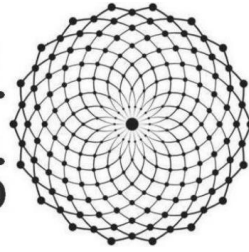


MODELAGEM
COMPUTACIONAL
DE CONHECIMENTO



DOUGLAS AFONSO TENÓRIO DE MENEZES

UM SISTEMA PORTFÓLIO-TUTOR BASEADO NO REUSO DE SOFTWARE

Maceió, Outubro de 2009

DOUGLAS AFONSO TENÓRIO DE MNEZES

UM SISTEMA PORTFÓLIO-TUTOR BASEADO NO REUSO DE SOFTWARE

Dissertação apresentada como requisito para a obtenção do grau de Mestre pelo Curso de Mestrado em Modelagem Computacional de Conhecimento da Universidade Federal de Alagoas.

Orientador: Prof. Dr. Arturo Hernández Dominguez

Maceió, Outubro de 2009

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária: Fabiana Camargo dos Santos

M543s Menezes, Douglas Afonso Tenório de.
Um sistema portfólio-tutor baseado no reuso de software / Douglas Afonso Tenório de Menezes. – 2009.
91 f. : il.

Orientador: Arturo Hemández Dominguez.
Dissertação (Mestrado em Modelagem Computacional de Conhecimento) – Universidade Federal de Alagoas. Programa de Pós-Graduação em Modelagem Computacional de Conhecimento. Maceió, 2009.

Bibliografia: f. 85-91.

1. Portfólio eletrônico. 2. Educação a distância. 3. Reuso de software. 4. Desenvolvimento de sistemas. 5. Software – Produtos. 6. Software - Componentes. I. Título.

CDU: 004.415.2:004.416.3



UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa de Pós-Graduação em Modelagem Computacional de Conhecimento
Avenida Lourival Melo Mota, Km 14, Bloco 09, Cidade Universitária
CEP 57.072-900 – Maceió – AL – Brasil
Telefone: (082) 3214-1364



Membros da Comissão Julgadora da Dissertação de Mestrado de Douglas Afonso Tenório de Menezes, intitulada: “Um Sistema Portfólio-Tutor baseado no Reuso de Software”, apresentada ao Programa de Pós-Graduação em Modelagem Computacional de Conhecimento da Universidade Federal de Alagoas em 08 de outubro de 2009, às 16h00min, na sala de aula do Mestrado em Modelagem Computacional de Conhecimento.

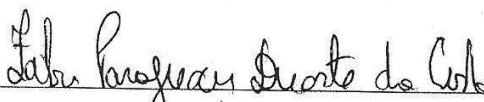
COMISSÃO JULGADORA



Prof. Dr. Arturo Hernández-Domínguez

UFAL – Instituto de Computação

Orientador



Prof. Dr. Fábio Paraguaçu Duarte da Costa

UFAL – Instituto de Computação

Examinador

Prof. Dr. Edilson Fereda

UCB – Pró-Reitoria de Pós-Graduação e Pesquisa

Examinador

Maceió, outubro de 2009.

Aos meus pais, Afonso Barros de Menezes e Josefa Tenório de Menezes, pelo incentivo,
apoio e dedicação.

“Já perdoei erros quase imperdoáveis, tentei substituir pessoas insubstituíveis e esquecer pessoas inesquecíveis.

Já fiz coisas por impulso, já me decepcionei com pessoas quando nunca pensei me decepcionar, mas também decepcionei alguém.

Já abracei pra proteger, já dei risada quando não podia, fiz amigos eternos, amei e fui amado, mas também já fui rejeitado, fui amado e não amei.

Já gritei e pulei de tanta felicidade, já vivi de amor e fiz juras eternas, "quebrei a cara muitas vezes"!

Já chorei ouvindo música e vendo fotos, já liguei só para escutar uma voz, me apaixonei por um sorriso, já pensei que fosse morrer de tanta saudade e tive medo de perder alguém especial (e acabei perdendo).

Mas vivi, e ainda vivo! Não passo pela vida... E você também não deveria passar!

Viva!

Bom mesmo é ir à luta com determinação, abraçar a vida com paixão, perder com classe e vencer com ousadia, porque o mundo pertence a quem se atreve e a vida é "muito" pra ser insignificante.”

(Charles Chaplin)

“Quem nunca errou, nunca experimentou nada novo.”

(Albert Einstein)

Agradecimentos

Algumas pessoas foram muito importantes e contribuíram de forma muito significativa para que este trabalho fosse concluído, e são a essas pessoas que expresso meus sinceros agradecimentos.

Aos meus Pais (Afonso Barros de Menezes e Josefa Tenório de Menezes) que dedicaram suas vidas para a educação dos filhos, pois tamanha dedicação e incentivo foi de fundamental importância para que eu persistisse a prosseguir nessa difícil caminhada, superando todas as dificuldades.

Ao meu filho (João Pedro Vasconcelos de Menezes), meu bem mais precioso, fonte inesgotável de todas as minhas inspirações. É por você que faço as mais longas e as mais difíceis jornadas pelo caminho da vida. Obrigado por você fazer parte da minha vida, pois sem você, ela não teria sentido algum.

Aos meus irmãos (Dani e Diogo), pelo carinho, respeito e admiração, pois eles também tiveram grande influência nessa conquista, me incentivando, me apoiando e ajudando de todas as formas possíveis. Meus sinceros agradecimentos!

À minha Tia Madalena (Madá), pois essa é uma das pessoas mais importantes na minha vida, talvez sem ela eu não pudesse ter feito uma faculdade e me tornado um profissional tão dedicado e apaixonado pela profissão. Agradeço do fundo do coração e espero sempre retribuir por tamanha dedicação. Muito obrigado!

Aos meus tios e primos que tanto me incentivaram e torceram por mim durante essa jornada.

Um agradecimento mais que especial ao Professor Arturo, meu orientador, que durante esses anos de convivência se tornou muito mais que um professor, se tornou um grande amigo a quem tenho grande admiração e respeito. Obrigado pelas conversas, pela paciência e compreensão, pelas correções de texto, pelas dicas, pelas exigências para cumprir prazos, pelas caminhadas na praia, pelo futebol de final de semana, enfim seus ensinamentos foram de suma importância para essa conquista.

Aos meus amigos do mestrado, pelas longas jornadas de estudo, pesquisa e compartilhamento de conhecimento, pois contribuíram de forma significativa para essa grande conquista, em especial à Ig Ibert, Alexandre Paes, Marcos Machado, Elba Maria, Marcílio Júnior, entre outros amigos que peço minhas sinceras desculpas pelo esquecimento injusto em não citar seus nomes.

Ao secretário do mestrado Vitor Menezes, pela amizade, pela organização e pelas informações sempre muito valiosas.

Agradeço aos professores do Mestrado, pela dedicação e empenho em todas as disciplinas, procurando sempre incentivar os alunos com auxílio de material de pesquisa, tornando os estudos muito mais prazerosos.

Finalizo agradecendo a todos àqueles que injustamente não foram citados, mas que nos bastidores fizeram com que eu conseguisse chegar ao objetivo.

Resumo

Com a grande popularização da internet, a EAD vem sendo bastante difundida pelo fato da grande facilidade de acesso, onde o aluno independente do local onde reside, poderá organizar seus horários, ampliando o alcance da modalidade de ensino presencial.

O objetivo do presente trabalho é o projeto e desenvolvimento de um Sistema Portfólio-Tutor Baseado no Reuso de Software através de técnicas de reutilização de software, objetivando o máximo de redução de trabalho por parte dos professores, melhorando consideravelmente a análise de dados e diminuindo as atividades repetitivas. Contudo, o sistema será capaz de integrar todas as partes que podem ser envolvidas no processo de aprendizagem.

Um portfólio é uma pasta onde ficam armazenadas as evidências da habilidade de cada aluno, sendo possível um acompanhamento cronológico de seu desempenho. Um portfólio eletrônico pode ser dividido em portfólio do aluno e portfólio do professor. É possível desenvolver atualmente dois tipos de portfólios de alunos: o Portfólio de Trabalho, onde são armazenados os trabalhos dos alunos e o Portfólio de Apresentação, onde é armazenado o melhor que o aluno produziu.

Palavras-Chave: Portfólio Eletrônico, Educação a Distância, Reuso de Software, Linha de Produtos de Software, Componentes de Software.

Abstract

With the great popularity of the Internet, the EAD has been widespread because of the great ease of access, where the student regardless of where they reside, can organize their schedules, broadening the scope of the method of teaching presence.

The objective of this work is the design and development of a System Portfolio- Tutor Based in the reuse of software through techniques of software reuse, to the maximum reduction of work by teachers, significantly improving data analysis and reducing the repetitive activities. However, the system will be able to integrate all the parts that may be involved in the learning process.

A portfolio is a folder where are stored the evidence of the ability of each student, with a possible time monitoring of their performance. A portfolio can be divided into electronic portfolio of student and teacher's portfolio. It is now possible to develop two types of portfolios of students: the Portfolio of Work, which are stored the work of students and Presentation of Portfolio, which is stored the best that the student produced.

Keywords: Electronic Portfolio, Distance Education, Reuse of Software, Product Software Line, Software Components.

Lista de Figuras

2.1	Representação UML de um Componente de Software -----	6
2.2	Demonstração gráfica do Framework -----	12
2.3	Processo de Desenvolvimento de uma Linha de Produto de Software -----	18
3.1	Portfólio Tradicional-----	22
3.2	Arquitetura do Portfólio Eletrônico para Web -----	26
3.3	Camadas do Portfólio-Tutor -----	27
3.4	Arquitetura Clássica de Um STI -----	28
3.5	Arquitetura do Portfólio-Tutor -----	29
3.6	Arquitetura do Sistema Portfólio Eletrônico POETA-----	30
3.7	Diagrama de Contexto do Sistema POETA -----	31
3.8	Tela de Login do Sistema PASS-PORT -----	35
3.9	Tela Principal do Sistema PASS-PORT -----	36
3.10	Tela de Login do Sistema Epsilen-----	38
3.11	Registrando o Portfólio -----	39
3.12	Portfólio Público do Professor -----	39
4.1	Modelo de um Sistema Portfólio-Tutor -----	43
4.2	Diagrama de Caso de Uso do Administrador -----	47
4.3	Diagrama de Caso de Uso do Professor-----	48
4.4	Diagrama de Caso de Uso do Aluno -----	49
4.5	Diagrama de Caso de Uso do Monitor -----	50
4.6	Arquitetura do Sistema Portfólio-Tutor Baseado em Componentes -----	51
4.7	Diagrama de Componentes -----	53
4.8	Diagrama de Classe do Componente Usuário-----	54
4.9	Diagrama de Classe do Componente Atividades -----	55
4.10	Diagrama de Classe do Componente Disciplina-----	56
4.11	Diagrama de Classe do Componente Avaliação -----	56
5.1	Disposição Física em 3 Camadas -----	59
5.2	Tela Inicial do Sistema Proposto-----	60
5.3	Tela Principal do Sistema -----	60

5.4	Tela de Para Mudança de Senha -----	61
5.5	Cadastro de Usuários (Informações Básicas dos Usuários) -----	61
5.6	Cadastro de Usuários (Perfil dos Usuários) -----	62
5.7	Cadastro de Usuários (Log de Acesso)-----	63
5.8	Perfil de Usuário-----	63
5.9	Cadastro de Usuários -----	64
5.10	Tela de Permissão de Acesso (Aba Ações)-----	64
5.11	Tela de Permissão de Acesso (Aba Controles)-----	65
5.12	Tela de Controle de Professores-----	66
5.13	Cadastro de Professores (Dados Pessoais)-----	66
5.14	Cadastro de Professores (Disciplinas Lecionadas) -----	67
5.15	Lista de Disciplinas Cadastradas no Sistema-----	67
5.16	Tela de Controle de Alunos-----	68
5.17	Cadastro de Alunos -----	68
5.18	Cadastro de Turmas-----	69
5.19	Tela Inicial Para o Usuário com Perfil de Professor -----	70
5.20	Tela de Controle de Atividades -----	71
5.21	Tela de Cadastro de Atividades -----	71
5.22	Tela de Cadastro de Táticas-----	72
5.23	Tela de Controle das Estratégias Didáticas-----	73
5.24	Tela de Cadastro das Estratégias Didáticas-----	74
5.25	Tela de Controle das Sessões de Ensino -----	75
5.26	Tela de Cadastro das Sessões de Ensino -----	75
5.27	Tela Inicial Para o Usuário com Perfil de Aluno -----	76
5.28	Tela de Visualização de Todas as Telas Disponíveis -----	77
5.29	Tela de Visualização de uma Atividade Específica -----	78
5.30	Tela de Controle das Atividades Finalizadas-----	79
5.31	Tela de Finalização de Uma Atividade-----	80
5.32	Tela de Aviso -----	80
5.33	Tela de Visualização de Avisos -----	81
5.34	Tática de Ensino Utilizando Debate Síncrono -----	82

