
- **Aplicabilidade:** Utilizado no contexto de revisões. Por exemplo, um aluno concluiu uma bateria de exercícios, sendo então, submetido a uma avaliação.
- **Estrutura:**
 1. Lecture (Exposição Teórica): Resumo de um tema, destacando técnicas e conceitos.
 2. Activity (Atividade): Atividades realizadas individualmente ou em grupo para fixação de conceitos, podendo ter a ajuda do tutor (se necessário). É recomendado que as atividades sejam realizadas em grupo.
 3. Student Presentation (Apresentação): Cada aluno (atividade individual) ou um representante do grupo apresenta os resultados das atividades que lhes foram atribuídas.
 4. Discussion (Discussão): Os alunos discutem os resultados, analisando diferenças, expondo dúvidas e pontos que não ficaram bastante claros, com outros alunos.
- **Conseqüências:**
 - Permitir ao tutor a oportunidade de resumir os pontos chaves de um determinado conceito;
 - Permitir aos alunos praticarem o aprendizado de um conceito ou conjunto de conceitos com o apoio de um tutor;
 - Possibilidade dos alunos esclarecerem suas idéias e pontos-de-vista para todos os grupos e ao tutor;
 - Aumento do potencial de aprendizado através da troca de experiências com outros alunos.
- **Pontos a considerar:**
 - O tutor deve permitir que os alunos aprendam uns com os outros.

- Atividades devem conter um nível de aprofundamento suficiente para que o aprendizado seja fixado e não simplesmente rever conceitos já apresentados na aula.
- Colaboração entre os grupos ao finalizar as atividades.
- **Utilização:** Pode ser utilizado no contexto de uma aula de análise orientada a objetos, como também sobre uma determinada linguagem de programação.

Padrão Pedagógico *Reading - Critique - Lecture - Activity - Presentation with discussion* (RCLAP)

Padrão pedagógico catalogado por Mary Gorman e Susan Burk (American Management Systems Training Services). Abaixo segue a estrutura especificada.

- **Objetivo:** Permite aos alunos iniciantes trabalhar com um modelo, criado por outros e mais complexo, objetivando o aprofundamento da aprendizagem.
- **Motivação:** As competências necessárias na construção detalhada, exata e consistente de um projeto orientado a objetos podem ser difíceis de identificar e aprender. Este padrão fornece um meio para avaliar a pró-eficiência atual do aluno (através da leitura e crítica de modelos de projeto), e em seguida, oferece uma maneira de aperfeiçoar o conhecimento através da exploração de novos conceitos mais detalhados.
- **Aplicabilidade:** Utilizado na modelagem orientada a objetos, como por exemplo, no ensino da UML abordando os diagramas (casos de uso, classes, estado, objetos, etc.).
- **Estrutura:**
 1. Reading (Leitura): Cada aluno lê o material oferecido independentemente. O material fornece pormenores sobre um determinado conteúdo, uma explicação detalhada do conteúdo oferecido, referências adicionais e diretrizes de garantia de qualidade.
 2. Critique (Crítica): Possibilita uma análise do material oferecido e de um modelo simples por parte do aluno, onde este terá a possibilidade de acrescentar comentários em relação ao material estudado.
 3. Lecture (Exposição Teórica): O instrutor fornece conteúdo e exemplos adicionais para o grupo de alunos, forçando uma discussão por parte dos alunos aquém da leitura individual e da crítica.
 4. Activity (Atividade): Os alunos, em pequenos grupos (ou individualmente), através do que foi aprendido nas fases anteriores, elaboram um estudo de caso explorando os conceitos aprendidos. O tutor servirá apenas como um orientador para os grupos.

5. *Presentation with discussion* (Discussão): Cada grupo apresenta seus resultados. Os outros grupos utilizam as orientações, a fim de verificar os resultados. A comparação dos diferentes resultados fornece conhecimentos adicionais para o aluno.

● **Conseqüências:**

- A leitura permite que o aluno dite o seu próprio ritmo de aprendizagem;
- Na fase de leitura, materiais complementares podem ser fornecidos para o aprofundamento de pesquisa por parte dos alunos;
- A fase da crítica obriga que os alunos tenham feito a leitura do material disponibilizado, para que este possa identificar aplicação de técnicas inadequadas e incorretas. Eles tornam-se mais familiarizados com os materiais freqüentemente referenciados;
- A crítica permite que tanto o aluno, quanto o instrutor, possam avaliar o que foi aprendido. Isto possibilita a identificação de áreas que necessitem de um apoio adicional.

● **Pontos a considerar:**

- Os materiais oferecidos devem permitir uma leitura clara, ser descritivos e bem organizados em componentes, sem perder a coesão dos conceitos;
- As fases de leitura e crítica devem ser feitas independentemente, antes do curso (mas não muito distante, para que os conceitos não sejam esquecidos);
- Baseado nos resultados obtidos durante a fase de crítica, o tutor pode ter que ajustar a quantidade de tempo da discussão para garantir que o grupo compreenda as falhas nos modelos apresentados e os reparos necessários;
- A descrição da atividade deve ser bem definida para exercer a maior parte da modelagem dos conceitos que serão apresentados na discussão.

- **Utilização:** Pode ser utilizado no contexto de uma aula de análise orientada a objetos. Pode ser aplicado, também, em outras áreas de ensino.

Padrão Pedagógico *Lecture - Examples - Activity - Student Presentation - Evaluation* (LE-ASPE)

Padrão catalogado por Martin L. Barrett (Department of Computer and Information Sciences - East Tennessee State University). Abaixo segue a estrutura especificada.

- **Objetivo:** Introduzir técnicas organizacionais na modelagem de classes.

- **Motivação:** Alguns alunos possuem experiência em modelagem de dados, porém, quando uma tarefa de modelagem é apresentada, a maioria não sabe por onde começar. A partir deste problema, algumas instruções introdutórias seguidas por trabalhos em grupo e avaliações podem dar um bom suporte à modelagem.
- **Aplicabilidade:** Utilizado na introdução de modelagem de classes, podendo ser usado quando estudantes vão ganhando experiência alcançando um alto nível de projeto e implementação.
- **Estrutura:**
 1. Lecture (Exposição Teórica): Material relacionado a área de modelagem orientada a objetos é mostrado ao aluno. O material aborda uma visão introdutória, técnicas de utilização e outros conceitos referentes a classes, objetos, etc.
 2. Examples (Exemplos): Exemplos resolvidos são mostrados aos alunos visando a aplicabilidade dos conceitos ensinados. Pode permitir também discussões sobre outras maneiras de se resolver os exemplos.
 3. Activity (Atividade): Atividades referentes aos conceitos e exemplos mostrados aos alunos. As atividades visam o aprofundamento do aluno com os conceitos de ensino, permitindo assim, que exercícios mais complexos sejam atribuídos aos alunos. As atividades podem ser feitas em grupo ou individualmente.
 4. Student Presentation (Apresentação): O aluno ou grupo apresenta as soluções dos exercícios a outros alunos ou ao professor.
 5. Evaluation (Avaliação): A classe, com orientação do professor, pode discutir e avaliar as soluções dos exercícios apresentados, podendo também, apresentar outras formas de soluções para os exercícios.
- **Conseqüências:**
 - Enfatiza a análise como atividade em grupo;
 - Não destaca a implementação como a principal atividade de programação;
 - Dá oportunidade aos estudantes apresentar e defender seu trabalho;
 - Expõe aos estudantes outras soluções e problemas similares.
- **Pontos a considerar:**
 - O instrutor deve encorajar participação total dos alunos nas atividades em grupo;
 - Devem ser desenvolvidos um bom conjunto de problemas a serem resolvidos, podendo incluir problemas do mundo real.

- A ênfase da atividade encontra-se em dados e relações de classe/objeto.
- **Utilização:** Pode ser utilizado no contexto de uma aula com ênfase na introdução do paradigma de orientação a objetos, análise orientada a objetos e projeto orientado a objetos.

2.3.3 Utilização dos Padrões Pedagógicos

No contexto deste trabalho, serão utilizados os Padrões Pedagógicos RCLAP e LEASPE, já apresentados na sub-seção 2.3.2. Os Padrões citados podem ser utilizados em diferentes disciplinas.

Neste trabalho, os padrões escolhidos foram experimentados com o apoio de um sistema portfólio-tutor desenvolvido a partir do framework FA_PorT, no contexto do curso de ciência da computação, utilizando conceitos da disciplina de engenharia de software.

Capítulo 3

Framework FA_PorT

Este capítulo descreverá a especificação do framework FA_PorT (Medeiros, 2006). Na seção 3.1 será descrita a arquitetura do Framework FA_PorT. Na seção 3.2 serão descritos os componentes atuais que fazem parte do FA_PorT. Na seção 3.3, é apresentada a especificação de um novo componente na camada de serviços. Na seção 3.4, é apresentado o funcionamento de uma sessão de ensino on-line. Na seção 3.5, é apresentado o funcionamento de um novo Sistema Portfólio-tutor e, por fim, na seção 3.6, descreve-se a construção de aplicações portfólio-tutor a partir do FA_PorT.

3.1 Arquitetura do FA_PorT

No contexto da especificação do Fa_PorT realizada por Fábio Nicácio (Medeiros, 2006) e Flávio Mota (Medeiros, 2007), a arquitetura divide-se em cinco camadas: Apresentação, Agentes, Tutor, Portfólio e Serviços. Na Figura 3.1 é apresentada a arquitetura do framework FA_PorT e as camadas associadas as aplicações construídas através do mesmo. Segundo Medeiros (2006), os elementos de cada camada de uma nova aplicação portfólio-tutor são representados por componentes estruturados da seguinte maneira:

- Camada Interface: representada pelos componentes `Componente_Cli`;
- Camada Agentes: representada pelos componentes `Componente_CAj`;
- Camada Tutor: representada pelos componentes `Componente_CTk`;
- Camada Portfólio: representada pelos componentes `Componente_CPl`;
- Camada Serviços: representada pelos componentes `Componente_CSm`;

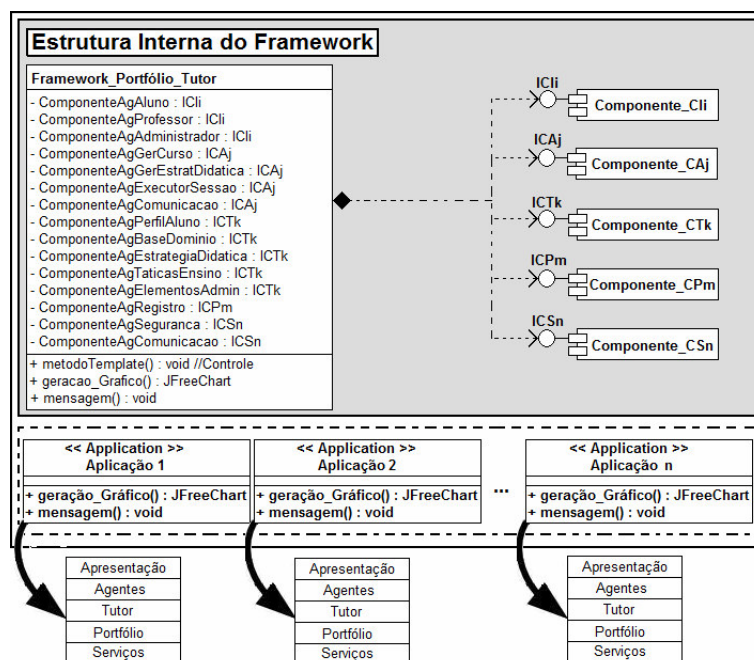


Figura 3.1: Arquitetura do framework FA_PorT para sistemas portfólio-tutor.