Sumário

1	INTRODUÇÃO	1
1.1	Contextualização da Dissertação	1
1.2	Objetivos da Dissertação	2
1.3	Organização da Dissertação	3
2	TÉCNICAS DE REUTILIZAÇÃO DE SOFTWARE	4
2.1	Reutilização de Software	4
2.2	Desenvolvimento Baseado em Componentes	6
2.2.1	Engenharia de Domínio	8
2.2.1.1	Etapas da Engenharia de Domínio	9
2.2.1.2	Fontes de Informação	9
2.2.1.3	Produtos Gerados pela Engenharia de Software	10
2.2.2	Processo de Software Baseado em Componentes	11
2.3	Framework	11
2.3.1	Desenvolvimento de Frameworks	13
2.3.2	Classificação de Frameworks	15
2.4	Linha de Produto de Software	15
2.4.1	Elementos de Uma Linha de Produto de Software	17
2.4.1.1	Variabilidade	18
2.4.1.2	Desenvolvimento de Núcleo de Artefatos	19
2.4.1.3	Desenvolvimento do Produto	20
2.5	Resumo	21
3	PORTFÓLIO ELETRÔNICO	22
3.1	Definição	22
3.2	Tipos de Portfólios Eletrônicos	24
3.2.1	Portfólio do Aluno	24
3.2.2	Portfólio do Professor	25
3.3	Portfólios Eletrônicos Existentes	25
3.3.1	Portfólio Eletrônico para Web	26
3.3.2	Portfólio – Tutor	27
3.3.3	POETA	30

3.3.4	Grady Profile (Auerbach & Associates) -----	32
3.3.5	Connectcut College e-Portfolio -----	33
3.3.6	PASS-PORT -----	33
3.3.7	Epsilen -----	36
3.3.8	Comparação entre os portfólios analisados -----	40
3.4	Resumo -----	41
4	MODELAGEM DO SISTEMA PORTFÓLIO – TUTOR -----	42
4.1	Modelo do Sistema Portfólio-Tutor -----	42
4.2	Requisitos -----	45
4.3	Casos de Uso Essenciais -----	46
4.3.1	Diagrama de Caso de Uso para Administrador -----	46
4.3.2	Diagrama de Caso de Uso para Professor -----	47
4.3.3	Diagrama de Caso de Uso para Aluno -----	48
4.3.4	Diagrama de Caso de Uso para Monitor -----	49
4.4	Arquitetura do Modelo do Sistema Portfólio-Tutor Proposto -----	50
4.5	Como Funciona uma Aplicação Portfólio-Tutor -----	52
4.5.1	Diagrama de Componentes -----	52
4.6	Especificação dos Componentes -----	53
4.6.1	Diagrama de Classe do Componente Usuário -----	53
4.6.2	Diagrama de Classe do Componente Atividades -----	54
4.6.3	Diagrama de Classe do Componente Disciplina -----	55
4.6.4	Diagrama de Classe do Componente Avaliação -----	56
4.7	Resumo -----	57
5	IMPLEMENTAÇÃO DO SISTEMA -----	58
5.1	Tecnologias Utilizadas na Implementação -----	58
5.2	Disposição Física dos Módulos do Sistema -----	58
5.3	Estudo de Caso -----	59
5.3.1	Interface do Sistema Portfólio-Tutor Baseado no Reuso de Software ----	59
5.3.1.1	Interface do Administrador -----	61
5.3.1.2	Interface do Professor -----	70
5.3.1.2.1	Cadastro de Atividades -----	71
5.3.1.2.2	Cadastro de Táticas -----	72

5.3.1.2.3	Cadastro de Estratégias Didáticas -----	73
5.3.1.2.4	Cadastro de Sessão de Ensino -----	74
5.3.1.3	Interface do Aluno -----	76
5.3.1.3.1	Visualização das Atividades -----	77
5.3.1.3.2	Entregar Atividades (Atividades Finalizadas)-----	78
5.3.1.3.3	Visualização de Avisos -----	80
5.4	Execução de uma Sessão de Ensino -----	81
5.5	Resumo -----	82
6	CONSIDERAÇÕES FINAIS -----	83
6.1	Resultados Obtidos -----	83
6.2	Trabalhos Futuros -----	84
	REFERÊNCIAS BIBLIOGRÁFICAS -----	85

1 INTRODUÇÃO

O grande problema para os professores que desejam gerenciar de forma tradicional um portfólio através de pastas em sala de aula é principalmente o armazenamento e gerenciamento do material existente, pois seria necessário um grande espaço físico para armazenamento de uma gama de informação, variando entre papeis, vídeos, projetos, CDs, entre outros, impossibilitando a análise e o manuseio das informações com rapidez. Para solucionar esse problema, seria necessário o desenvolvimento de um sistema portfólio eletrônico, onde as informações seriam armazenadas de forma digital.

O desenvolvimento de sistemas portfólio eletrônico sem uma abordagem orientada a reuso apresenta-se demorada, com duplicação de funcionalidades e código associado.

Com o uso de técnicas de reuso é possível obter produtos de forma mais econômica em um curto prazo de tempo.

O custo de produção para até três produtos de software no mesmo domínio, e que não utilizam técnicas de Reuso de Software inicialmente será menor, mas quando a exigência é o desenvolvimento de produtos em quantidade, o custo será consideravelmente inferior com a utilização da técnica de Reuso de Software para o desenvolvimento do sistema (Aragon, 2004). O ponto de retorno é o momento em que o projeto utilizando a técnica de Reutilização de Software, tenha um retorno financeiro diferencial em relação ao projeto que não utiliza a mesma técnica.

Diante do exposto é necessário utilizar um Sistema Portfólio-Tutor Baseado no Reuso de Software, pois esta abordagem de desenvolvimento justifica-se, já que ela permite uma sistematização no reuso de software do núcleo comum de componentes e também permite levar em conta as particularidades específicas para cada produto de software. Com uma arquitetura baseada no sistema Portfólio-Tutor será possível a criação de portfólios eletrônicos específicos de forma rápida e de menor custo.

1.1 Contextualização da Dissertação

O Mestrado em Modelagem Computacional de Conhecimento da Universidade Federal de Alagoas está dividido em três linhas de pesquisa: Descoberta de Conhecimento e Otimização

de Decisões, Modelagem Computacional em Educação e Modelos Quantitativos e de Simulação, onde se organiza em uma única área de concentração, que é a Modelagem Computacional de Conhecimento.

O presente trabalho está voltado à linha de pesquisa em Modelagem Computacional em Educação, com o foco em portfólios eletrônicos.

No presente trabalho será apresentado um Sistema Portfólio-Tutor Baseado no Reuso de Software, onde o professor terá apoio de suas atividades e ajuda em relação ao processo de ensino, onde será possível um acompanhamento mais detalhado de todas as suas atividades disponibilizadas aos seus alunos e conseqüentemente acompanhando a evolução da aprendizagem dos mesmos com base nas estratégias didáticas. Essas funcionalidades serão fornecidas pelo Portfólio Eletrônico controlando as atividades e a realização de sessões de ensino on-line.

O Portfólio Eletrônico é uma ferramenta, que possibilita mostrar o progresso de aprendizado, acumulando registros da vida acadêmica de um determinado aluno. Ele representa uma coleção de informações do progresso de um aluno, e é um canal de comunicação entre o aluno e o professor (Silva, 2002). Com o portfólio eletrônico, é possível obter o registro histórico de artefatos associados à aprendizagem dos alunos.

1.2 Objetivos da Dissertação

O objetivo principal dessa dissertação é a construção de um modelo de Sistema Portfólio-Tutor Baseado no Reuso de Software, através do desenvolvimento baseado em componentes, objetivando auxiliar o professor no acompanhamento do progresso dos alunos no processo de aprendizagem, melhorando consideravelmente o gerenciamento das atividades envolvidas no processo de aprendizagem.

1.3 Organização da Dissertação

Esta dissertação foi dividida em seis capítulos, que foram organizados da seguinte maneira:

Capítulo 2 – Técnicas de Reuso de Software: No segundo capítulo são apresentadas algumas técnicas de reuso de software. Este capítulo está dividido em cinco categorias: Reuso de Software, Desenvolvimento Baseado em Componentes, Padrões de Projeto, Frameworks e Linha de Produto de Software.

Capítulo 3 – Portfólio Eletrônico: No terceiro capítulo é apresentado o conceito de portfólio eletrônico e seus tipos. Também são mostrados alguns sistemas portfólios eletrônicos existentes, detalhando suas arquiteturas e funcionamento.

Capítulo 4 – Modelagem do Sistema Portfólio - Tutor Baseado no Reuso de Software: No quarto capítulo é detalhada a modelagem do sistema proposto, detalhando a arquitetura e os componentes do mesmo.

Capítulo 5 – Implementação do Sistema Portfólio - Tutor Baseado no Reuso de Software: No quinto capítulo mostra-se a interface e funcionamento do sistema.

Capítulo 6 – Considerações finais: Neste capítulo são realizadas as considerações finais da dissertação. Serão apresentadas também as contribuições desse trabalho e trabalhos futuros que poderão ser desenvolvidas a partir dessa dissertação.

2 TÉCNICAS DE REUTILIZAÇÃO DE SOFTWARE

No capítulo 2 serão abordados os conceitos de Reuso de Software utilizados no desenvolvimento do sistema proposto. Na Seção 2.1 serão apresentados os conceitos de reutilização de software, suas vantagens e desvantagens. Na seção 2.2 serão mostradas as vantagens da utilização do processo baseado em componentes. Na Seção 2.3 Serão apresentados os conceitos de frameworks e seus tipos. Finalizando com a seção 2.4, onde serão abordados os conceitos de Linha de Produto de Software.

2.1 Reutilização de Software

Nos projetos de engenharia, a base principal é o reuso de componentes. O projeto é baseado em componentes que já foram testados em outros sistemas. O reuso de software é primordial para aumentar e agilizar o retorno de investimento feito para o desenvolvimento do produto. Através do reuso generalizado, é possível conseguir menores custos de produção e manutenção de software, maior rapidez na entrega e aumento na qualidade dos produtos (Sommerville, 2007).

Softwares que utilizam reusabilidade podem aparecer de diversos tamanhos:

- **Reuso de Sistemas de Aplicações** – que pode ser utilizado pelo desenvolvimento de famílias de aplicações, também conhecida como Linha de Produto, que tem como principal característica um conjunto de aplicações relacionadas, que tem uma arquitetura de domínio específica em comum.
- **Reuso de Componentes** – com o reuso de componentes previamente desenvolvidos é possível reduzir o tempo de desenvolvimento do software e conseqüentemente ter maior confiabilidade, pois o componente que está sendo reutilizado já foi testado anteriormente em outras aplicações.
- **Reuso de Funções** – todo componente que implementa uma única função, pode ser reutilizado.

As técnicas de reutilização de sistemas já vêm sendo usadas há muitos anos, desde a implementação dos sistemas até ajustes para diferentes ambientes. A maior vantagem na utilização da técnica de reuso de software, é que os custos de desenvolvimento ficam

consideravelmente reduzidos. Essa redução é apenas uma vantagem na utilização do reuso, porém existem várias outras vantagens na utilização dessa técnica mostradas a seguir:

- ***Maior Confiabilidade*** – os sistemas que usam componentes reutilizados têm maior confiabilidade do que os sistemas que usam novos componentes, pois esses componentes já foram testados em diferentes ambientes. Como esses componentes já foram testados, as chances de descoberta de defeitos são minimizadas, reduzindo assim o número de falhas na sua reutilização.
- ***Redução dos Riscos de Processo*** – com a utilização de componentes reutilizados, as dúvidas a respeito do investimento para com estes componentes é bem menor que o investimento para o desenvolvimento. É um fator importante, pois reduz as incertezas nas estimativas de custos de projeto.
- ***Uso Efetivo de Especialistas*** – os especialistas desenvolvem componentes reutilizáveis, que envolve seu conhecimento.
- ***Conformidade com Padrões*** – com a utilização de padrões, a probabilidade de erros cometidos pelos usuários, é consideravelmente diminuída, tomando como exemplo a utilização de componentes reutilizáveis para a implementação de menus em uma interface com o usuário.
- ***Desenvolvimento Acelerado*** – o reuso de componentes acelera a produção, pois o tempo de desenvolvimento deve ser reduzido.

Também é possível destacar algumas desvantagens na utilização do reuso de sistemas:

- ***Aumento nos Custos de Manutenção*** – com a indisponibilidade do código-fonte do componente, os custos de manutenção poderão aumentar, tornando-se incompatível com as mudanças do sistema.
- ***Falta de Ferramentas de Apoio*** – a interação de ferramentas CASE com um sistema de biblioteca de componentes, pode se tornar difícil ou impossível.
- ***Manutenção de Uma Biblioteca de Componentes*** – torna-se complicado assegurar que desenvolvedores de software utilizem uma biblioteca de componentes.

Três requisitos são fundamentais para o projeto e desenvolvimento de software baseado em reuso:

- Possibilidade de encontrar componentes reutilizáveis. É necessária uma base de componentes reutilizáveis bem documentados e componentes de fácil localização.
- Com a reutilização dos componentes é necessário que sejam confiáveis e que se comportem adequadamente.
- A documentação é necessária para ajudar o usuário a compreender e adaptar. Essa documentação deve informar onde os componentes foram reutilizados e sobre qualquer problema que venha a ser encontrado.

Existem custos e problemas com relação ao reuso, podendo dificultar a utilização desse método e diminuir as reduções de custo total de desenvolvimento (Sommerville, 2007). Por isso, o uso sistemático deve ser planejado e introduzido por um programa de reuso usado por toda a organização.

2.2 Desenvolvimento Baseado em Componentes

Um componente de software é uma unidade de composição com interfaces especificadas contratualmente e somente dependências explícitas de contexto. Através de uma interface é identificado um ponto de interação entre um componente e seu ambiente, onde o mesmo poderá ser utilizado por outros componentes de software. Existem dois tipos para a interface do componente: uma que identifica um ponto de acesso a serviços onde o componente é capaz de fornecer para o ambiente e a outra que identifica um ponto de acesso a serviços que o componente requer do ambiente. Na Figura 2.1 podemos ver a representação de um componente, usando a notação UML.

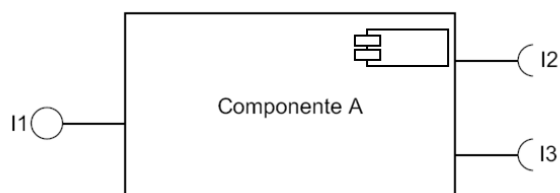


Figura 2.1: Representação UML de um Componente de Software

A mais de 30 anos é utilizado o conceito de desenvolvimento de software a partir de composição de artefatos de software. Durante a “crise do software”, no período de 1968,

McIlroy falou da necessidade das empresas produzirem componentes reutilizáveis. Em seu ponto de vista, os desenvolvedores escolheriam seus componentes e ajustariam às suas necessidades (Sommerville, 2007).

O desenvolvimento baseado em componentes surgiu no final da década de 90 como um estudo baseado no reuso para o desenvolvimento de software. A maior motivação para o desenvolvimento baseado em componentes foi o fracasso do desenvolvimento orientado a objetos em relação à extensa utilização da reusabilidade, como foi proposto originalmente. As previsões para o mercado eram boas, mas era necessário o conhecimento detalhado das classes para a sua utilização, impossibilitando o desenvolvimento de um mercado significativo para os objetos individuais (Silva & Price, 1999).

O nível de abstração dos componentes é maior do que as classes de objeto, podendo ser considerados provedores de serviços stand-alone.

Duas características importantes são enfatizadas em um componente reutilizável, com a visualização como um provedor de serviços (Oliveira Júnior & Gimenez, 2003):

- O componente é totalmente independente em sua execução. A sua compilação é individual, pois o código-fonte não está disponível.
- As interações são feitas através de interfaces publicadas pelos componentes. Onde o estado interno da interface nunca é exposto.

Em muitos casos, os componentes são imaginados como duas interfaces relacionadas, a interface *provides*, que define os serviços fornecidos pelo componente e a interface *requires*, que especifica quais serviços devem estar disponíveis. Essas interfaces devem ser fornecidas, para o funcionamento do componente (Gimenes, 2004).

Existem vários níveis de abstração em componente, desde uma simples rotina até uma aplicação completa. Meyer (1999) identificou cinco níveis de abstração:

- **Abstração funcional** – uma única implementação feita pelo componente.
- **Agrupamentos causais** – o componente é uma coleção de entidades que estão relacionadas de forma adequada.
- **Abstração de dados** – em uma linguagem orientada a objetos a abstração de dados ou classe é representada pelo componente.
- **Abstração em clusters** – o componente é o grupo de classes relacionadas que trabalham em conjunto. Às vezes, chamados de *Framework*.

- **Abstração de sistemas** – o componente é um sistema inteiramente auto contido. Reutilizar abstrações de nível de sistema é, às vezes, chamado de reuso de produtos COTS.

Com o constante crescimento do tamanho e da complexidade dos sistemas de software, o interesse em desenvolver sistemas baseado em COTS aumenta. A principal vantagem é a diminuição de custos e tempo de desenvolvimento.

2.2.1 Engenharia de Domínio

Para que exista uma reutilização de código, implicitamente é envolvida a reutilização de análise e projeto. É recomendável que não seja feita a reutilização de uma informação em um nível, sem que esta seja precedida da reutilização da informação correspondente no nível mais abstrato, não permitindo assim a explicitação dos caminhos seguidos desde a especificação do produto até a sua implementação (Silva & Price, 1999).

Geralmente as informações não estão organizadas de forma coerente e de fácil utilização, dificultando a reutilização de informações das etapas de análise e projeto. Alguns problemas identificados afetam de forma considerável as etapas de desenvolvimento. Com esses problemas é necessária a criação de métodos para a organização coerente do domínio da aplicação. Dentre os principais problemas destacam-se (Werner & Braga, 2000):

- **Desconhecimento do domínio da aplicação** – trabalhando-se com uma nova área de aplicação, tal área torna-se totalmente desconhecida. É necessário um processo de aprendizagem, que conseqüentemente requer tempo e técnicas para esta aprendizagem;
- **Vocabulário conflitante** – muitas vezes, grandes sistemas de software utilizam-se das mais diversas áreas de aplicação, sendo necessária a cooperação de diversos profissionais, das mais diversas áreas para fazer uma integração, podendo ter vocabulários distintos ou muitas vezes conflitantes;
- **Incerteza nas decisões** – a incerteza acontece, quando um profissional está trabalhando em uma área que não é de sua especialidade;
- **Conhecimento errôneo do domínio da aplicação** – algumas vezes os desenvolvedores recebem informações incorretas do funcionamento de determinados componentes, sendo necessário o entendimento das informações e possam compará-las.

Com todos esses problemas a qualidade do software desenvolvido fica ameaçada. Com a análise de domínio, torna-se possível obter informações para aumentar a reutilização no processo de desenvolvimento, podendo utilizar, durante a produção de novas aplicações o uso de componentes reutilizáveis.

Uma análise de domínio identifica os conceitos e funcionalidades requeridos por uma família de sistemas representados através de um modelo genérico (Werner & Braga, 2000).

2.2.1.1 Etapas da Engenharia de Domínio

Baseado nos métodos de Engenharia de Domínio existente na literatura, três etapas são identificadas:

- **Análise do domínio** – são identificados e generalizados os padrões e procedimentos, operações e objetos e as características de sistemas similares, onde a partir disso, será definido um modelo que servirá como uma fonte unificada de definições para referências. Alguns benefícios como a criação de um vocabulário comum entre os desenvolvedores, melhor planejamento e redução de custos são destacados.
- **Projeto do domínio** – nesta etapa serão utilizados os artefatos que foram criados para construir uma arquitetura de software que implemente soluções para os requisitos comuns do domínio.
- **Implementação de domínio** – Transforma as oportunidades de reutilização e soluções do projeto para um modelo que possa ser implementado, e que inclui serviços tais como: a identificação, reengenharia e/ou construção, e manutenção de componentes reutilizáveis que suportem estes requisitos e soluções de projeto (Werner & Braga, 2000).

2.2.1.2 Fontes de Informação

São disponibilizadas diversas fontes de informações para a engenharia de domínio, apresentando diversas vantagens e desvantagens.

- **Livro** – a principal vantagem é a fonte segura de conhecimento teórico do domínio, e tem como desvantagem a apresentação de uma visão ideal do domínio.
- **Padronizações** – tem como vantagem a constante utilização de uma nomenclatura amplamente aceita no domínio. Tem como desvantagem a validação para verificação de sua aceitação no domínio.
- **Aplicações existentes** – esta fonte apresenta diversas vantagens, tais como: a modelagem, se existir, é bem realista, pode ser diretamente utilizada para se determinar os requisitos do usuário, a arquitetura pode ser abstraída para a arquitetura do domínio, se houver certo consenso entre as aplicações podendo permitir a generalização e empacotamento de componentes que possam ser implementados. A desvantagem é, que se o domínio é dinâmico, as aplicações podem não apresentar mais a realidade do domínio (Oliveira Júnior, 2002).
- **Especialistas do domínio** – a principal vantagem é a permissão da captura mais consistente da dinâmica e da evolução do domínio. E tem como desvantagem, o poder de refletir uma visão muito particular do domínio (Werner & Braga, 2000).

2.2.1.3 Produtos Gerados Pela Engenharia de Domínio

A documentação dos resultados provém de representações específicas para cada uma de suas fases. O escopo ou abrangência do domínio é prioridade para estas representações, descrevendo problemas passíveis de serem resolvidos e prover implementações que solucionem os problemas encontrados no domínio.

As representações disponibilizadas capturam o contexto e a abrangência do domínio, através da análise do domínio. O modelo deve acompanhar uma documentação detalhada de cada uma delas, mostrando a possibilidade de mapeamento em requisitos de aplicações.

2.2.2 Processo de Software Baseado em Componentes

O desenvolvimento de software baseado em componentes tem como objetivo a definição de componentes interoperáveis, com interfaces bem definidas, evidenciando os tipos de relacionamentos permitidos por este componente. Desta forma, a complexidade no desenvolvimento é reduzida, assim como os custos, através de reutilização de componentes exaustivamente testados.

Para esta técnica de reutilização de software ser efetiva, ela deve reduzir a *distância cognitiva*¹ entre o conceito inicial de um sistema e sua implementação final. Sendo assim, no contexto de um processo de desenvolvimento de software baseado em componentes, a aplicação da reutilização das fases iniciais do desenvolvimento deve ter como resultado a reutilização de componentes na implementação da aplicação. Assim, deve haver uma relação estreita entre os conceitos de um dado domínio de aplicação considerados reutilizáveis nas fases iniciais do processo de desenvolvimento e os componentes codificados que serão utilizados na implementação da aplicação.

Um processo de software eficiente incrementa a produtividade de desenvolvimento por várias razões: auxilia a entender os fundamentos de como o software é desenvolvido favorecendo a tomada de decisões e a escolha de ferramentas. Além disso, auxilia na padronização, promovendo reuso e consistência entre as equipes de projeto e favorecem oportunidades para introduzir melhores práticas como inspeção de código, gerenciamento de configuração do desenvolvimento.

A construção de software por meio de componentes leva a uma maior reutilização e conseqüentemente aumenta a qualidade e produtividade, o que enfatiza a crença que esta nova modalidade de desenvolvimento de software venha a crescer cada vez mais.

2.3 Frameworks

¹ É o conjunto de processos mentais usados no pensamento e na percepção, também na classificação, reconhecimento e compreensão para o julgamento através do raciocínio para o aprendizado de determinados sistemas e soluções de problemas (Godoy 2006).

Apesar dos avanços das técnicas de desenvolvimento de software, a construção de software continua a ser um processo extremamente complexo. De acordo com (Weber, Rocha & Nascimento, 2001), os produtos de software têm, em geral, seus cronogramas atrasados, custo maior do que o esperado e apresentam defeitos. Isto resulta em uma série de inconveniências para os usuários e traz consigo uma enorme perda de tempo e recursos.