3.2 Componentes do FA_PorT

Atualmente o FA_PorT dispõe de onze componentes distribuídos através das camadas citadas na seção 3.1. Segundo Medeiros (2007), na camada de serviços encontram-se os componentes: segurança, conexão com o banco de dados e comunicação. Na camada portfólio encontram-se os componentes registros e elementos administrativos. Na camada tutor, os componentes: perfil, estratégias didáticas e base de domínio. Por fim, na camada dos agentes, os componentes: agente comunicação, agente professor e agente executor de sessões. O diagrama dos componentes citados está representado na Figura 3.2.

3.3 Especificação de um novo Componente e sua Interface na Camada Serviços

Esta seção especifica a criação de um novo componente, o Componente Fórum.

3.3.1 Componente Fórum

Descrição

Esta subseção tem como objetivo descrever o Componente Fórum. Este componente é responsável pela manipulação (inserção, atualização e remoção) de tópicos do fórum, assim como, o controle das mensagens que serão adicionadas a um tópico, com a possibilidade de anexar arquivos no escopo da mensagem. Este componente é utilizado na tática de ensino do tipo fórum

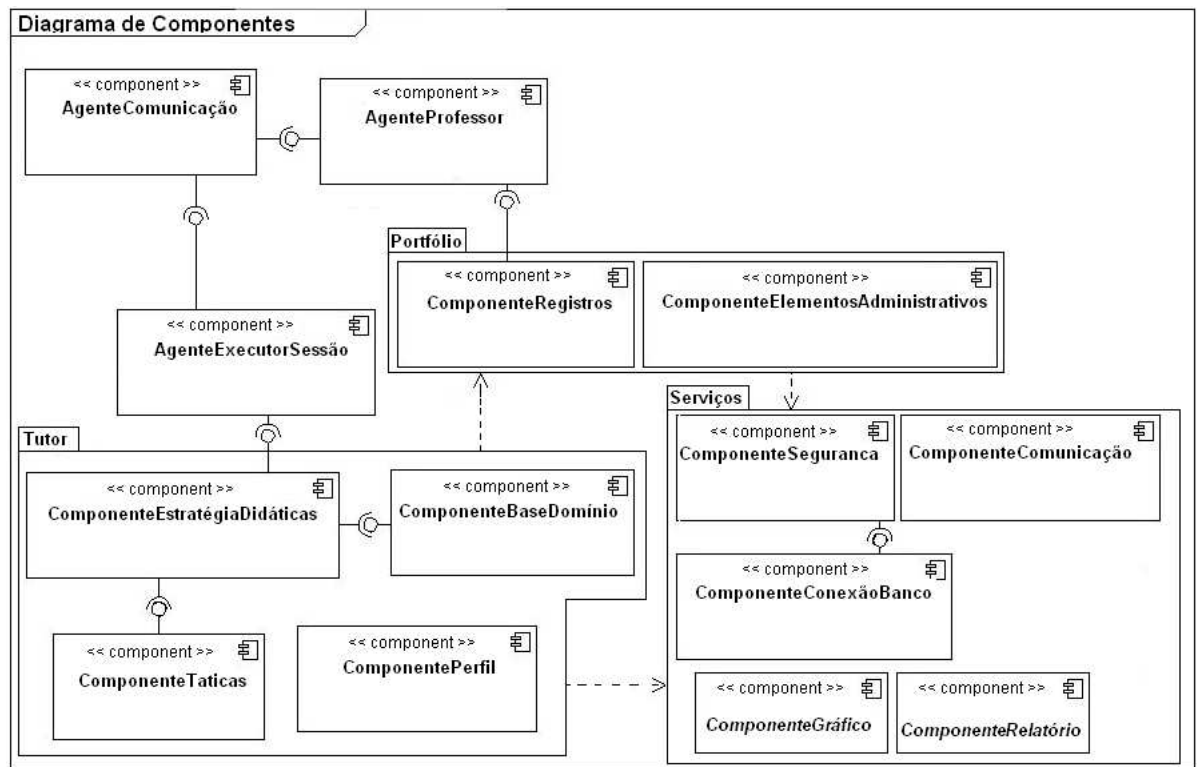


Figura 3.2: Diagrama de Componentes do Framework FA_PorT

e em uma sessão de ensino contendo uma estratégia didática com a tática de fórum associada.

Interface

A interface do Componente Fórum está representada no código 3.1.

Código 3.1: Interface do Componente Fórum

```

1
2 public Interface ForumFacade {
3
4     public int inserirForum(Forum forum);
5
6     public List<Forum> consultarTopicosForum(String nome);
7
8     public Forum consultarTopicoForum(int codigoForum);
9
10    public int atualizarForum(Forum forum);
11
12    public int removerForum();
13
14    public int inserirForumTatica(int codigoForum,

```

```
15
16         int codigoTatica);
17
18     public Forum consultarForumTatica(int codigoTatica);
19
20     public List<Forum> consultarTopicoForumSessao(
21
22         int codigoSessao) throws Exception;
23
24     public int inserirMensagemForum(Forum forum)
25
26         throws Exception;
27
28     public List<Forum> consultarMensagensForum(int codigoForum)
29
30         throws Exception;
31
32     public int removerMensagemForum(int codigoMensagem)
33
34         throws Exception;
35
36     public Forum consultarMensagemForum(int codigoMensagem)
37
38         throws Exception;
39 }
```

Operações

1. Inserir, excluir e atualizar Fórum:

- Descrição: O componente recebe como entrada o fórum e executa as operações de manipulação: inserção, exclusão ou remoção.
- Interação: Operação chamada pela interface do fórum e realizada pelo componente fórum.

2. Consultar tópicos do Fórum:

- Descrição: O componente recebe como entrada a condição da consulta, no caso, o nome do tópico, e então, retorna uma lista de tópicos do fórum.

- Interação: Operação chamada pela interface estratégia didática e realizada pelo componente fórum.
3. Inserir um tópico do fórum em uma tática de ensino do tipo fórum
 - Descrição: O componente recebe como entrada o código do fórum e o código da tática de ensino.
 - Interação: Operação chamada pela interface estratégia didática toda vez que um professor insere uma tática de fórum, selecionando os tópicos que farão parte da estratégia didática. Esta operação é realizada pelo componente fórum.
 4. Inserir uma mensagem com anexo associada a um tópico no fórum
 - Descrição: O componente recebe como entrada o código do tópico, a mensagem e o anexo inseridos pelo usuário.
 - Interação: Operação chamada pela interface sessão de ensino toda vez que um professor ou aluno desejam inserir uma mensagem com anexo referente ao tópico que está sendo exibido durante uma aula na sessão de ensino.
 5. Remover uma mensagem associada a um tópico no fórum
 - Descrição: O componente recebe como entrada o código da mensagem que se deseja remover.
 - Interação: Operação chamada pela interface sessão de ensino toda vez que o professor ou aluno queiram remover uma mensagem.

Diagrama de Classes

O diagrama de classes do Componente Fórum está representado na Figura 3.3

3.4 Especificação de uma sessão de ensino on-line no FA_PorT

Segundo Silva (2000), uma sessão de ensino on-line (Figura 3.4)(Medeiros, 2007) associada a um novo sistema portfólio-tutor é especificada para a aprendizagem de um grupo virtual específico de alunos e é iniciada quando a camada tutor inicia uma estratégia. Uma estratégia de ensino no FA_PorT é representada por um conjunto de táticas de ensino. No contexto deste trabalho, uma nova tática de ensino foi especificada e incluída no sistema. A nova tática denomina-se: tática de fórum.

As táticas de ensino consideradas estão listadas abaixo:

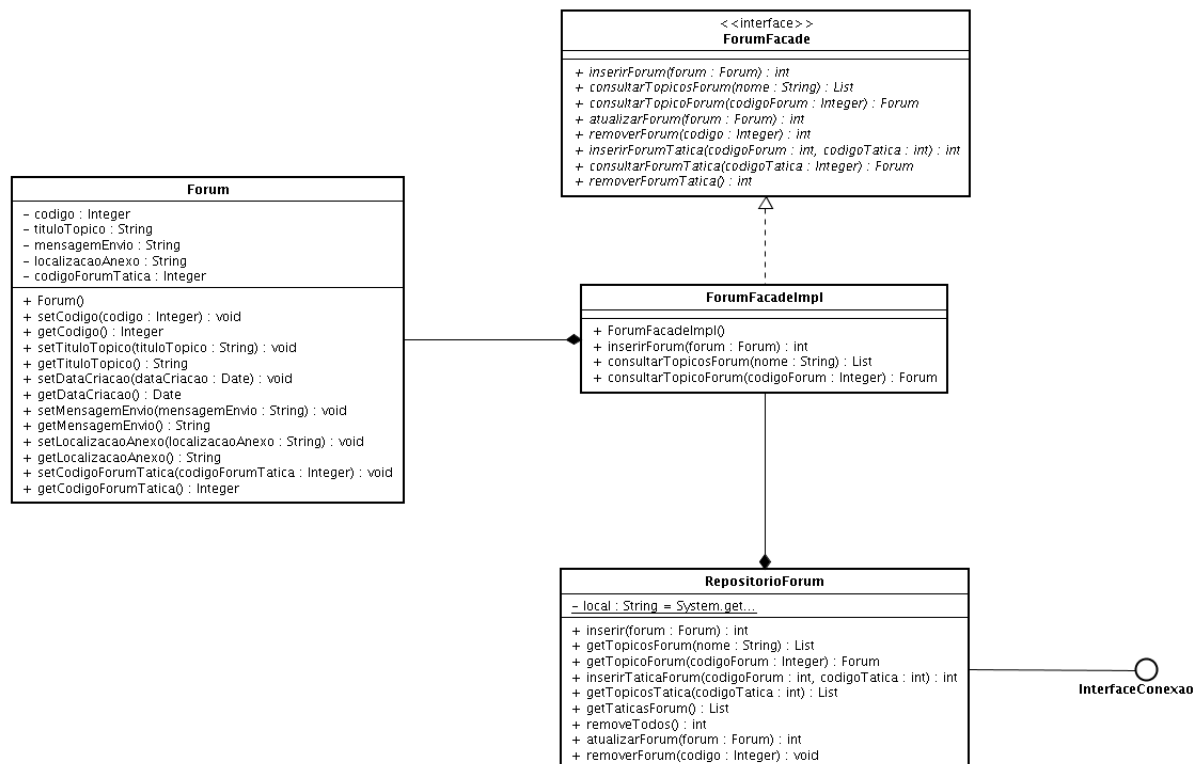


Figura 3.3: Diagrama de classes do Componente Fórum.

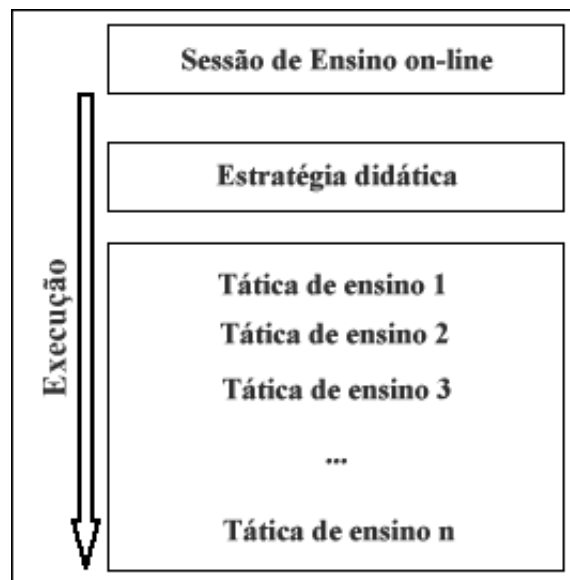


Figura 3.4: Representação de uma sessão de ensino on-line no FA_PorT.

- **Tática de Reuso:** indica que será apresentado algum recurso didático durante um tempo específico, como por exemplo, uma definição, um exemplo, um exercício ou um estudo de caso.

- Exemplo: Reuso(exercício, recurso, tempo, grupo)- este exemplo indica que um recurso didático do tipo exercício será mostrado por um determinado tempo, para um determinado grupo de alunos.
- **Tática de Debate Síncrono (Chat):** representa um chat ou bate-papo, onde em determinado grupo, alunos podem interagir entre eles e com o professor responsável pela sessão durante um tempo especificado.
 - Exemplo: DebateSincrono(professor, grupo, tempo)- um chat é ativado para o professor e os alunos por um período de tempo.
- **Tática de Envio de Informação:** envio de informações através de e-mail para alunos e professores contendo documentos que serão estudados pelos alunos.
 - Exemplo: EnvioInformacao(estudo de caso, recurso, grupo)- um e-mail é enviado contendo um recurso didático do tipo estudo de caso para um grupo de alunos.
- **Tática de Mudança de Estratégia:** permite a mudança de uma estratégia para uma outra, possibilitando o reuso de outras já utilizadas. Os professores selecionam estratégias bem sucedidas através de uma biblioteca de estratégias, podendo assim, reutilizá-las em uma nova sessão de ensino.
 - Exemplo: MudancaEstrategia(nova estratégia)- permite a mudança da estratégia atual para uma nova estratégia passada por parâmetro.
- **Tática de Geração de Relatório:** envio de relatórios contendo o desempenhos dos alunos a partir dos exercícios que são respondidos em uma tática de reuso, tanto para os alunos, quanto para os professores.
 - Exemplo: GeracaoRelatorio(professor, grupo)- envia através de e-mail o desempenho dos alunos para os professores e alunos.
- **Tática de Regra:** permite que uma condição seja verificada e algumas ações executadas, como por exemplo, uma mudança de estratégia ou envio de relatórios para o professor.
 - Exemplo: Regra(condição, acao1, acao2)- permite o desvio condicional na execução de uma estratégia didática, passando para a ação dependendo se a mesma é verdadeira ou não.
- **Tática de Fórum:** permite que alunos e professores selecionem um tópico referente a um assunto abordado e através do tópico possam trocar mensagens com anexos por um determinado tempo.

- Exemplo: Forum(professor, grupo, mensagem, anexo, tempo)- permite que professores e grupos de alunos possam trocar mensagens síncronas com anexos em um determinado período de tempo.

3.5 Funcionamento de um novo Sistema Portfólio-Tutor

Segundo Medeiros (2007), um novo sistema portfólio-tutor (ou aplicação) derivada do framework FA_PorT terá um funcionamento pré-definido. Isto acontece pelo fato do FA_PorT possuir um método template, onde este representa o controle de cada sistema portfólio-tutor. Cada aplicação gerada é estruturada em camadas conforme mostra a Figura 3.5.

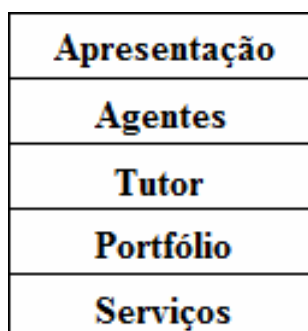


Figura 3.5: Camadas de um aplicação construída a partir do framework FA_PorT.

3.5.1 Camadas de uma nova aplicação

A camada de apresentação (ou interface) compreende a interface do aluno, professor e administrador.

A camada dos agentes é responsável pelas sessões de ensino on-line, desde o momento em que é iniciada, controlando as ações até o momento que é finalizada. Atualmente estão implementados três agentes: agente executar de sessões (responsável pelo controle das sessões de ensino), agente professor (responsável pela correção automática de exercícios) e o agente comunicação (responsável pela comunicação entre os agentes).

A camada tutor é responsável pelas táticas de ensino e as estratégias didáticas. Esta camada possui os componentes: base de domínio, perfil do aluno e estratégias didáticas.

A camada portfólio contém os componentes elementos administrativos (responsável pelas informações à respeito dos alunos, professores e disciplinas), registros (armazena o desempenho dos alunos nas atividades realizada pelo professor) e um comportamento pró-ativo. Este comportamento é responsável pelo quadro de avisos e pelo envio de informações por e-mail.

A camada serviços contém os componentes conexão com o banco de dados (acesso ao banco), segurança (controle de permissões) e comunicação (envio de informações por e-mail).

De acordo com a inclusão da nova tática de ensino (tática de fórum), fez-se necessária a criação de um novo componente e este acoplado ao FA_PorT. O componente de fórum foi criado para a troca de mensagens síncronas com a possibilidade de envio de arquivos, portanto, a nova especificação do FA_PorT conterá mais um componente na camada serviços.

3.6 Construção de Aplicações Portfólio-Tutor a partir do FA_PorT

Segundo Medeiros (2007), uma nova aplicação desenvolvida a partir do framework Fa_PorT é construída através de uma instância de classe que herda da classe abstrata *FrameworkPortfolioTutor*, como mostra o código 3.2. Cada nova aplicação construída possuirá um comportamento default que podem ser adaptadas através da redefinição de pontos adaptáveis (métodos) e da inclusão de novos componentes. O código 3.3 exemplifica os pontos adaptáveis atuais do framework FA_PorT.

Código 3.2: Nova aplicação gerada a partir do Framework

```

1
2 public PortfolioTutor extends FrameworkPortfolioTutor {
3
4     public PortfolioTutor () {
5         //comportamento default da nova aplicação
6         super ();
7     }
8 }
```

Código 3.3: Nova aplicação gerada e customizada a partir do Framework

```

1
2 public PortfolioTutor extends FrameworkPortfolioTutor {
3
4     public PortfolioTutor () {
5         super ();
6     }
7
8     public JFreeChart customizeGrafico(int aprovados ,
9         int reprovados , List desempenhos) {
10         List alunos = null;
11         try{
12             alunos = this.consultarAlunos("");
```

```

13         } catch (Exception e) {
14             System.out.println(e.getMessage());
15         }
16         DefaultCategoryDataset dataset =
17             new DefaultCategoryDataset();
18         for(Iterator j = alunos.iterator(); j.hasNext();){
19             Aluno aluno = (Aluno)j.next();
20             float totalNotas = 0;
21             int total = 0;
22             for(Iterator i = desempenhos.iterator();
23                 i.hasNext();){
24                 DesempenhoAluno desempenho =
25                     (DesempenhoAluno)i.next();
26                 if(desempenho.getAluno().getCodigo()
27 == aluno.getCodigo()){
28                     ItemAvaliar item = null;
29                     try{
30                         item = this.consultarItemAvaliar(
31                             desempenho.getItemAvaliar().getCodigo());
32                     } catch (Exception e) {
33                         System.out.println(e.getMessage());
34                     }
35                     total = total + (int)item.getPeso();
36                     totalNotas = totalNotas +
37                         desempenho.getNota() * item.getPeso();
38                     }
39                 }
40                 if(total != 0 && totalNotas != 0){
41                     dataset.setValue(totalNotas/total ,
42                                     aluno.getNome(), "");
43                 }
44             }
45         JFreeChart chart = ChartFactory.createBarChart(
46             "Desempenho_dos_Alunos","", "Nota", dataset ,
47             PlotOrientation.VERTICAL, true, true, true);
48
49         return chart;
50     }
51

```

3.6. CONSTRUÇÃO DE APLICAÇÕES PORTFÓLIO-TUTOR A PARTIR DO FA_PORT 27

```
52      public void enviarMensagem(Usuario usuario ,  
53          String mensagem) {  
54          //Código para envio  
55      }  
56 }
```

Capítulo 4

Padrões Pedagógicos no contexto de uma sessão de ensino no FA_PorT

Este capítulo irá abordar a utilização de dois padrões pedagógicos no contexto de uma sessão de ensino numa aplicação desenvolvida a partir do FA_PorT. Também serão apresentadas algumas ferramentas utilizadas durante o processo de desenvolvimento do novo componente, assim como na criação das sessões de ensino. A seção 4.1 apresenta algumas ferramentas utilizadas no desenvolvimento. A seção 4.2 apresenta a especificação do padrão pedagógico RCLAP. A seção 4.3 apresenta a especificação do padrão pedagógico LEASPE, e por fim, a seção 4.4 apresenta os padrões sendo utilizados numa aplicação desenvolvida a partir do FA_PorT.

4.1 Ferramentas Utilizadas

Alguns ajustes no FA_PorT foram necessários para o funcionamento de algumas funcionalidades existentes e para dar suporte a novas funcionalidades. Durante a especificação dos padrões pedagógicos percebeu-se a necessidade de incluir uma nova funcionalidade na estrutura do FA_PorT. A nova funcionalidade foi uma tática de ensino, chamada tática de fórum. Esta nova tática permitiu o desenvolvimento do componente fórum, onde este foi acoplado à estrutura do FA_PorT.

Para o desenvolvimento do novo componente no Fa_PorT foram utilizadas as seguintes ferramentas:

4.1.1 Java

De acordo com Horstmann & Cornell (2005), Java é toda uma plataforma, com uma vasta biblioteca, contendo quantidades de códigos reaproveitáveis, sendo também um ambiente de execução que fornece serviços como segurança, portabilidade para outros sistemas operacionais e coleta automática de lixo. Java foi lançada pela *Sun Microsystems* em maio de 1995 e até hoje

está sendo evoluída sempre com melhorias e adicionando novas extensões. (Sun, 2007).

Horstmann & Cornell (2005) apontam algumas características da linguagem java, tal como o fato dela ser uma linguagem orientada a objetos e interpretada, possuir verificação dinâmica em tempo de execução, possuir benefícios por ser multithread (responsividade interativa e comportamento em tempo real), não possuir o recurso de herança múltipla (substituído pelo conceito de interfaces), etc.

4.1.2 Webwork

Atualmente, com a necessidade de se construir sistemas Web de maneira organizada, estruturada e de fácil implementação tem-se possibilitado a utilização de arquitetura em três camadas. A arquitetura MVC (modelo-visão-controle) encaixa-se neste contexto, simplificando e agilizando o desenvolvimento e manutenção de sistemas na WEB.