Uma das abordagens para diminuir a complexidade e aumentar a produtividade e a qualidade, tem sido a reutilização de software. Entretanto, esta não é uma tarefa fácil. A crescente heterogeneidade de arquiteturas de hardware e software somados à diversidade de sistemas operacionais e plataformas de comunicação dificulta a reutilização de projetos. Podem-se citar também aspectos culturais e de formação que dificultam a reutilização.

Pouco se aproveita de sistemas de informação já existentes quando se precisa construir um novo sistema.

Uma das propostas de reutilização bastante aceita na área de orientação a objetos é de se usar *framework*. Um framework é um conjunto de classes integradas que define uma estrutura reutilizável para um domínio específico de aplicações. O framework provê uma arquitetura particionada em classes abstratas e define as responsabilidades e um modelo de colaboração para essas classes. O desenvolvedor adapta o framework para uma aplicação particular especializando e agregando instâncias de suas classes. A figura 2.2 ilustra uma aplicação a partir de um framework. A estrutura mais escura corresponde ao framework e o restante é o que foi produzido pelo usuário do framework no desenvolvimento dessa aplicação específica (Knabben & Robert, 2002).

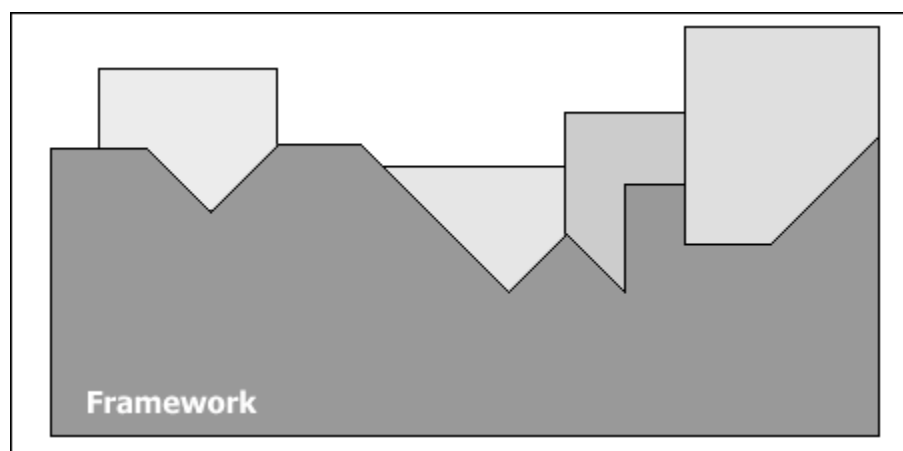


Figura 2.2: Demonstração gráfica do framework

Frameworks portam a infra-estrutura de projeto, característica que reduz a quantidade de código a ser desenvolvida na criação de aplicações. O modelo de colaboração entre as classes de um framework define a arquitetura da aplicação livrando o desenvolvedor dessa responsabilidade.

A arquitetura dinâmica de um framework é caracterizada por uma inversão de controle. A inversão de controle permite que o framework (e não a aplicação) determine que métodos invocar em resposta a determinados eventos (Knabben & Robert, 2002).

2.3.1 Desenvolvimento de Frameworks

O desenvolvimento de um framework é ligeiramente diferente do desenvolvimento de uma aplicação comum. A grande diferença é que um framework tem que cobrir os conceitos relevantes a um domínio, enquanto uma aplicação cobre apenas os conceitos mencionados nos seus requisitos. O desenvolvimento de um framework pode ser dividido da seguinte maneira:

Análise do Domínio: Nesse processo são identificados e organizados os conhecimentos a respeito de uma classe de problemas (Domínio de Aplicações) para suportar a descrição e solução desses problemas. É um passo fundamental na criação de artefatos de softwares reutilizáveis, pois elementos gerados através de uma análise de domínio capturam a funcionalidade essencial requerida por um domínio.

Uma análise de domínio é um passo importante no desenvolvimento de frameworks, pois faz com que o desenvolvedor do framework obtenha uma maior compreensão dos conceitos do domínio, funcionalidade e *hot spots*² que um framework deve possuir.

Modelagem: consiste na especificação da estrutura de classes desse framework. Essa estrutura de classes deve possuir algumas características importantes:

- *Flexibilidade* – É a capacidade do mesmo se alterar em função da aplicação específica do domínio do framework, que difere de outras aplicações em alguns aspectos. Através de hot spots pode-se modelar a flexibilidade com que o framework trabalhará.
- *Extensibilidade* – onde se refere a manutenção do framework. O desenvolvimento de frameworks é um processo iterativo, isto é, à medida que o framework é utilizado novos

² Um aspecto variável de uma aplicação é chamado de “Ponto de Adaptação” (em inglês, “Hot Spot”).

recursos podem ser agregados a sua estrutura. Assim, durante o projeto de um framework, deve-se tentar prever futuras utilizações para este framework e futuras extensões no domínio tratado.

Implementação: a implementação de frameworks segue as linhas gerais da implementação de uma aplicação comum. Todas as técnicas e padrões para o desenvolvimento de código podem ser usados na implementação de um framework.

É importante ressaltar que o sucesso da implementação de um framework depende da qualidade do projeto desse framework. Todas as características definidas durante a fase de projeto devem ser mantidas na fase de implementação para que o framework não perca sua capacidade de generalizar um domínio de aplicações.

Testes: um framework é validado através da criação de aplicações teste, usadas para determinar se o framework provê as funcionalidades desejadas e para avaliar sua usabilidade. Se, no processo de criação de aplicações, forem encontradas características do domínio tratado que não estão presentes no framework deve-se reavaliar o projeto e atualizar a implementação do framework para que essas características sejam adicionadas. Essa retro-alimentação faz parte do ciclo de vida dos frameworks.

Documentação: o desenvolvimento e a utilização de frameworks são tarefas complexas e, por isso, uma boa documentação é essencial. A documentação deve prover informações sobre o domínio tratado pelo framework, sua estrutura e funcionamento.

Técnicas padronizadas de análise e projeto orientados a objetos – técnicas padronizadas como a UML modelam visões estáticas (diagramas de classes e objetos, etc.) e visões dinâmicas (diagramas de estado, atividades dentre outros), porém não representam de forma clara os hot spots (variabilidade de um método ou extensibilidade de uma classe) nem como os mesmos podem ser instalados. Assim sendo, a diferença entre o núcleo de um framework (frozen spots) e as partes variantes (hot spots) não é explicada nas técnicas padronizadas de análise o projeto OO.

Visando contornar essas limitações, propôs a UML-F, uma extensão da UML, para modelar framework. A UML-F pode ser considerada uma evolução em relação às abordagens anteriores, pois representa e classifica os hot spots de frameworks de uma forma mais clara e direta (Fontoura, Pree & Rumpe, 2000).

Outro modo de documentar um framework, talvez o mais elementar, é a disponibilização de código fonte aos usuários. O código fonte de um framework ou de

aplicações desenvolvidas sob esse framework é uma rica fonte de documentação. Entretanto, é muito difícil entender o funcionamento de um framework apenas pelo seu código fonte e, por isso, é recomendável que o código não seja a única fonte de referência disponibilizada pelo autor do framework.

2.3.2 Classificação dos Frameworks

Os frameworks podem ser classificados de acordo como fornecem seus hot spots, ou seja, a forma como ele pode ser estendido para gerar uma nova aplicação. A primeira forma de classificação são os frameworks de herança ou “caixas brancas” que contam com herança e ligação dinâmica para conseguir a sua extensibilidade (Johnson & Foote 2001). Em contraste, existem os frameworks de interface fixa ou “caixas pretas”, que são estruturados usando a composição e a delegação de objetos ao invés de herança. Existem ainda um terceiro tipo de framework, que é um intermediário entre os outros dois anteriores, os frameworks desse tipo são chamados “caixas cinza”. Estes possuem flexibilidade e extensibilidade suficientes dos frameworks “caixa branca” e consegue esconder informações desnecessárias dos desenvolvedores de aplicação, característica dos frameworks “caixas preta” (Fayad, 1999).

Em geral, uma maior flexibilidade implica num maior esforço de programação por parte dos desenvolvedores que fazem uso dos frameworks. Por outro lado implica num menor esforço para os criadores desse framework, pois não há necessidade de previsão de todas as alternativas de implementações possíveis, ou seja, definição completa de suas interfaces fixas. Quanto mais os frameworks forem constituídos de interfaces fixas, maior será a redução de seu nível de flexibilidade e sua reutilização se tornará mais fácil (Johnson & Foote 2001).

2.4 Linha de Produto de Software

Com a reutilização de soluções, o processo de software tem sua produtividade e qualidade consideravelmente melhorada. Aprimoramento nas mais diversas áreas, tais como, arquitetura de software, desenvolvimento baseado em componentes, engenharia de domínio,

padrões de projeto e framework, estão sendo aplicados e estudados, para uma melhor aplicação na produção de software.

Uma linha de produto de software envolve um conjunto de aplicações similares dentro de um domínio que podem ser desenvolvidas a partir de uma arquitetura genérica comum, a arquitetura de linha de produto, e um conjunto de componentes que povoam a arquitetura (Lazilha, 2002).

O objetivo é a identificação dos aspectos comuns e as diferenças entre os artefatos de software no processo de desenvolvimento da linha de produto. Esta abordagem é aplicada nos domínios, em que existe a procura por produtos específicos, existindo pontos de variabilidade totalmente definidos e características em comum (Gimenes, Lazilha & Halmeman, 2000).

É necessária a identificação de pontos de variabilidade, importantes para diferenciar as características dos produtos. Geralmente, os pontos de variabilidade aparecem na definição dos requisitos da linha de produto, e obrigatoriamente, tendo que ser representado em todo o processo de desenvolvimento.

Existe muita confusão no enfoque de linha de produto, se considerarmos algumas técnicas que foram aplicadas anteriormente e que tinham objetivos parecidos, destacando-se: engenharia de domínio e framework (Lazilha, 2002).

A linha de produto e as técnicas de engenharia de domínio e frameworks, são complementares. A linha de produto de software serve como uma forma de organizar essas técnicas, que tem como maior exemplo a Celsius Tech, uma empresa que conseguiu um grande sucesso, aumentando consideravelmente a demanda de seus produtos, vindo assim a necessidade de encontrar uma forma de gerenciar sua família de produtos. Poulin (1997) afirma que linha de produto e domínios representa o mesmo nível de abstração e que o termo linha de produto foi criado para facilitar a comunicação entre o pessoal técnico e os gerentes que pensam em termos de produtos.

Frameworks e Padrões de Projetos também são recursos importantes para o estudo de linha de produto (Lazilha 2002). Para se trabalhar com linha de produto, as aplicações devem possuir uma generalização que permita instanciações para aplicações específicas. O framework é um bom exemplo para esse tipo de aplicação, pois ele pode ser classificado como uma arquitetura de software semi-definida, que fornece um esqueleto de aplicações e interconexões entre elas, formando uma pré-fabricação para o desenvolvimento de aplicações em um domínio específico.

A abordagem de linha de produto de software é um assunto recente na comunidade de engenharia de software. O primeiro workshop sobre a prática de linha de produto de software

(Bass, 1997) foi promovido pelo Software Engineering Institute (SEI) em dezembro de 1996. O objetivo era compartilhar experiências da indústria de software e do governo com a prática de abordagem de linhas de produto visando explorar aspectos técnicos e não técnicos envolvidos (Lazilha 2002).

2.4.1 Elementos de uma Linha de Produto de Software

Segundo Clements (2001), as três atividades essenciais da construção de uma linha de produto são:

- Desenvolvimento do núcleo de artefatos;
- Desenvolvimento do produto;
- Gerenciamento da linha de produtos.

As atividades acima citadas estão relacionadas de tal forma que a alteração em uma delas, implica em analisar todas as outras.

Segundo Gimenes (2000), o processo de desenvolvimento de uma linha de produto de software pode ser visto como dois modelos de ciclo de vida, conforme mostra a figura 2.3.