A ferramenta Webwork foi construída para suportar este tipo de arquitetura em três camadas, fazendo o papel de controlador, abstraindo a camada de modelo da camada de visão. O grupo *Opensymphony* (quem abriga o Webwork), define o Webwork como um framework para o desenvolvimento de aplicações Web em Java, fornecendo recursos para o desenvolvimento ágil de aplicações, simplicidade na codificação, suporte robusto de templates para interfaces, internacionalização, validação de formulários, etc.

O framework Webwork é um projeto comunitário de código aberto, sendo desenvolvido e mantido através do apoio de diversas empresas que contribuem com a Opensymphony. O grupo mantém outros projetos de código aberto que servem de complemento para o Webwork e que trabalham, também, de maneira independente.

4.1.3 MySQL

O MySQL é um sistema (SGBD) robusto de gerenciamento de banco de dados SQL (Linguagem estruturada para consultas) rápida, multi-tarefa e multi-usuário. Pode ser usado em sistemas que são constantemente sobrecarregados com muitos dados e pode ser embutido em programa de uso em massa. (Manual de Referência do MySQL, 2006).

O MySQL é desenvolvido, distribuído e tem suporte da empresa MySQL AB. A MySQL AB é uma empresa comercial, fundada pelos desenvolvedores do MySQL, cujos negócios é fornecer serviços relacionados ao MySQL. (Manual de Referência do MySQL, 2006).

No Manual de Referência dos MySQL (2006) encontram-se algumas características da ferramenta, tais como: suporte à linguagens de programação (PHP, JAVA, C++, Eiffel, Python e Perl), funciona em diversas plataformas (Linux, Microsoft Windows, FreeBSD, MAC OS X), suporte ao ODBC (Open-DataBase-Connectivity) para Win32 (com fonte aberto), etc.

4.1.4 Eclipse

A IDE Eclipse é um projeto que faz parte da comunidade de código aberto Eclipse. Os projetos da comunidade Eclipse estão focados no desenvolvimento de plataformas de frameworks extensíveis, desenvolvimento e manutenção de softwares de acordo com o seu ciclo de vida, construção de ferramentas e *runtimes*.

A IDE Eclipse permite personalizar o ambiente de acordo com o projeto que está sendo desenvolvido, tanto Web, quanto desktop. Permite também, a utilização de plugins que trabalham integrados com as ferramentas. A versão da IDE utilizada neste trabalho de conclusão de curso foi a versão “Europa” (3.3), com os recursos disponíveis para ambiente J2EE: Servlets, JSP e XML.

4.2 Especificação do padrão Leitura, Crítica, Exposição Teórica, Atividade e Apresentação com discussão (RCLAP)

Esta seção abordará a especificação do Padrão Pedagógico RCLAP com ênfase numa aprendizagem significativa no contexto do FA_PorT. Isto se dará através da criação de uma estratégia de ensino. Inicialmente escolheu-se a UML e seus conceitos como domínio. A partir da escolha do domínio, foram coletados materiais sobre UML, e então, a elaboração de recursos didáticos. Para criar a estratégia, algumas táticas de ensino do FA_PorT foram utilizadas, juntamente com os recursos elaborados. A disposição das táticas para a estratégias foram alocadas de acordo com cada etapa do Padrão Pedagógico, utilizando o processo da aprendizagem significativa. As seções subseqüentes definem estes passos, assim como a estratégia a partir do padrão.

4.2.1 Conceitos e Recursos didáticos

O FA_PorT permite que o professor insira conceitos que ele deseja relacionar com os recursos didáticos. Cada vez que o professor irá criar um recurso didático, ele escolhe um conceito e o tipo de recurso que deseja disponibilizar. Atualmente o FA_PorT conta com os tipos de recurso: Definição, Exemplo, Estudo de Caso, Exercício automático (exercícios de verdadeiro ou falso e múltipla escolha) e avaliação. Abaixo seguem os conceitos definidos e os recursos baseados nesses conceitos.

Conceitos

1. Introdução da UML - Evolução, visão geral, etc.
2. Notação da UML - Modelagem, diagramas de caso de uso, classes, componentes, sequência, colaboração, etc.

Recursos didáticos

1. Definição - Conceito de UML.
2. Exemplo - Definição de UML.
3. Definição - Notação UML (Diagramas de Caso de Uso).
4. Estudo de Caso - Diagramas de Caso de Uso.
5. Exercício - Elaboração de Diagramas de Caso de Uso.

4.2.2 Definição da Estratégia de Ensino

Com os recursos definidos na subseção 4.2.1, o próximo passo será a definição das táticas de ensino, e então, a criação da estratégia didática. Como foi apresentado na seção 3.4, o FA_PorT possui sete táticas de ensino, sendo que, no contexto deste trabalho foram utilizadas apenas quatro.

Para a estratégia de ensino baseada no Padrão RCLAP, utilizou-se as táticas de Reuso, Debate síncrono (Chat), Envio de Informação por E-mail, e Geração de Relatório. Cada etapa da estratégia didática e as táticas escolhidas foram baseadas de acordo com os conceitos da aprendizagem significativa.

Estratégia Didática

- **Reading** (Leitura)

- Conceito Significativo: Organizadores Prévios (Através do conteúdo introdutório apresentado, espera-se que os alunos possam ter acesso a significados relevantes a sua estrutura cognitiva no contexto da UML).
- Tática de ensino: Reuso.
- Recurso didático: Definição de UML.

- **Critique** (Crítica)

- Conceito Significativo: Subsúnciores (Espera-se que os alunos, através de debate, possam ampliar ainda mais o conhecimento prévio abordado na etapa de Leitura).
- Tática de ensino: Debate Síncrono (Chat).
- Conhecimento prévio explorado de acordo com os organizadores prévios explicitados na etapa de *Reading* através de debate.

- **Lecture** (Exposição Teórica)

- Conceito Significativo: Organizadores Prévios (Com uma quantidade de subsunçores já existentes na estrutura cognitiva do aluno, novos organizadores prévios são utilizados para aprofundar o conteúdo).
 - Tática de ensino: Reuso.
 - Recurso didático: Exemplo (Definição de UML).
 - Tática de ensino: Reuso.
 - Recurso didático: Definição (Notação UML - Diagramas de Caso de Uso).
- **Activity** (Atividade)
 - Conceito Significativo: Assimilação e Subsunçores (Com os conhecimentos previamente adquiridos nas etapas anteriores, espera-se que os exercícios ocasionem a interação de novos significados com os já existentes, aprimorando-os e possibilitando o surgimento de novos subsunçores).
 - Tática de ensino: Envio de Informação Por E-mail.
 - Recurso didático: Exercício de elaboração de Diagramas de Caso de Uso.
- **Presentation With Discussion** (Apresentação com discussão)
 - Conceito Significativo: Diferenciação Progressiva (A medida que o aluno vai aprendendo significativamente os conceitos sobre UML (introdução, notação, etc.), os subsunçores vão se tornando cada vez mais elaborados, mais diferenciados, servindo de âncoras para atribuições de significados a novos conhecimentos).
 - Tática de ensino: Debate Síncrono (Chat).
 - Exploração do conteúdo no chat: Questionamentos sobre o exercício e explanação dos casos de uso elaborados.

4.3 Especificação do padrão Exposição Teórica, Exemplos, Atividades, Apresentação e Avaliação (LEASPE)

Esta seção abordará a especificação do Padrão Pedagógico LEASPE com ênfase numa aprendizagem significativa no contexto do FA_PorT. Isto se dará através da criação de uma estratégia de ensino. Inicialmente escolheu-se os conceitos sobre Frameworks como domínio. A partir da escolha do domínio, foram coletados materiais sobre Framework (fundamentos, modelagem, estrutura, exemplos de códigos fonte), e então, a elaboração de recursos didáticos. Para criar a estratégia, algumas táticas de ensino do FA_PorT foram utilizadas, como citado na seção 4.2.

4.3.1 Conceitos e Recursos didáticos

Vide sub-seção 4.2.1 para maiores detalhes sobre conceitos e recursos didáticos.

Conceitos

1. Introdução a Framework - Fundamentos.
2. Especificação de um Framework - Notação UML-F.
3. Exemplo de Framework - Código.
4. Estrutura de um Framework.

Recursos didáticos

1. Definição - Introdução a Frameworks.
2. Exemplo - Exemplos de frameworks utilizando códigos fonte.
3. Definição - Especificação de um Framework.
4. Definição - Estrutura de um Framework.
5. Exercício - Uma questão de verdadeiro ou falso e duas questões objetivas.

4.3.2 Definição da Estratégia de Ensino

Vide subseção 4.2.2 para maiores detalhes sobre definição de uma estratégia didática no FA_PorT.

Estratégia Didática

- **Lecture** (Exposição Teórica)
 - Conceito Significativo: Organizadores Prévios (Materiais introdutórios referentes a conceito de Frameworks, especificação e estrutura são apresentados ao aluno);
 - Tática de ensino: Reuso;
 - Recurso didático: Definição (Introdução a Frameworks);
 - Tática de ensino: Reuso;
 - Recurso didático: Especificação de um Framework (Notação UML-F);
 - Tática de ensino: Reuso;
 - Recurso didático: Estrutura de um Framework.

- **Examples** (Exemplos)

- Conceito Significativo: Organizadores Prévios (Novos organizadores prévios em forma de exemplos são utilizados para fazer pontes entre os significados que o aluno já tem e os que ele precisaria ter para aprender significativamente sobre Framework);
- Tática de ensino: Reuso;
- Recurso didático: Exemplo (Exemplos de frameworks);

- **Activity** (Atividade)

- Conceito Significativo: Subsunçores (Espera-se que com os exercícios aplicados, novos significados interajam com os já existentes na estrutura cognitiva do aluno);
- Tática de ensino: Reuso;
- Recurso didático: Uma lista de exercícios contendo o que foi abordado anteriormente. Os exercícios conterão uma proposição de verdadeiro ou falso e duas proposições objetivas, onde os alunos irão modelar pequenos exemplos;

- **Student Presentation** (Apresentação)

- Conceito Significativo: Assimilação e Subsunçores (À medida que o conhecimento prévio das etapas anteriores serve de base para a atribuição de significados à nova informação, os subsunçores vão adquirindo novos significados, se tornando mais diferenciados, mais estáveis. Isto irá possibilitar a formação de novos subsunçores);
- Tática de ensino: Fórum;
- Tópico do Fórum: Questionamentos sobre os exercícios e conteúdos apresentados durante a fase de *lecture*, *example* e *activity*;

- **Evaluation** (Avaliação)

- Conceito Significativo: Reconciliação Integrativa (Conceitos sobre Frameworks existentes na estrutura cognitiva dos alunos, já estáveis e diferenciados são percebidos como relacionados, adquirem novos significados e levam a uma reorganização da estrutura cognitiva. Isto possibilitará aos alunos verem os conceitos de maneira mais abrangente e com um certo grau de clareza.);
- Tática de ensino: Fórum;
- Tópico do Fórum: Avaliação sobre o conteúdo apresentado em *Student presentation*, apontando erros e outras soluções possíveis.