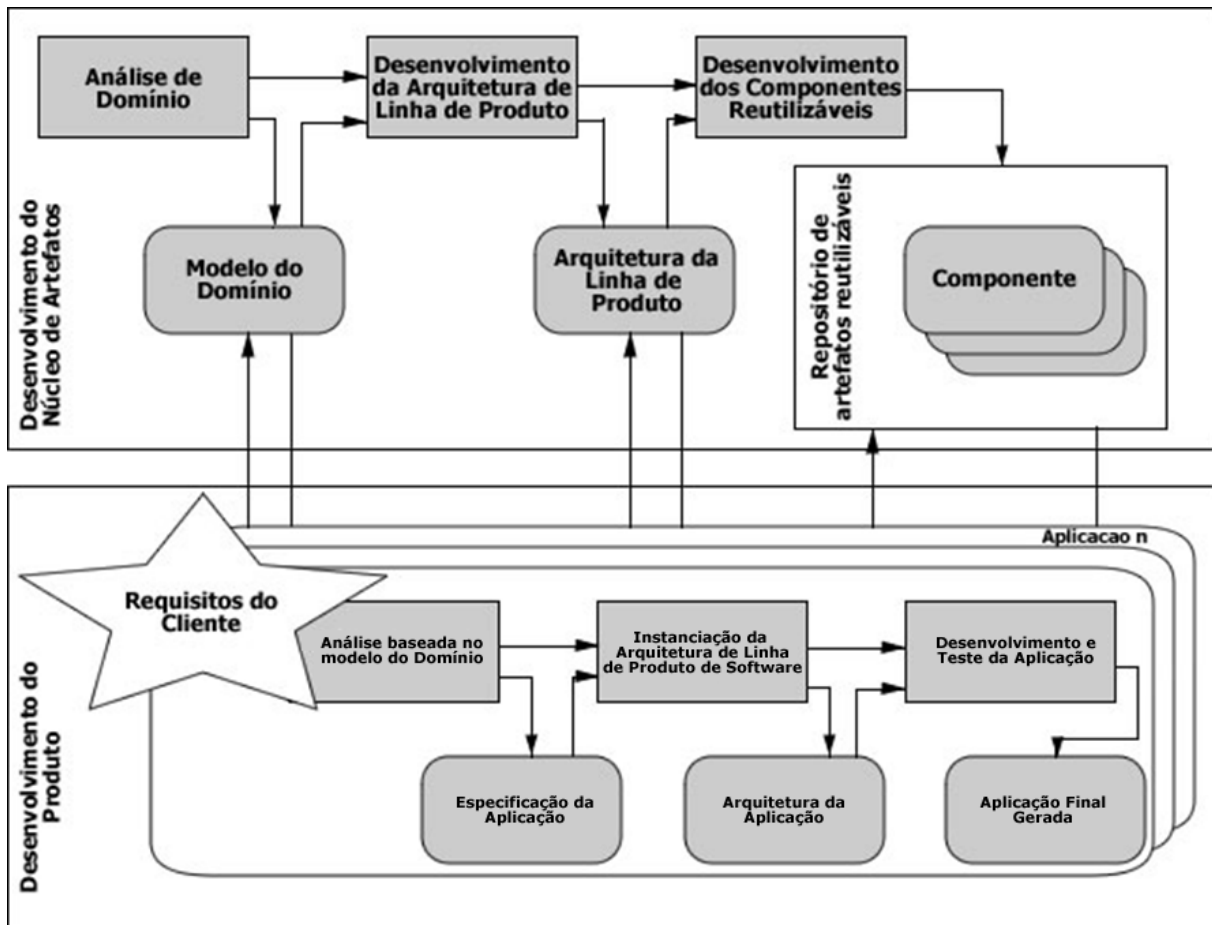


Figura 2.3: Processo de desenvolvimento de uma linha de produto de software

2.4.1.1 Variabilidade

A abstração e representação das variabilidades associadas, é um dos pontos mais importantes no desenvolvimento de uma Linha de Produto de Software, pois estão associadas à uma arquitetura e aos seus componentes. Diferenças em termo de comportamento, atributos de qualidade, plataforma, configuração entre outros, existem e podem diferenciar em relação a produtos em uma linha de produto (Clements, 2001).

A arquitetura, os componentes, as interfaces entre componentes ou as conexões entre componentes são distribuídas e reveladas por diferentes variações entre produtos. Em qualquer fase do ciclo de desenvolvimento essas variações podem ser reveladas, começando da análise de requisitos (Gimenes & Travassos, 2002).

A representação da orientação a objetos é feita em nível de projeto através de determinadas classes. Vários mecanismos são utilizados para o tratamento da variabilidade, que

nada mais é que o momento de reconhecimento de um ponto de variação até o mapeamento do ponto para uma instância de variação.

O conceito de feature é muito importante na representação de pontos de variação, pois o feature é uma característica de um produto que usuários e clientes consideram importantes na descrição e distinção de membros de uma família de produto (Griss, 1998). Uma feature pode estar relacionada a algumas características de alguns produtos, tais como funcionalidade, usabilidade e desempenho ou como característica de implementação como a plataforma de execução (Lazilha, 2002).

O modelo de feature geralmente é representado por um grafo, onde os nós contêm as features e seus atributos.

2.4.1.2 Desenvolvimento do Núcleo de Artefatos

Três aspectos são muito importantes no desenvolvimento dos artefatos: a definição do contexto da linha de produto, o núcleo de artefatos e o plano de produção. Um artefato é um item reutilizável de software utilizado como bloco de construção de uma linha de produto. Pode ser uma arquitetura de software, um componente, um framework, ou alguma documentação relativa à arquitetura de software ou a algum componente (Lazilha, 2002).

A arquitetura tem um papel muito importante entre os artefatos da linha de produto, pois representa a infra-estrutura central. A arquitetura de software descreve os elementos que compõem o sistema e as interações entre estes. A arquitetura de software também apóia questões importantes de projeto, tais como: a organização do sistema como composição de componentes, as estruturas de controle globais, composição de elementos do projeto, e a designação da funcionalidade dos componentes do projeto, sendo muito importante a existência de técnicas de engenharia, tornando a especificação precisa e possibilitando a flexibilização necessária para poder produzir vários produtos de uma mesma família.

O meta-projeto das aplicações que representam a família de produtos é representado pela arquitetura genérica ou framework de arquitetura da linha de produto. Vejamos alguns requisitos importantes da arquitetura de linha de produtos:

- Estabilidade, durante a vida da linha de produtos, sofrendo poucas alterações;

- Flexibilidade da integração das novas funcionalidades durante o ciclo de vida da arquitetura;
- Permitir a representação explícita de variações de modo a maximizar a reutilização;
- Fornecer mecanismos para implementar variações;
- Permitir as variações inerentes da família de aplicação, e permitir as variações impostas pelo mercado.

2.4.1.3 Desenvolvimento de Produto

Na engenharia de aplicação, em geral, alguns artefatos são incluídos:

- O modelo do domínio, que é a base para a identificação dos requisitos do cliente;
- Um framework de arquitetura de linha de produto serve como base para a especificação de uma arquitetura para um membro da família;
- Para gerar um produto, é necessária a integração de vários componentes.

A Figura 2.3 mostra a primeira atividade, descrita como análise baseada no modelo do domínio. Este modelo serve para mostrar as informações sobre a aplicação que será desenvolvida, definindo uma lista de características para a aplicação em questão. Sendo assim, este modelo vai representar as necessidades do cliente em relação à linha de produto e os componentes disponíveis. O modelo do domínio está vinculado ao modelo de decisão que define os principais aspectos a serem considerados no desenvolvimento do produto (Gimenes & Travassos, 2002). Com este modelo, é possível a identificação dos ajustes necessários nos valores iniciais das características descritas pelos componentes do modelo. Novos requisitos é a denominação para as necessidades que não são cobertas pelo modelo. Os novos requisitos podem implicar na necessidade de estender a arquitetura da linha de produto e adquirir novos componentes. Para os requisitos não satisfeitos pelo modelo existente, deve ser investigado o custo de desenvolvimento do software por encomenda, modificar a arquitetura ou os componentes existentes. A especificação do produto é formada pela lista de características da aplicação recuperada do modelo do domínio, com aquelas definidas nos novos requisitos.

A instanciação da arquitetura do produto é a segunda atividade. O framework de arquitetura é percorrido de cima para baixo, tomando-se decisões até que a arquitetura genérica

seja transformada na arquitetura específica do produto (Lazilha, 2002). Pontos de variação e decisões ainda não resolvidas podem ser pendências da instância inicial. Muitas vezes, essas pendências ocorrem devido aos requisitos específicos da aplicação. A implementação desses requisitos pode provocar modificações no framework de arquitetura de linha de produto ou apenas afetar a instância da arquitetura da aplicação (Lazilha, 2002).

2.5 Resumo

Neste capítulo foram apresentadas as Técnicas de Reutilização de Software, tais como, as vantagens e desvantagens na sua utilização, englobando os conceitos do Desenvolvimento Baseado em Componentes, Engenharia de Domínio e suas etapas.

Tentando diminuir de forma considerável o tempo de produção de um software, existem várias técnicas de reuso de software (Padrões, Componentes, Frameworks, Linha de Produto) no contexto da Engenharia de Software buscando constantemente uma significativa melhora na qualidade dos artefatos de software.

3 PORTFÓLIO ELETRÔNICO

No capítulo 3 serão abordados os conceitos de portfólio eletrônico. Na Seção 3.1 é apresentada a definição de portfólio eletrônico. Na seção 3.2 são apresentados os tipos de portfólio eletrônicos. Na seção 3.3 são destacados alguns portfólios eletrônicos existentes.

3.1 Definição

Foi na década de 80 que se deu origem a utilização dos Portfólios como meio de avaliação e de programas pedagógicos em muitas escolas públicas dos EUA (Mokhtari & Yellin 1996).

Com a reforma educacional, o uso dos portfólios na sala de aula vem crescendo constantemente, possibilitando manter um arquivo mais completo e organizado dos estudantes. Com a utilização do portfólio, é possível que os estudantes consigam avaliar seu progresso como aprendizes, principalmente por permitir a documentação da evolução dos alunos, possibilitando a análise das vivências diárias dos alunos, adaptando-se a qualquer área disciplinar (Wolf, 1989). Desta forma, os professores também podem acompanhar praticamente, em tempo real, o desenvolvimento do trabalho dos alunos, identificando problemas e orientando-os caso necessário.

O portfólio é uma ferramenta de acúmulo do progresso dos alunos em uma instituição de ensino servindo como base para um desenvolvimento profissional (Sistêlos, 1999).

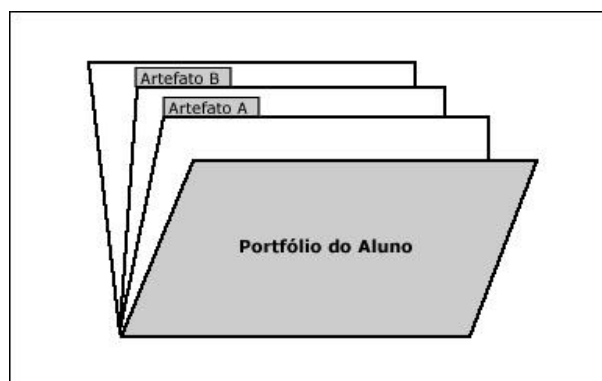


Figura 3.1: Portfólio Tradicional

De acordo com a figura 3.1, todo portfólio é composto de artefatos, que são todos os documentos que compõe o mesmo. Eles são os elementos de composição e são representados como pastas com orelhas, onde cada orelha representa um artefato.

Os artefatos representam conhecimento e habilidades que foram adquiridos pelo aluno no decorrer do curso.

Existem várias definições para portfólio, dentre elas podemos destacar:

“Um portfólio é uma pasta que mantém evidências da habilidade individual, idéias, interesses, e acompanhamentos de um estudante no tempo. Ele pode ser representado por um classificador preenchido com papéis selecionados, ... , ou por um disco laser que armazene imagens selecionadas dos trabalhos dos estudantes, tais como: arquivos de texto, sons, fotos, filmes, etc. Todavia ele deverá ser mais do que uma coleção de documentos ou imagens. Um portfólio bem projetado se constitui em um indiscutível corpo de trabalho do estudante” (Hart 1994).

“Um portfólio não é meramente um arquivo de projetos de curso e anotações, nem se trata de um colecionador de memoráveis atos de ensinamentos. Um portfólio é uma documentação organizada e direcionada a alvo do crescimento profissional e competência alcançada no complexo ato de ensinar. Através desta coleção de documentos, um portfólio é uma evidência tangível de um largo campo de conhecimentos, disposições e habilidades que se processam quando do crescimento profissional/educacional. Além disso, documentos em um portfólio são auto-selecionados, refletindo sua individualidade e autonomia” (Campbell, 1997).