4.4 Utilização dos Padrões Pedagógicos RCLAP e LEASPE no contexto de sessões de ensino no FA_PorT

Esta seção irá mostrar os dois padrões especificados no contexto de uma sessão de ensino, juntamente com a nova tática de ensino em funcionamento. Para testar e analisar a aplicabilidade dos padrões com o FA_PorT foi necessária a utilização de uma aplicação derivada a partir do Fa_PorT (Medeiros, 2007). A aplicação escolhida possui um comportamento default, logo não foi preciso redefinir os pontos adaptáveis. No comportamento default, o gráfico que será exibido está no formato de *pizza*.

A Figura 4.1 mostra a tela inicial da aplicação, onde o usuário deverá efetuar login para ter acesso às funcionalidades do sistema.



Figura 4.1: Tela inicial da aplicação portfólio-tutor.

4.4.1 Sessão de ensino baseada no Padrão Pedagógico RCLAP

Na tabela 4.1 é apresentada a sessão de ensino baseada no padrão pedagógico RCLAP, sendo representada através das táticas de ensino definidas na seção 3.4.

Tabela 4.1: Táticas de ensino da estratégia que representa o padrão pedagógico RCLAP

Leitura Crítica	Tática de Reuso(Evolução de UML, 10min); Tática de Debate Síncrono(Grupo nível básico, Professor, 15min);
Exposição Teórica	Tática de Reuso(Exemplo de Definição de UML, 10min); Tática de Reuso(Definição da notação UML, 10min);
Atividade	Tática de Reuso(Exercício V/F, 10min) Tática de Geração de Relatório(Professor, Grupo nível básico);
Apresentação com discussão	Tática de Envio de Informação por E-mail(Elaboração de diagramas de caso de uso, estudo de caso, 10min); Tática de Debate Síncrono(Grupo nível básico, Professor, 15min);

A seguir serão mostradas algumas telas referentes aos passos executados durante uma sessão de ensino on-line, onde o aluno terá a possibilidade de visualizar o material disponibilizado e interagir com outros alunos e o professor para discussões mais aprofundadas.

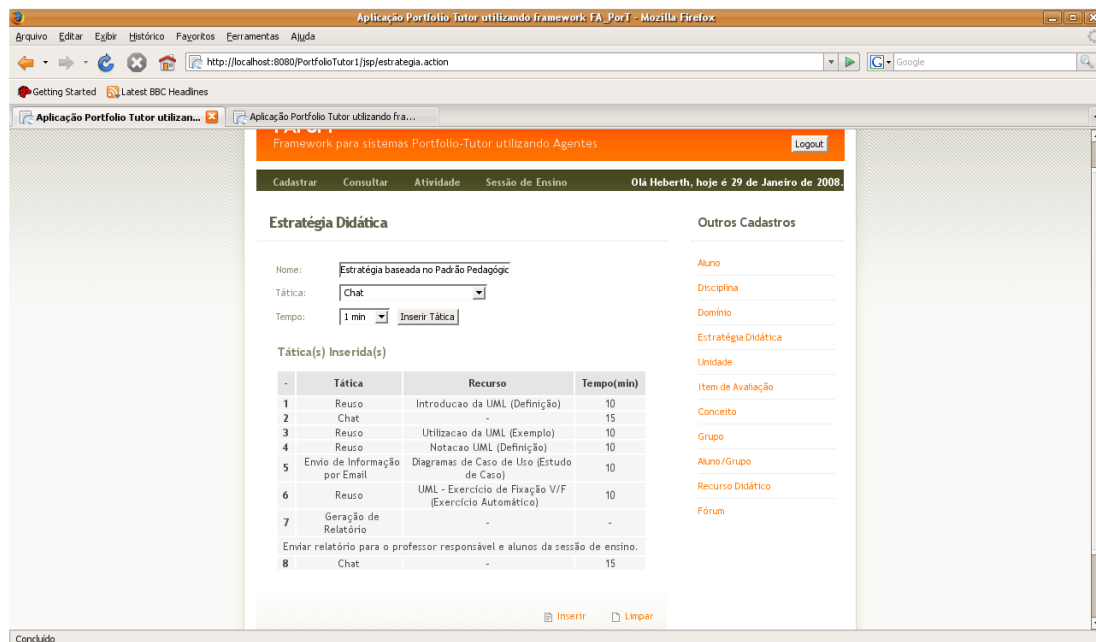


Figura 4.2: Tela da estratégia didática baseada no padrão pedagógico RCLAP.

Para que uma sessão de ensino funcione é necessário a utilização de estratégias didáticas, juntamente com as táticas de ensino. A Figura 4.2 apresenta a tela referente a estratégia didática baseada no padrão pedagógico RCLAP e as táticas associadas.

A Figura 4.3 apresenta a tela de criação da sessão de ensino baseada na estratégia didática RCLAP, citada anteriormente.

A Figura 4.4 apresenta a tela exibindo a tática de ensino “Reuso” para definição da UML com a opção de visualizar o recurso didático em anexo. Como as etapas utilizando a tática de reuso exibem a mesma funcionalidade, então, só será exibida a tela para definição da UML.

A Figura 4.5 apresenta a tela que será exibida ao aluno quando este clicar no recurso em anexo.

A Figura 4.6 apresenta a tela da tática de ensino de exercício verdadeiro ou falso durante a etapa de “Atividade” do padrão RCLAP.

4.4. UTILIZAÇÃO DOS PADRÕES PEDAGÓGICOS RCLAP E LEASPE NO CONTEXTO DE SESSÕES DE ENSINO NO FA_PORT

37

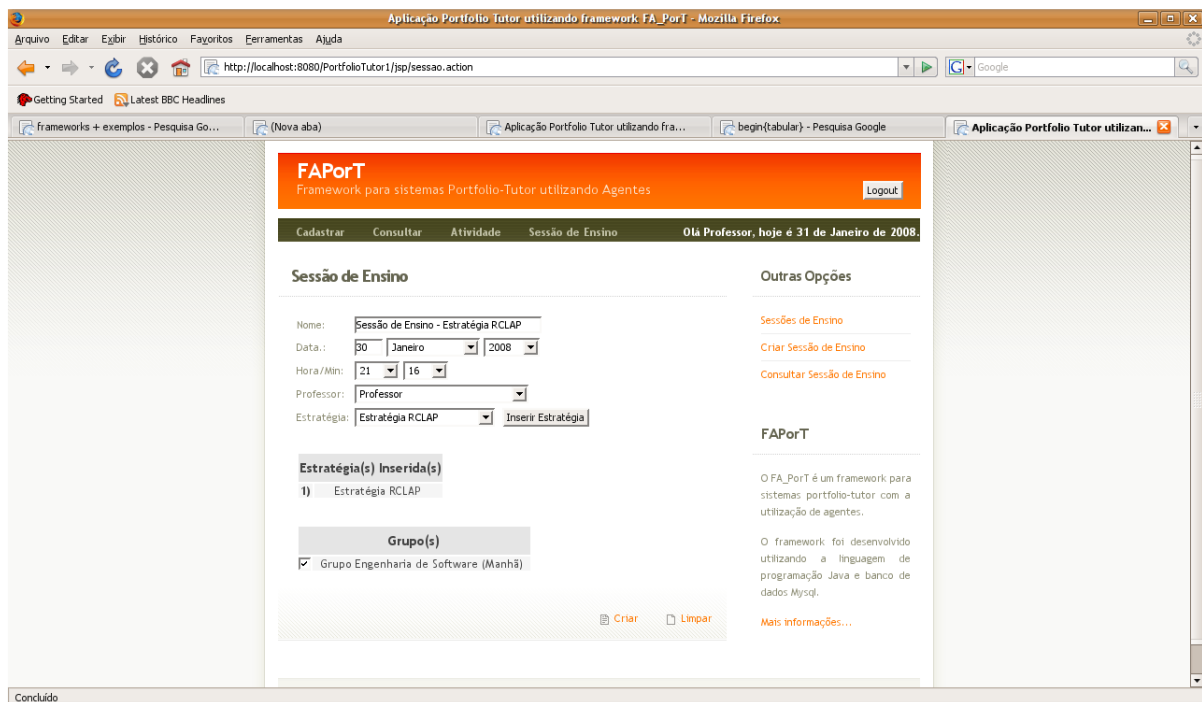


Figura 4.3: Tela da criação da sessão de ensino baseada na estratégia didática RCLAP.

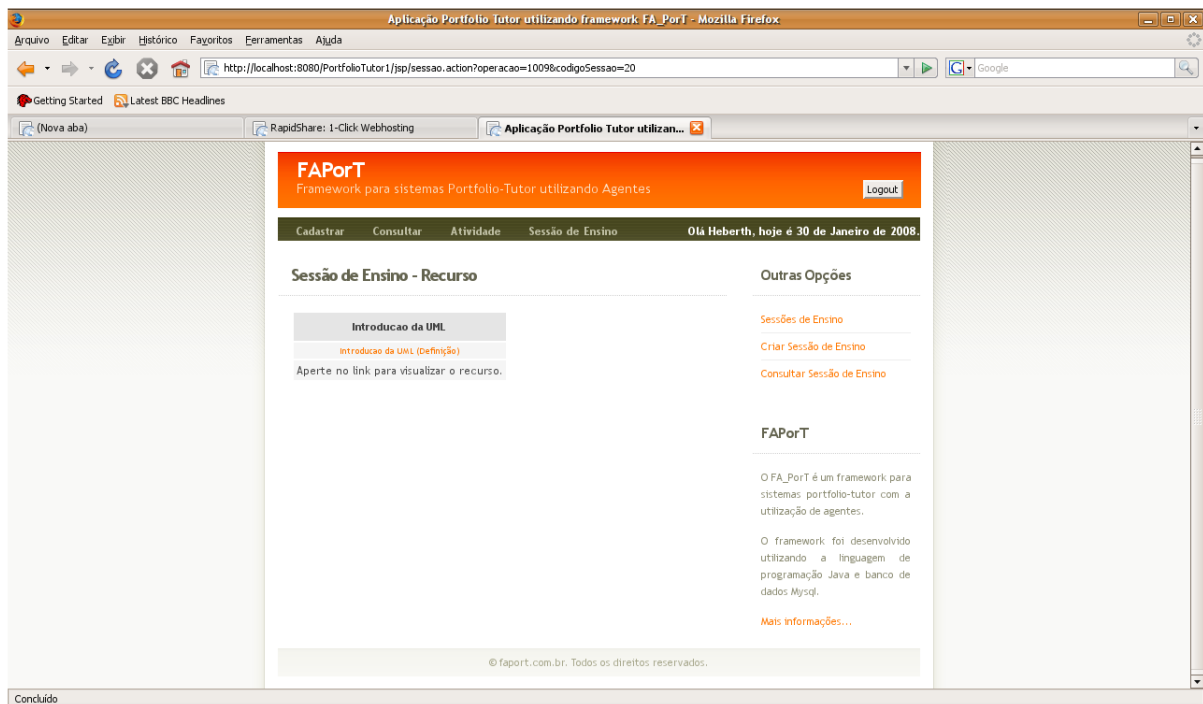


Figura 4.4: Tela da tática de ensino “Reuso” para Evolução da UML.

A figura 4.7 apresenta a tática de geração de relatório mostrando o desempenho do aluno no exercício V/F. O relatório é enviado através do e-mail para os alunos e/ou professor.