Hoje em dia é muito difícil a adoção de programas de avaliação de aprendizagem que utilizam portfólios, pois o armazenamento e gerenciamento dessa ferramenta se tornam complexos pelo fato do acúmulo de muito papel, vídeos e fitas de áudio, pois seria necessária a utilização de várias salas para poder acumular tamanha quantidade de informação.

Para solucionar tal problema, seria necessária a utilização de portfólios eletrônicos, onde seria possível armazenar todas as informações acima citadas de forma eletrônica podendo ser acessadas através de computadores e gerenciadas eletronicamente (Lankes 1995).

Com a grande propagação da internet entre a população em geral, é possível armazenar o portfólio eletrônico em um servidor disponibilizando todas as informações aos alunos e o acompanhamento por parte dos professores em relação às atividades desenvolvidas pelos alunos seria muito mais dinâmica e quase que em tempo real.

Muitas são as razões para a utilização de um portfólio eletrônico, entre elas:

- Fazer com que os estudantes tenham mais segurança e motivação, e conseqüentemente uma participação mais efetiva no processo de auto-avaliação;
- Facilitar aos professores a realização de um acompanhamento mais preciso do progresso de um determinado aluno ou até mesmo de um grupo de alunos;
- Verificar os pontos fortes e fracos de uma determinada instituição de ensino, e posteriormente encontrar maneiras de minimizar os pontos fracos;
- Servir como complemento nos testes de avaliação.
- Diminuir consideravelmente a quantidade de papeis utilizada;

3.2 Tipos de Portfólios Eletrônicos

De acordo com (Sistêlos 1999) um portfólio eletrônico pode armazenar informações pessoais de alunos, tais como, trabalhos desenvolvidos, avaliação dos professores, atividades extracurriculares, orientações de professores, etc.

Segundo (Nascimento 2002), um portfólio eletrônico pode ser dividido em Portfólio do Aluno e Portfólio do Professor.

3.2.1 Portfólio do Aluno

De acordo com (Sistêlos 1999) é possível desenvolver atualmente dois tipos de portfólios de alunos: o Portfólio de Trabalho e o Portfólio de Apresentação.

No Portfólio de Trabalho serão armazenados todos os trabalhos dos alunos, sendo possível caracterizar o crescimento sistemático da documentação e ter informações sobre a progressão do aluno e sobre seus pontos fortes e fracos.

No Portfólio de Apresentação o objetivo é convencer que o aluno aprendeu o conteúdo passado pelo professor, pois nele estão guardados os melhores trabalhos que foram selecionados tanto pelos alunos como pelo professor. É a seleção de todos os trabalhos que melhor reflete a

competência arquivada no Portfólio de trabalho (Campbell 1997). Como exemplos, podemos citar: trabalhos realizados, apresentações, projetos, etc.

3.2.2 Portfólio do Professor

No Portfólio do Professor serão documentados os atributos pessoais e suas qualificações, tais como as disciplinas que serão lecionadas, participações em cursos, seminários e as estratégias didáticas que poderão ser utilizadas em sala de aula pelo professor (Nascimento 2002). Enquanto os professores avaliam os alunos, é possível mostrar aos professores através da construção do portfólio do professor um determinado crescimento profissional.

A seguir podemos identificar as principais funcionalidades do Portfólio do Professor:

- Identificação e registro das atividades que são realizadas por cada aluno;
- Acompanhar o desempenho do aluno, da turma, etc;
- Avaliação dos recursos utilizados;
- Serão geradas chamadas ao aluno de acordo com os prazos determinados pelos professores para uma determinada atividade.

3.3 Portfólios Eletrônicos Existentes

Vários tipos de portfólios eletrônicos foram desenvolvidos, como Portfólio Eletrônico para Web (Silva 2002), Portfólio-tutor (Nascimento 2002), POETA (Sistêlos 1999), Grady Profile (Auerbach & Associates), Connecticut College e-Portfolio (Eportconsortium 2009), PASS-PORT (Eportconsortium 2009), Epsilen (Eportconsortium 2009).

3.3.1 Portfólio Eletrônico para Web

É um sistema desenvolvido com base nos conceitos de portfólio eletrônico e a utilização de Banco de Dados Ativo e Temporal possibilitando aos seus usuários um acompanhamento contínuo da evolução do aluno (Silva 2002).

Arquitetura

A arquitetura do sistema portfólio eletrônico para web é desenvolvida em três camadas:

- **Camada de Apresentação:** composta pela interface do sistema (janelas, formulários de cadastro, formulários de consultas, etc.);
- **Camada de Aplicação:** é composta por dois pacotes (Domínio e Serviço). O pacote de Domínio é composto por oito pacotes, distribuídos da seguinte maneira: Turma, Disciplina, Usuário, Atividade, Equipe, Perfil, Análise de Dados e Fórmula.
- **Camada de Armazenamento:** é composta pelo mecanismo de armazenamento, que é representado pelo Banco de Dados Relacional.

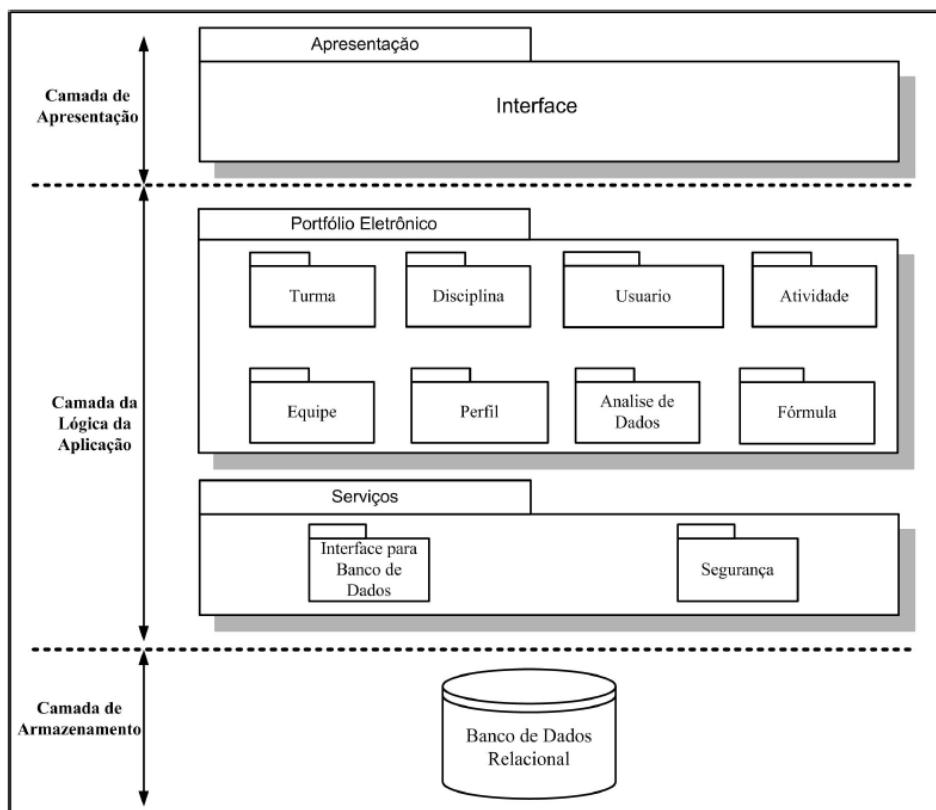


Figura 3.2: Arquitetura do Portfólio Eletrônico para Web

3.3.2 Portfólio – Tutor

O Sistema Portfólio-Tutor é parte integrante do projeto ACVA (Arquitetura de uma Classe Virtual Adaptativa) (Hernández-Domínguez 1997), onde seu principal objetivo é a construção de um ambiente de ensino/aprendizagem, com foco no ensino personalizado e na criação de um ambiente colaborativo (Nascimento 2002). Normalmente os alunos em uma turma de ensino, possuem conhecimentos e comportamentos diferentes entre si, sendo assim, a classe virtual poderá ser formada por grupos heterogêneos.

O POETA (Portfólio Eletrônico Temporal e Ativo) desenvolvido por Sistêlos (Sistêlos, 1999), será à base da funcionalidade da camada portfólio nesse sistema (Portfólio-Tutor) e essa aplicação foi desenvolvida para internet (Nascimento, 2002).

Arquitetura

De acordo com a figura 3.3, a arquitetura do sistema Portfólio-Tutor é desenvolvida em quatro camadas. Uma camada de persistência, composta pelo Sistema Gerenciador de Banco de Dados, duas camadas de negócio, composta pelo Portfólio Eletrônico (cujo principal funcionamento é o gerenciamento das informações, registro e acompanhamento das atividades) e a camada Tutor-ACVA (responsável pela parte de ensino). A última camada é a camada de Apresentação, representada pela interface.

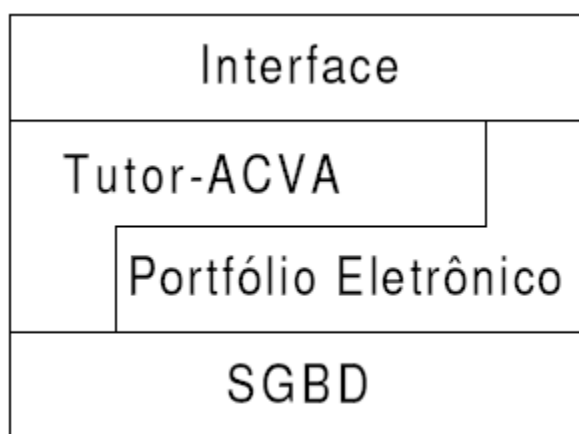


Figura 3.3: Camadas do Portfólio-Tutor

A seguir, é possível detalhar a estrutura de camadas da arquitetura do Portfólio-Tutor:

- **Camada de Armazenamento:** está composta pelo banco de dados relacional, objetivando o armazenamento das informações.
- **Camada de Serviços:** é composta por três pacotes (Acesso ao Banco de Dados, Segurança e Comunicação). O pacote Acesso ao Banco de Dados será responsável pela conexão do sistema ao banco de dados. O pacote de Segurança controlará o acesso às funcionalidades do sistema. O pacote de Comunicação será responsável pela comunicação Síncrona, envio de correio eletrônico e transferência de arquivos.
- **Camada de Domínio:** é composta por dois pacotes (Portfólio e Tutor-ACVA). O pacote Portfólio é composto por dois pacotes, distribuídos da seguinte maneira: Elementos Administrativos e Registros.

Já o pacote Tutor-ACVA é composto pelos seguintes pacotes: Base de Domínio, Perfil do Aluno, Estratégia Didática.

O Tutor-ACVA é baseado na arquitetura de um Sistema Tutor Inteligente.

O principal objetivo dos Sistemas Tutores Inteligentes (STI) (Giraffa 1999) é ensinar um determinado conteúdo ao aluno de acordo com suas necessidades individuais e ritmo de aprendizado. A Figura 3.4 mostra a arquitetura de um STI.

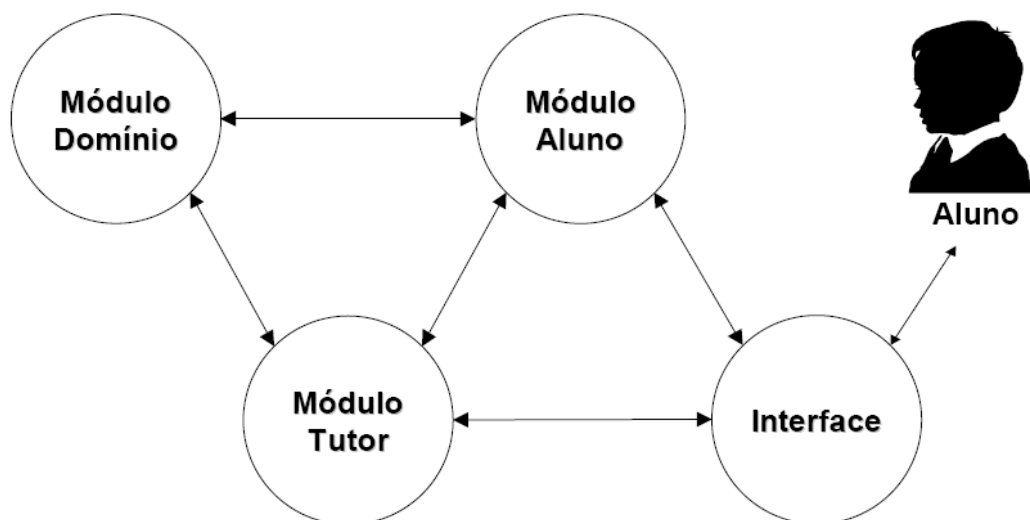


Figura 3.4: Arquitetura Clássica de um STI

Descrição dos modelos STI:

- **Módulo Aluno:** neste módulo são armazenadas as características individuais de cada aluno;
- **Módulo Tutor:** é onde está o conhecimento sobre as estratégias.
- **Módulo Domínio:** contém o conhecimento da disciplina no formato de regras de produção.
- **Interface:** permite a interação entre o aluno e o tutor.

A arquitetura do Portfólio-Tutor é representada na figura 3.5.

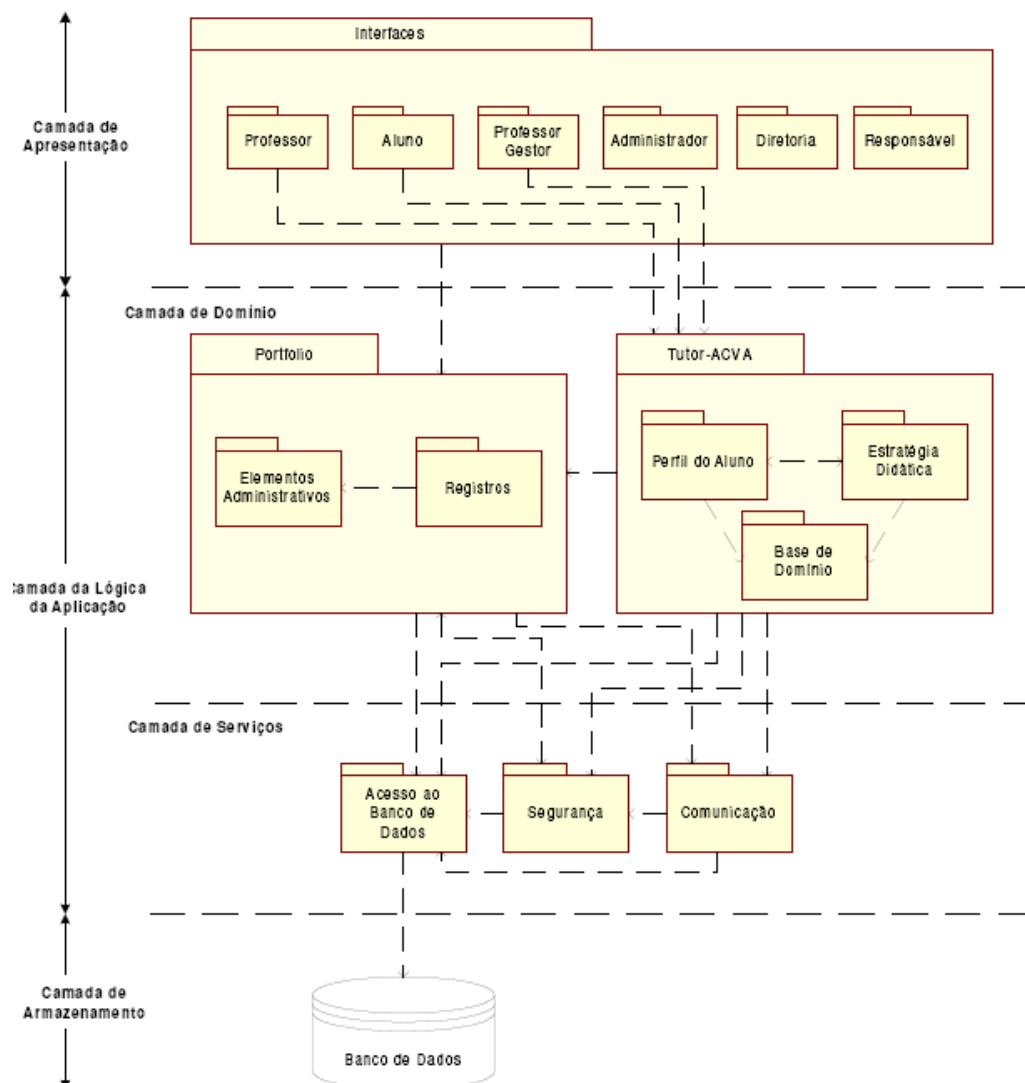


Figura 3.5: Arquitetura do Portfólio-Tutor

3.3.3 POETA

Utilizando o conceito de portfólio eletrônico, o POETA (Portfólio Eletrônico Temporal e Ativo) se torna um ambiente favorável à dinâmica de uma avaliação autêntica (Sistêlos 1999). Pode se tornar uma poderosa ferramenta de trabalho na sala de aula, principalmente pela utilização do portfólio do aluno, tanto na visão de apresentação como na visão de trabalho.

O POETA utiliza a tecnologia de banco de dados ativo e temporal, se tornando ativo no envio de mensagens e controle do gerenciamento das atividades do aluno (Sistêlos 1999).

Arquitetura

Com a sua utilização pela internet, é possível se ter uma avaliação do aprendizado dos alunos, fazendo uso do trabalho interativo e enfatizando a construção e avaliação do conhecimento adquirido. De acordo com a figura 3.6 o POETA foi projetado e desenvolvido com intuito de funcionar em uma rede local.

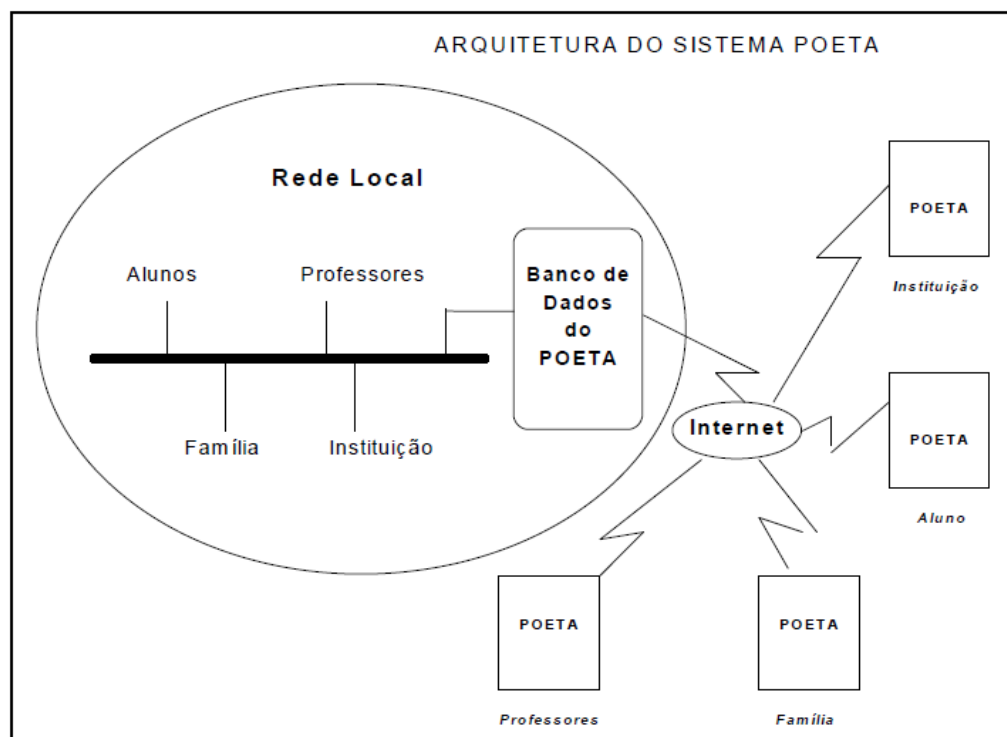


Figura 3.6: Arquitetura do Sistema Portfólio Eletrônico POETA

Vários elementos estão envolvidos no aprendizado do aluno, inclusive a interação da família com a instituição de ensino.

Na interação do aluno com o professor é possível se ter um trabalho em conjunto, onde os professores elaboram tarefas e os alunos as executam, os professores avaliam essa execução e os alunos têm acesso a essa avaliação, com a possibilidade de uma possível correção. No Portfólio de Apresentação do aluno é possível selecionar um artefato tanto pelo aluno como pelo professor.

O progresso do aluno e seu desempenho em sala de aula poderão ser acompanhados pela família a qualquer momento. Na Figura 3.7 mostram-se as interações entre o portfólio eletrônico e o aluno, professor, instituição e família.

DIAGRAMA CONTEXTUAL

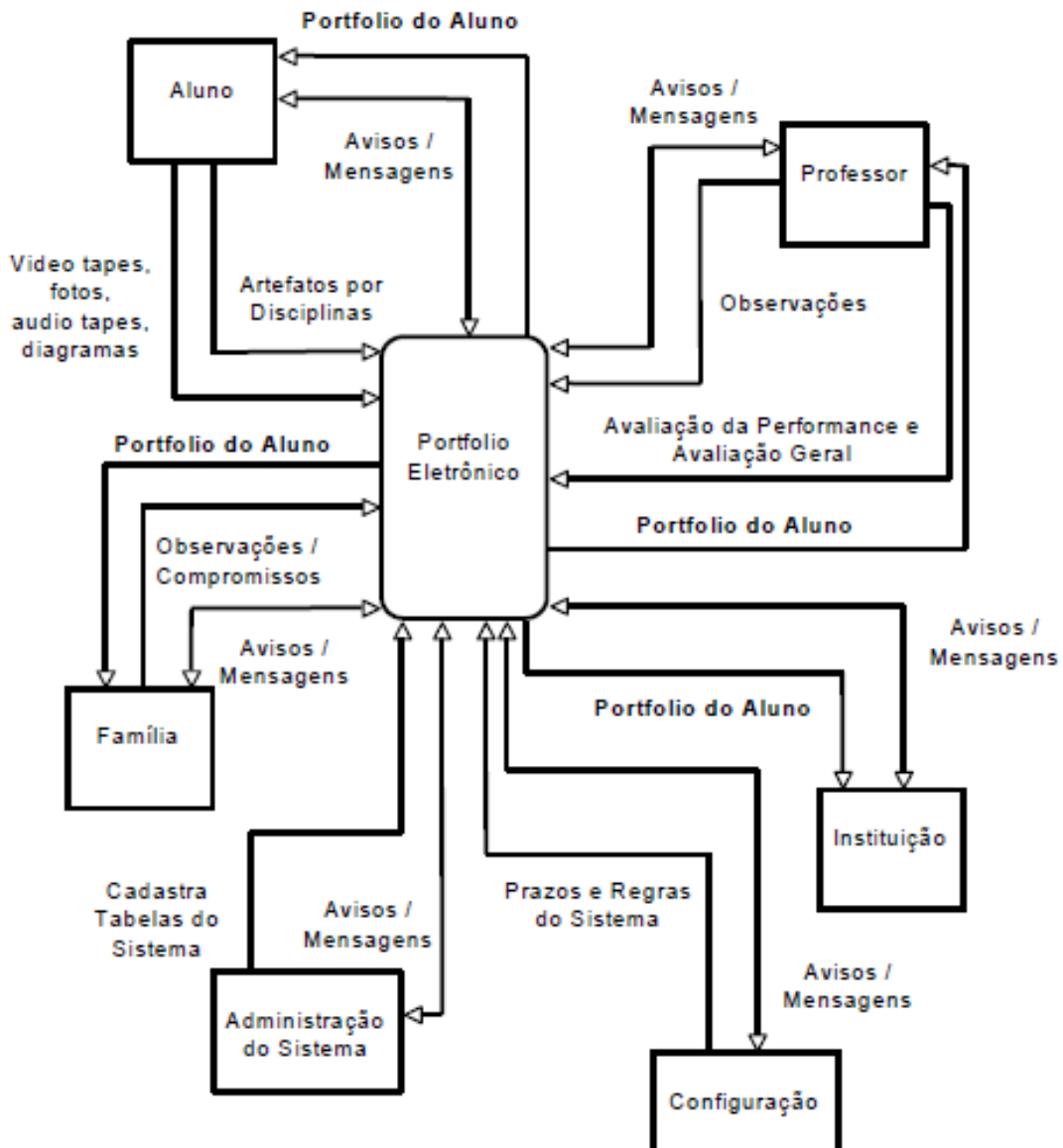


Figura 3.7: Diagrama de Contexto do Sistema POETA

3.3.4 Grady Profile (*Auerbach & Associates*)

É um dos softwares mais completos em sua área, e foi desenvolvido pela Auerbach & Associates (Eportconsortium 2009).