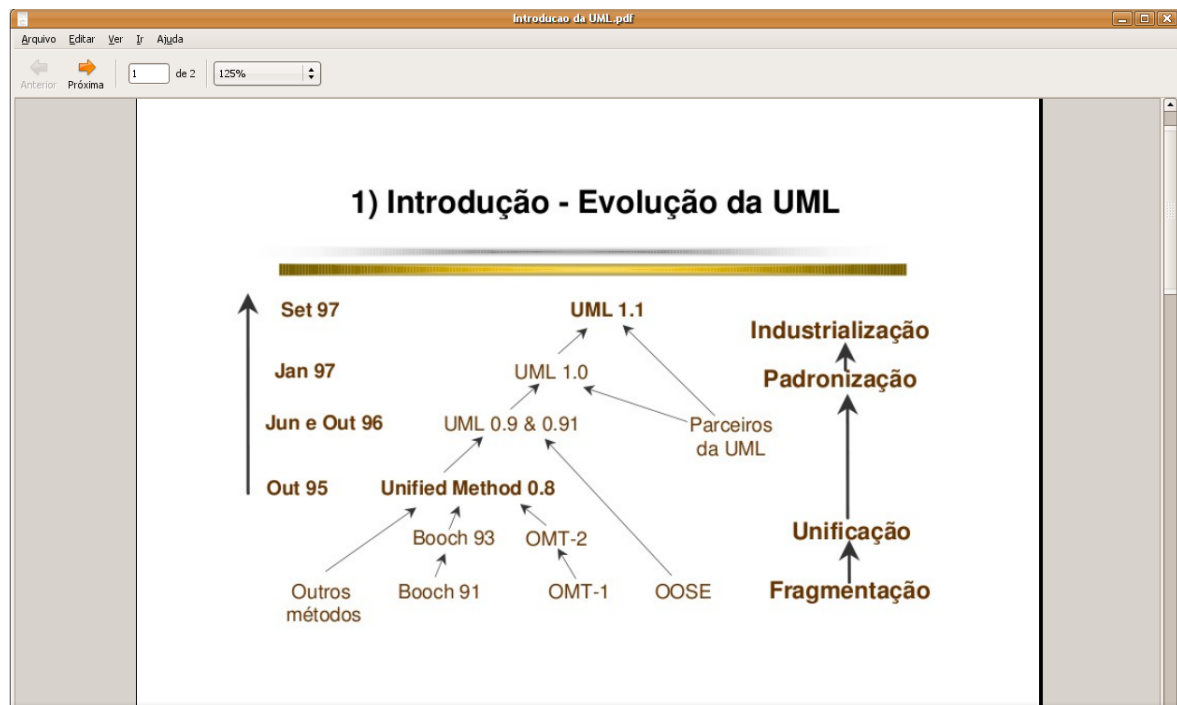


Figura 4.5: Tela exibindo o recurso didático em anexo.

A interface do FAPoRT, acessada via Mozilla Firefox, apresenta uma barra de navegação com "Consultar", "Atividade" e "Sessão de Ensino". O cabeçalho indica a data: "Olá Joao, hoje é 30 de Janeiro de 2008.". A seção "Sessão de Ensino - Recurso" contém o título "UML - Exercício de Fixação V/F" e cinco perguntas com respostas selecionadas em menus suspensos:

- 1) Um diagrama de caso de uso descreve o comportamento de um sistema percebida por fluxos externos. Resposta: Verdadeiro
- 2) Casos de uso definem funcionalidades e comportamentos fornecidos pelo sistema. Resposta: Falso
- 3) Representa um papel que um ser humano, um dispositivo de hardware desempenha quando interage com o sistema. Resposta: Falso
- 4) Casos de uso validam e verificam o desenvolvimento do sistema. Resposta: Verdadeiro
- 5) Um caso de uso não descreve o que um sistema não faz. Resposta: Falso

À direita, há uma seção "FAPoRT" com uma descrição do framework e links para "Portal da Educação", "Ciência e Tecnologia" e "Informação Científica".

Figura 4.6: Tela da tática de ensino “Exercício” para o exercício de verdadeiro ou falso.

A Figura 4.8 apresenta a tela exibindo a tática de debate síncrono utilizada na etapa de “Apresentação com discussão” do padrão RCLAP.

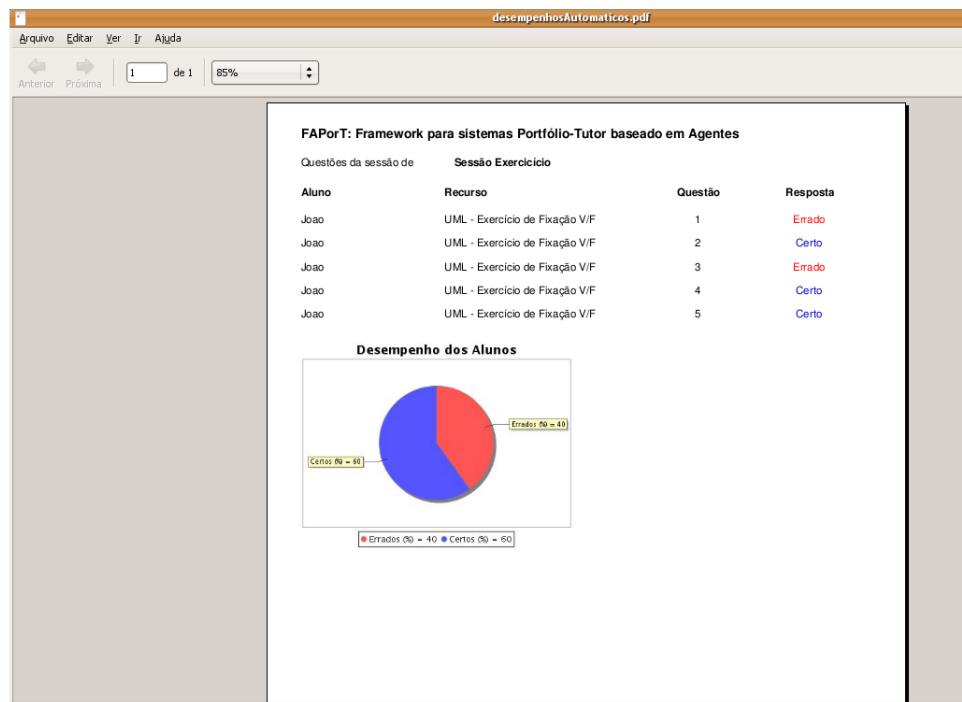


Figura 4.7: Tela da tática de ensino “Geração de relatório”.



Figura 4.8: Tela da tática de ensino debate síncrono.

4.4.2 Sessão de ensino baseada no Padrão Pedagógico LEASPE

Na tabela 4.2 é apresentada a sessão de ensino baseada no padrão pedagógico LEASPE, sendo representada através das táticas de ensino definidas na seção 3.4.

A seguir serão mostradas algumas telas referentes aos passos executados durante uma sessão

Tabela 4.2: Táticas de ensino da estratégia que representa o padrão pedagógico LEASPE

Exposição Teórica	Tática de Reuso(Definição de Framework, 10min); Tática de Reuso(Especificação de Framework(Definição), 10min); Tática de Reuso(Estrutura dos Frameworks, 10 min); Tática de Reuso(Exemplos de Frameworks, Professor, 10min);
Exemplos	Tática de Reuso(Lista de exercícios, 10min);
Atividade	Tática de Fórum(Professor, Grupo nível básico, mensagem no fórum, arquivo anexado, 25min);
Apresentação	Tática de Fórum(Professor, Grupo nível básico, mensagem no fórum, arquivo anexado, 25min);
Avaliação	

de ensino on-line, onde o aluno terá a possibilidade de visualizar o material disponibilizado e interagir com outros alunos e o professor para discussões mais aprofundadas. O procedimento que será utilizado a seguir é semelhante ao utilizado na sub-seção 4.4.1.

A Figura 4.9 apresenta a tela referente a estratégia didática baseada no padrão pedagógico LEASPE e as táticas associadas.

The screenshot displays the 'FA_PoRT' web application interface. The main section is titled 'Estratégia Didática' (Didactic Strategy). It includes a 'Nome:' field with the value 'Estratégia LEASPE', a 'Tática:' dropdown menu set to 'Fórum', and a 'Tempo:' field set to '1 min'. Below these are checkboxes for 'Inserir Tática' and 'Inserir Recurso'. A table titled 'Tópico(s) do Fórum' lists several topics, including 'Teste Framework Fórum', 'Introdução a UML', 'Apresentação dos exercícios sobre fundamento, estrutura e especificação de UML - Geral', and 'Avaliação sobre fundamento, estrutura e especificação de Framework'. To the right, there is a section for 'Outros Cadastros' (Other Registrations) with links for 'Aluno', 'Disciplina', 'Domínio', 'Estratégia Didática', 'Unidade', 'Item de Avaliação', 'Conceito', 'Grupo', 'Aluno/Grupo', 'Recurso Didático', and 'Fórum'. At the bottom, there is a table titled 'Tática(s) Inserida(s)' (Inserted tactic(s)) with columns for 'Tática', 'Recurso', and 'Tempo(min)'. It lists six tactics, all of type 'Reuso', with durations ranging from 10 to 25 minutes. The interface also shows a 'Logout' button and a date 'Ola Professor, hoje é 31 de Janeiro de 2008'.

Figura 4.9: Tela da estratégia didática baseada no padrão pedagógico LEASPE.

A Figura 4.10 apresenta a tela exibindo a tática de ensino “Reuso” para definição de Framework com a opção de visualizar o recurso didático em anexo. Como as etapas utilizando a

tática de reuso exibem a mesma funcionalidade, então, só será exibida a tela para definição de Framework.

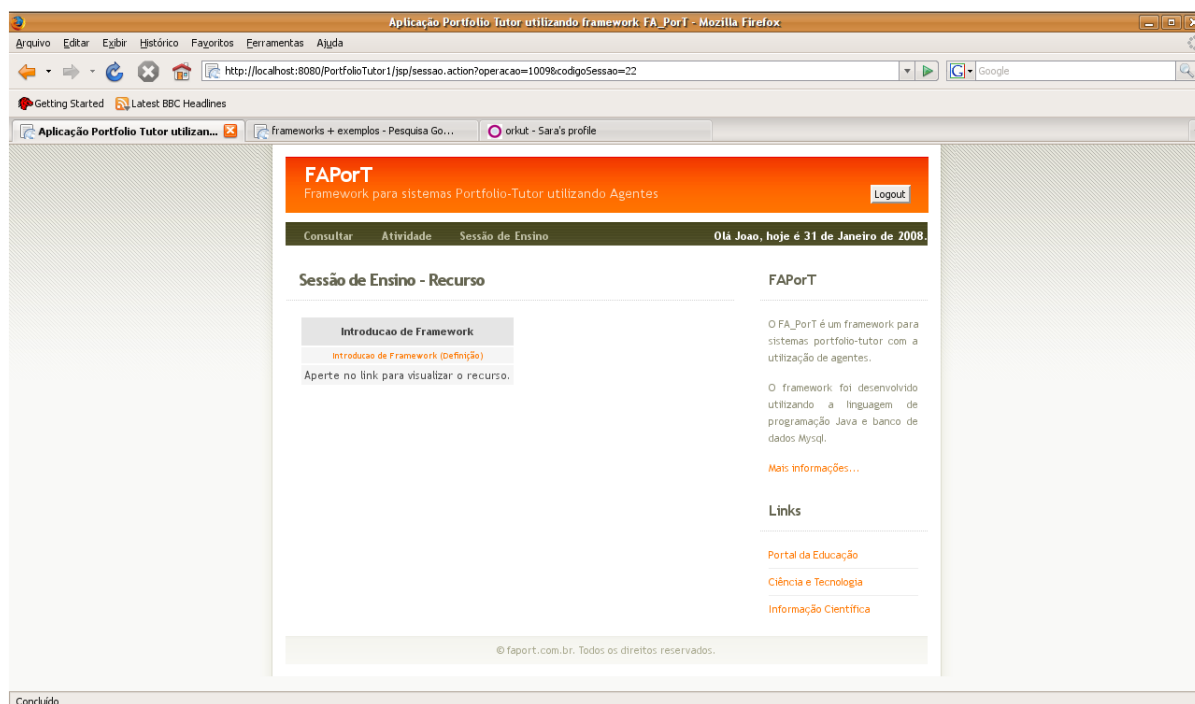


Figura 4.10: Tela da tática de ensino “Reuso” para Introdução sobre Framework.

A Figura 4.11 apresenta a tela que será exibida ao aluno quando este clicar no recurso em anexo.

A Figura 4.12 apresenta a nova funcionalidade implementada no FA_PorT: a tática de ensino “Fórum”. Durante a especificação do padrão LEASPE, percebeu-se que algumas das etapas precisavam de uma interação maior entre o professor e os alunos. O FA_PorT já possuía o debate síncrono (Chat) para a troca de mensagens entre aluno e professor, porém tornou-se limitada, uma vez que se desejava a troca de arquivos simultaneamente para uma maior interação e enriquecimento de conhecimento.

O “Fórum” no FA_PorT funciona de maneira bastante simples: o professor cria um ou mais tópicos referentes ao assunto que será abordado na sessão de ensino. Na sessão de ensino, especificamente na tática de Fórum, serão exibidos os tópicos e a data em que foram criados. Tanto o aluno, quanto o professor, ao clicarem no tópico correspondente, serão redirecionados para a tela de envio de mensagens com arquivos em anexo, como mostra a Figura 4.13.

A Figura 4.14 mostra as mensagens já postadas e a exibição do arquivo ao clicar no link no rodapé do corpo da mensagem.

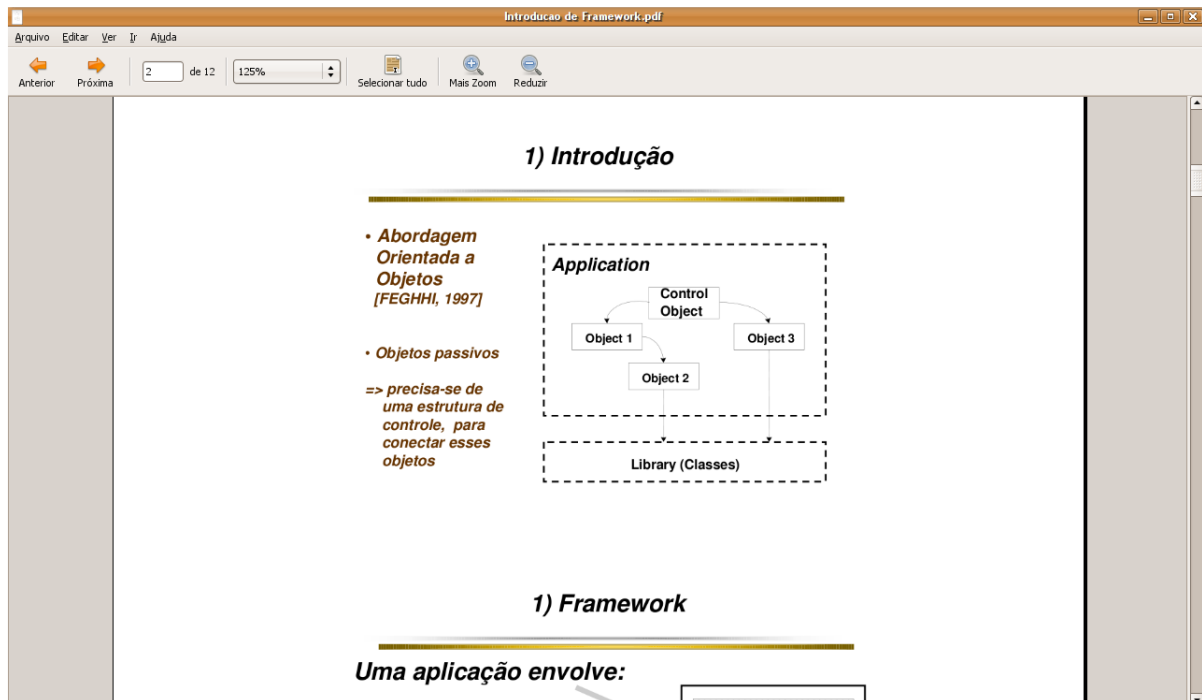


Figura 4.11: Tela exibindo o recurso didático em anexo.

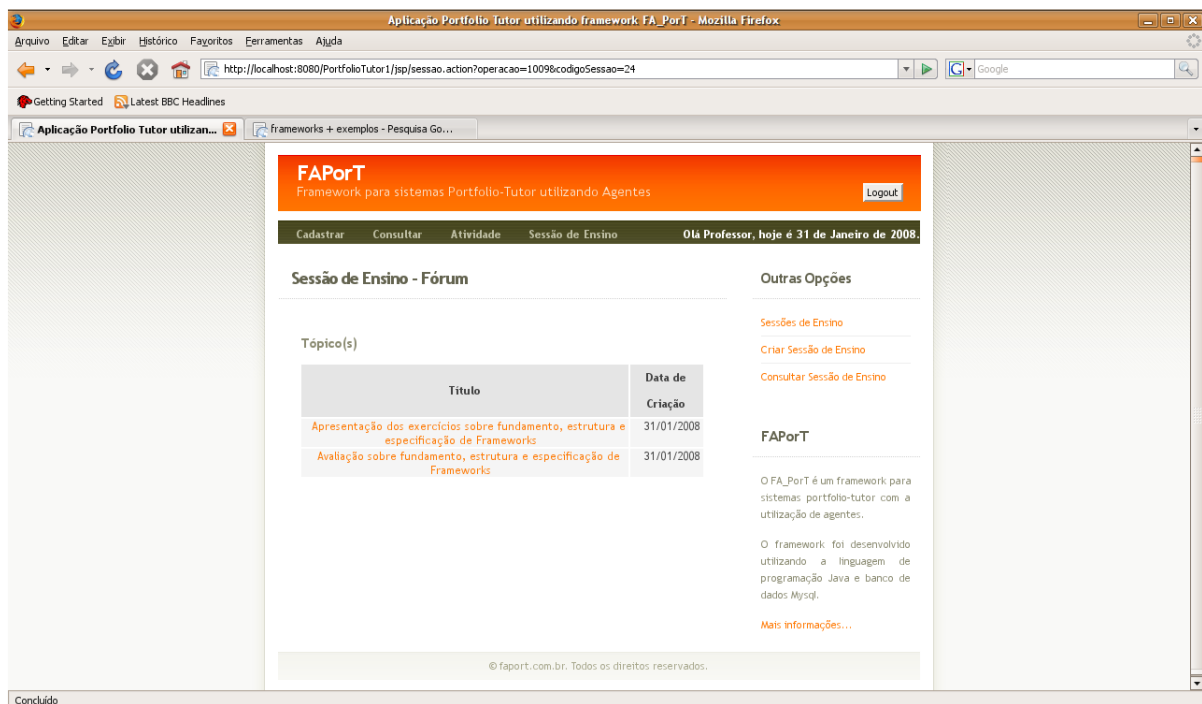


Figura 4.12: Tela exibindo os tópicos disponíveis durante a tática de Fórum numa sessão de ensino.

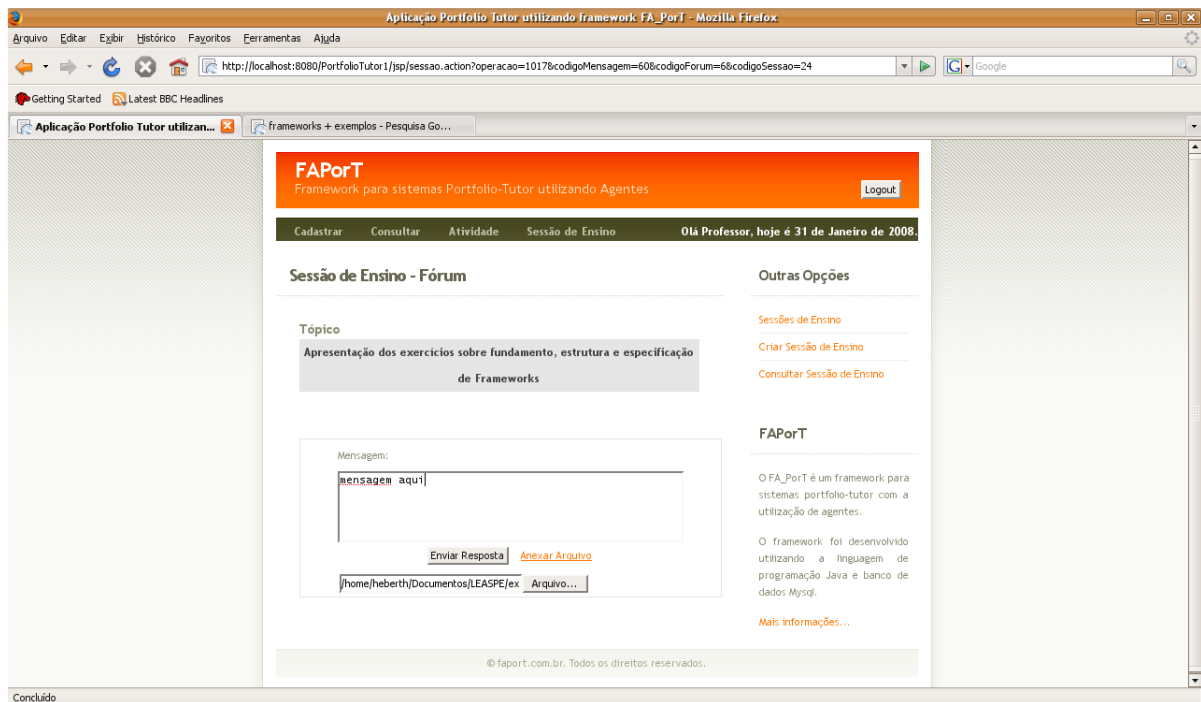


Figura 4.13: Tela exibindo as mensagens enviadas.

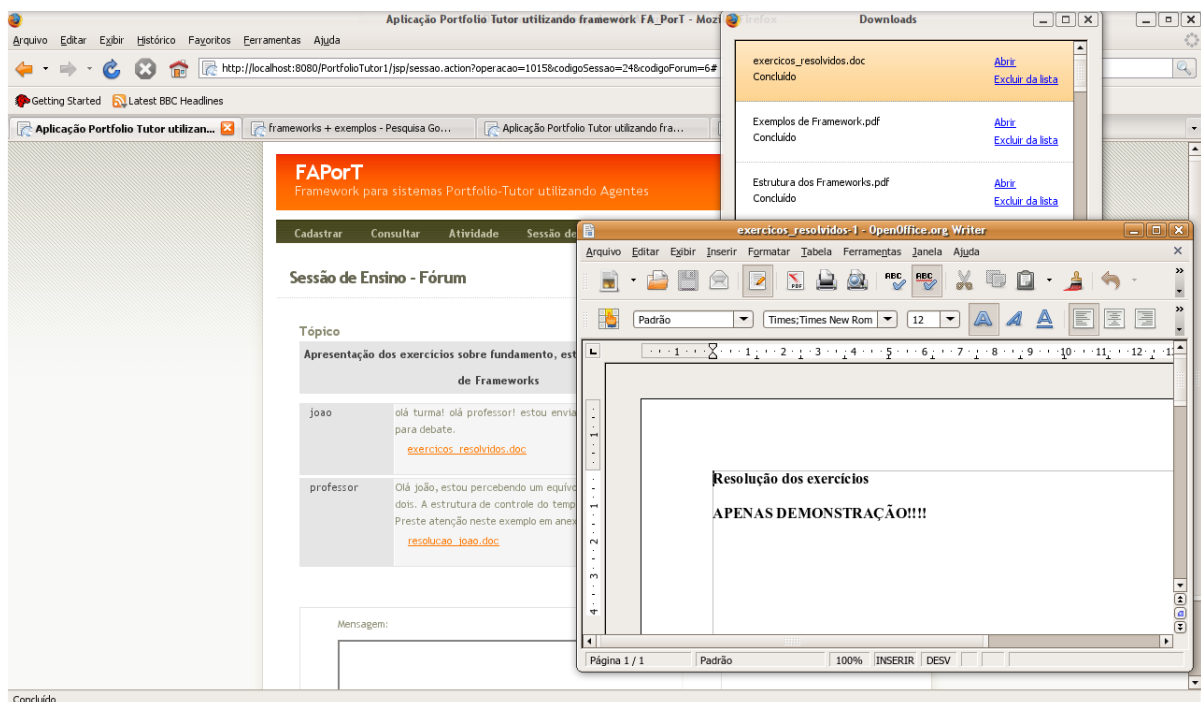


Figura 4.14: Tela exibindo as mensagens enviadas e um anexo.

Capítulo 5

Considerações Finais

A proposta desta monografia teve como objetivo: a utilização de dois padrões pedagógicos no contexto de sessões de ensino on-line a partir de uma aplicação portfólio-tutor construída através do framework FA_PorT.

Os padrões pedagógicos foram especificados através do processo da aprendizagem significativa proposto por Ausubel (1980), permitindo ao aluno uma maior interação com as funcionalidades dispostas pelo sistema portfólio-tutor, além de motivá-lo na aquisição de conhecimentos relevantes a sua estrutura cognitiva.

5.1 Resultados Obtidos

A partir das funcionalidades que a aplicação derivada do FA_PorT dispõe, podem ser citados alguns resultados:

- A aplicação portfólio-tutor permite que os professores disponibilizem recursos didáticos, além de poder criar sessões de ensino on-line representadas através de estratégias didáticas. As estratégias são compostas por táticas de ensino: reuso, envio de informações por e-mail, debate síncrono (Chat), geração de relatório, tática de regra, mudança de estratégia, etc.
- Com a análise do sistema portfólio-tutor, percebeu-se que era possível aplicar padrões pedagógicos no contexto de uma sessão de ensino on-line. Os padrões pedagógicos podem auxiliar os professores na organização de estratégias didáticas a partir do nível de conhecimento de cada aluno.
- Foram feitas análises de três padrões pedagógicos no contexto da orientação a objetos, porém, apenas dois foram escolhidos como estudo de caso para este trabalho. Os padrões foram escolhidos de acordo com o que o sistema portfólio-tutor permitisse no contexto das sessões de ensino on-line. Com os dois padrões escolhidos, partiu-se então para suas especificações, utilizando o processo de aprendizagem significativa. O processo de

aprendizagem significativa permite definir cada etapa do padrão pedagógico de maneira significativa, disponibilizando materiais potencialmente significativos, por exemplo, na elaboração de recursos didáticos. Isto resulta na motivação do aluno em querer aprofundar mais seu conhecimento e ao professor, que terá uma base para preparar suas aulas.

- Com a especificação dos padrões através da aprendizagem significativa e a utilização destes nas sessões de ensino, percebeu-se a necessidade de incluir uma nova tática de ensino ao FA_PorT. A nova tática de ensino adicionada foi a tática de Fórum. A nova tática mostrou-se de grande utilidade no processo de aprendizagem, visto que, a interação entre professor, alunos e conteúdo disponível é mais satisfatório que apenas uma sessão de Chat. A tática de fórum permite a troca de mensagens e arquivos entre alunos e professor simultaneamente.

5.2 Sugestões Para Trabalhos Futuros

- Analisar, especificar e implementar outros padrões pedagógicos disponíveis no sítio do Projeto de Padrões Pedagógicos (Pedagogical Patterns Project, 2007).
- Permitir que o professor possa registrar o planejamento de suas aulas a partir da elaboração de mapas conceituais proposto por Ausubel.
- Implementar novas funcionalidades no FA_PorT, como por exemplo, permitir que o aluno avise, durante uma etapa da sessão de ensino, que ele já acabou de ler um conteúdo ou resolveu os exercícios. Isto possibilitará que ele avance para a outra etapa sem esperar que o tempo de cada tática de ensino se esgote. Isso vale também para o professor que poderá ir acompanhando o aluno.

Referências Bibliográficas

- AUSUBEL, D. P., NOVAK, J. D. & HANESIAN, H. (1980), *Psicologia Educacional*, 2 edn, Interamericana - RJ.
- FARIAS, A. J. O. (1982), Mapeamento cognitivo em um curso individualizado: Um estudo sobre o efeito da abordagem ao conteúdo, Dissertação de mestrado, Universidade Federal do Rio Grande do Sul.
- FAYAD, M. E. & SCHMIDT, D. C. (1997), 'Object-oriented application frameworks', *Communications of the ACM* **40**, 32–38.
- GAMMA, E., HELM, R., JOHNSON, R. & VLISSIDES, J. M. (2005), *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley.
- HORSTMANN, C. S. & CORNELL, G. (2005), *Core Java 2: Fundamentos*, Vol. 1, 7 edn, The Sun Microsystems Press : Java Series.
- KATZ, S., DABROWSKI, C. & LAW, M. (1993), Glossary of software reuse terms, Prepared for the department of defense/ballistic missile defense organization, U.S. Department of Commerce/National Institute of Standards and Technology.
- LILLY, S. (1996), 'Patterns for pedagogy', *Object Magazine* **5(8)**, 93–96.
- MARTINS, M. (2003), 'Guia para iniciantes do webwork', *Grupo de Usuários Java*, <http://www.guj.com.br>.
- MEDEIROS, F. M. (2007), Implementação do fa_port: Framework para sistemas portfólio-tutor utilizando agentes, Trabalho de conclusão de curso, Universidade Federal de Alagoas.
- MEDEIROS, F. M., HERNÁNDEZ-DOMÍNGUEZ, A. & de MEDEIROS, F. N. (2007), 'Um sistema de ensino na web baseado no padrão pedagógico exposição teórica-exemplos-atividade-apresentação-avaliação', *Anais do XVIII Simpósio Brasileiro de Informática na Educação* **1**, 133–144.

- MEDEIROS, F. N. d. (2006), Fa_port: Framework para sistemas portfólio-tutor baseado em agentes, Dissertação de mestrado em modelagem computacional de conhecimento, Instituto de Computação, Universidade Federal de Alagoas.
- MEDEIROS, F. N. D., MEDEIROS, F. M. & HERNÁNDEZ-DOMÍNGUEZ, A. (2006), 'Fa_port: Um framework para sistemas portfólio-tutor utilizando agentes', *XVII Simpósio Brasileiro de Informática na Educação (Anais)* pp. 407–416.
- MOORE, M. G. & KEARSLEY, G. (2005), *Educação a Distância: Uma Visão Integrada*, Thomson.
- MySQL (2006), '**Manual de Referência do MySQL**'. Disponível em: <http://dev.mysql.com/doc/>, último acesso em 25 de janeiro de 2008.
- NASCIMENTO, D. M. C., HERNÁNDEZ-DOMÍNGUEZ, A. & SCHIEL, U. (2002), 'Portfólio-tutor: Um tutor acoplado a um portfólio eletrônico no contexto da educação a distância', *SBIE 2002 - Simpósio Brasileiro de Informática na Educação, XIII - Simpósio Brasileiro de Informática na Educação*.
- NETO, J. A. D. O. (2000), Suporte de ferramenta de software para o padrão pedagógico aula em mapa de conceitos, Dissertação de mestrado, Universidade Federal da Paraíba.
- NOBRE, V. M., MOTTA, C. L. & ELIA, M. (2003), 'Projeto de aprendizagem - um proposta de interface gráfico-pedagógico para *e-learning*', *XIV SBIE Simpósio Brasileiro de Informática na Educação* **1**, 35–44.
- NUNES, S. d. C. & SANTOS, R. P. D. (julho de 2006), 'Análise pedagógica de portais educacionais conforme a teoria da aprendizagem significativa', *Universidade Federal do Rio Grande do Sul, CINTED-UFRGS - Revista Novas Tecnologias na Educação* **4**(2).
- OLIVEIRA, T. J. V. d. (2005), Implementação e aplicação do padrão pedagógico lecture - examples - activity - student presentation - evaluation no contexto de um sistema de ensino à distância, Trabalho de conclusão de curso, Universidade Federal de Alagoas.
- OPENSYPHONY (2007), '**Webwork web application framework**'. Disponível em: <http://www.opensymphony.com/webwork/>, último acesso em 26 de dezembro de 2007.
- PRENTZAS, J. & HATZILYGEROUDIS, I. (2004), Knowledge representation requirements for intelligent tutoring systems, in '7th International Conference, ITS (Intelligent Tutoring Systems), Maceió, Alagoas, Brasil', Springer-Verlag, pp. 87–97.
- Project, P. P. (2007), "Pedagogical patterns project", <http://www.pedagogicalpatterns.org/>, último acesso em 30 de dezembro de 2007.

- SAUVÉ, J. P. (2000), '**Projeto De Software Orientado a Objeto**'. Disponível em: <http://www.dsc.ufcg.edu.br/jacques/cursos/map/html/frame/oque.htm>, último acesso em 15 de janeiro de 2008.
- SILVA, A. M. (2007), Sistemas de ensino na web utilizando padrões pedagógicos com ênfase na aprendizagem significativa, Proposta de projeto de dissertação de mestrado, Instituto de Computação, Modelagem Computacional do Conhecimento, Universidade Federal de Alagoas.
- SILVA, A. S. D. (2000), Tuta - um tutor baseado em agentes no contexto do ensino à distância, Dissertação de mestrado, Universidade Federal da Paraíba.
- SOMMERVILLE, I. (2005), *Engenharia de Software*, Addison-Wesley.
- SZYPERSKI, C. (1998), *Component Software: Beyond Object-Oriented Programming*, Addison-Wesley.