Foi desenvolvido para a plataforma Macintosh, e é necessário estar cadastrados para se ter um acesso inicial, onde será solicitada uma senha de acesso ao usuário e todo o material do estudante estará disponível ao mesmo após o login. É uma ferramenta de aprendizagem tanto para o aluno quanto para o professor, pois nele é possível fazer uma avaliação honesta referenciando experiências anteriores.

Algumas funcionalidades do Grady Profile podem ser destacadas:

- É possível poupar tempo e espaço por parte do professor em relação ao material de aula disponibilizado;
- Encontrar itens específicos com facilidade, organizando os documentos através de datas e títulos;
- Permite ter acesso aos trabalhos de estudantes em formato digital:
 - **Sons**: Músicas, língua estrangeira, entrevistas, etc.;
 - **Leitura oral**: é possível fazer uma gravação oral de um determinado material disponibilizado em sala de aula;
 - **Gráficos**: desenhos, fotos, gráficos, etc.;
 - **Vídeo**: apresentações, trabalhos em grupo, artes, etc.;
 - **Páginas Web**: onde os estudantes podem gerar suas páginas web.
- Avaliação do progresso de um determinado aluno através de comparação de amostras de trabalhos ao longo do tempo;
- Disponibilização de um local onde o aluno pode fazer uma auto-avaliação e refletir sobre seus trabalhos;
- Gera relatórios em diversos formatos;

O sistema trabalha com hypercards para cada artefato, sendo possível sua customização através do usuário, onde é possível conter informações dos professores e dos alunos em uma determinada disciplina (Sistêlos 1999). O sistema funciona como um registro de dados pessoais de cada estudante.

3.3.5 Connecticut College e-Portfolio

Desenvolvido por Career Enhancing Life Skills (CELS) (Eportconsortium 2009), o Connecticut College e-portfolio é uma ferramenta voltada para internet, onde é documentado toda a experiência de desenvolvimento durante o período universitário do aluno, tais como, estudos acadêmicos, acompanhamento extra-curriculares entre outros.

Por ser uma ferramenta on-line, toda a documentação armazenada no portfólio pode ser apresentada de forma dinâmica ao professor. As habilidades e experiências acumuladas durante o curso são apresentadas como evidência de seu desenvolvimento e automaticamente podem ser incluídas em seu portfólio de apresentação.

É grande a interação entre o aluno e o professor, pois o estudante pode dar permissão a um determinado portfólio de apresentação utilizando um ambiente seguro. Também é possível se fazer upload de arquivos, onde os estudantes documentam informações em seus arquivos acadêmicos. De acordo com os conceitos de portfólio, (ver seção 2.2.1), o Connecticut College e-Portfolio também possui um portfólio de apresentação, onde é possível arquivar todo o material que melhor reflete a competência do aluno durante o curso, armazenando informações sobre si e suas experiências extra-curriculares nos mais diversos formatos multimídia. É também disponibilizado pelo portfólio um *Programa de Resumo*, onde é criado um documento com o resumo do aluno. É possível compartilhar informações de alunos para outros professores e até mesmo outros alunos através da função *Notas*.

3.3.6 PASS-PORT

O gerenciamento eletrônico dos dados é uma das principais funcionalidades dos serviços que o PASS-PORT oferece. É uma ferramenta que pode ser utilizada para ajudar a coletar, organizar e agregar os seus dados (Eportconsortium 2011).

O PASS-PORT pode ser facilmente personalizado para garantir que ele atenda às suas necessidades de acordo com o seu desenvolvimento.

Fornecer uma riqueza de recursos para facilitar a coleta de dados em uma variedade de locais. Com o PASS-PORT você pode obter:

- Portfólios eletrônicos baseados em padrões: onde você poderá construir portfólios a partir do zero ou a partir de modelos e em seguida submetê-los a avaliação ou simplesmente publicá-los na web;
- Registro de aprendizagem experimental: onde você terá documentado todo o serviço de aprendizagem, experiência de campo, aprendizado de estudante associando sites de forma que possa avaliar todos os membros participantes;
- Relatórios integrados: é possível obter os dados através de planilhas eletrônicas, exportando todas as informações não só para planilhas, mas também para vários tipos de documentos, tais como, .doc, .pdf, .html.

O PASS-PORT está disponível em várias versões, cada uma projetada para atender as necessidades específicas de cada organização.

PASS-PORT HE: foi projetado para ajudar as faculdades e escolas a coletar e agregar uma grande variedade de tipos de dados. Esta função tem levado ao desenvolvimento de características que são fortemente adequadas para a avaliação de candidatos a professores. permite que você defina pontos de transição para cada programa, cada uma com suas próprias necessidades. Estes pontos de transição, chamado Portais, pode exigir de seus candidatos a apresentação de uma variedade de artefatos para a avaliação por meio de múltiplos avaliadores. Esta informação serve a dois propósitos: ajudar a identificar os alunos que precisam de correção, mas o mais importante, ele fornece uma medida uniforme de todos os candidatos no mesmo ponto em sua progressão, com os mesmos requisitos, utilizando os mesmos instrumentos de avaliação.

PASS-PORT BE: PASS PORT Business Edition se desenvolve a partir das técnicas altamente eficaz pioneira na versão HE (para os programas de preparação de professores). O Business Edition em grande parte difere da versão HE no que se refere a nomenclatura de educação específica, tais como escolas e professores, e substituí-los com termos mais genéricos, como sites e contatos.

PASS-PORT CE: traz o mesmo conjunto poderoso de ferramentas de outras edições de avaliação de educação geral. A edição campus também contém um conjunto especial de ferramentas para agregar dados através de múltiplas instalações de PASS-PORT em um único

campus, permitindo faculdades manter seus próprios dados, possibilitando o acesso a diversos tipos de relatórios.

Logo abaixo poderemos observar a tela inicial do PASS-PORT, onde é solicitado o usuário e senha para acesso ao mesmo.



University of Louisiana at Lafayette

WELCOME BROWSE FACULTY F.A.Q. CREDITS CONTACT

NEWS

No News

Welcome to
PASS-PORT

First Time Users Click Here

Sign In

username helpdesk

password ●●●●●●

GO

Forgot Your Password?
Click Here

Search

All Folios

For

GO

Figura 3.8: Tela de Login do Sistema PASS-PORT

Após o login o usuário é redirecionado para outra página, onde a partir da mesma será possível ter acesso a todas as funcionalidades do sistema apresentado.

University of Louisiana at Lafayette LAUNCH NAV-TOOL

MY PASS-PORT ACCOUNT INFO ARTIFACTS FOLIOS HELP CENTER CONTACT

NEWS
3/27/04
ALERT
Need help using PASS-PORT? Weekly Open Lab Session; Lee Hall - Room 213. Every Monday, 11:00 AM to 12:00. Every Wednesday, 5:00 to 6:00 PM. Please bring questions. ***** You do not need to register for these sessions.

Export My Portfolios Analyze My Browser

RECOMMENDATIONS REQUEST RECOMMENDATION ?

You have not received any recommendations.

TASKS ASSIGNED TO ME VIEW ARCHIVE ?

Due Date	Task	Origin
None	ALERT from PASS-PORT Created: 3/27/2004 1:20:14 PM - Pending	Douglas Williams

TASKS I HAVE COMPLETED VIEW ARCHIVE ?

Figura 3.9: Tela Principal do Sistema PASS-PORT

3.3.7 Epsilen

O Ambiente Epsilen é o resultado de seis anos de pesquisa e desenvolvimento na Escola de Engenharia e Tecnologia da Universidade de Indiana.

O sistema é projetado para envolver os alunos de hoje em seu mundo digital, totalmente combinado e integrado com a Web 2.0, e com as melhores práticas do curso de eLearning de entrega, de modo que professores e alunos possam colaborar facilmente no campus e em todo o mundo (Eportconsortium 2011).

Cursos e avaliações são entregues de forma rentável por meio de um software, sem necessidade de hardware local ou suporte, possui uma interface consistente que acelera a curva de aprendizagem e incentivo ao uso do sistema.

O Ambiente Epsilen pode ser facilmente integrado com outros sistemas, como sistemas de informação do estudante para eliminar a necessidade de entradas duplicadas, bem como em outros sistemas de gestão de aprendizagem.

O Ambiente Epsilen inclui três módulos interligados, Epsilen ePortfolio, Epsilen Rede Global, e Epsilen Sistema Global de Gestão de Aprendizagem. Todos os três módulos

compartilham um conjunto de ferramentas de rede integrada. Uma configuração alternativa, sem a rede LMS, Epsilen ePortfolio e Global, também está disponível.

Com o ambiente Epsilen é possível coletar seu histórico de trabalho ao longo da vida, pois ele é totalmente integrado ao ambiente de eLearning. Este módulo inovador permite que educadores e estudantes armazenem uma grande variedade de materiais para mostrar o desempenho e progresso em uma coleção constante de artefatos multimídia, blogs, wikis e discussões.

As instituições de ensino pode encontrar alunos em seu mundo digital e mudar profundamente a forma como elas ensinam. Os alunos podem levar seus ePortfolio completos com eles após a formatura e adicionar a ele indefinidamente durante a formação contínua e ao longo da sua carreira.

Formato Multimídia

Fácil de usar, pois possui um formato multimídia que suporta perfis, fotos, vídeos e criação de artefatos, permitindo uma gama de informações a serem apresentadas aos potenciais empregadores ou avaliadores de forma convincente.

Opções de Privacidade

Com acesso controlado pelo usuário, e não pela instituição, alunos e professores podem selecionar as configurações de privacidade e opções de compartilhamento para permitir quais assessores e avaliadores externos poderão ter acesso a essas informações.

Avaliações Institucionais

As instituições também podem coletar o trabalho de ePortfolios dos estudantes para documentar o progresso do aluno a longo prazo. Estas coleções de arquivos podem ser utilizadas em avaliações institucionais do curso, departamento, programa ou a nível de instituição.

Compatibilidade LMS

Possui Compatibilidade com a maioria dos outros sistemas de gestão de aprendizagem garantindo a integração uniforme e eficaz.

Rede integrada Epsilen global

O conjunto de ferramentas Epsilen Rede Global está totalmente integrado no Epsilen ePortfolio, permitindo que alunos e professores possam incorporar projetos de seus cursos, ou notas de arquivo, blogs e fóruns de discussão.

Nas próximas figuras podemos observar a tela de login do Epsilen e a tela de um professor com todo o conteúdo compartilhado.

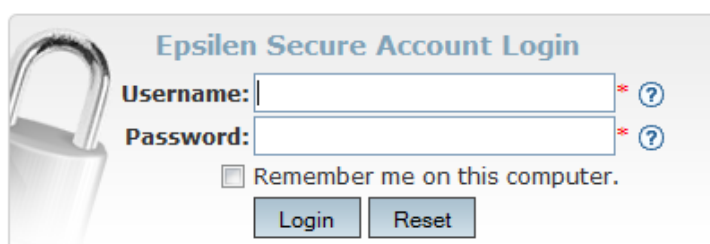
The image shows a login form titled "Epsilen Secure Account Login". On the left side of the form is a graphic of a silver padlock. The form contains two input fields: "Username:" and "Password:". Each field has a red asterisk and a question mark icon to its right. Below the password field is a checkbox labeled "Remember me on this computer.". At the bottom of the form are two buttons: "Login" and "Reset".


Figura 3.10: Tela de Login do Sistema Epsilen

Logo após a inserção das informações do nome de usuário e sua respectiva senha, o mesmo será direcionado para sua página pessoal, onde será possível compartilhar todas as suas informações e documentos, que poderão ser analisados pelas pessoas que terão acesso à essas informações. Ainda na tela de login, caso o usuário tenha esquecido seu login ou até mesmo sua senha, é possível através de uma opção de preenchimento do seu e-mail, que previamente foi cadastrado na criação do seu portfólio, ou seja, a partir do momento em que você se registrou no Epsilen, recuperar essas informações.

Para se registrar no sistema é necessário quatro passos: inserir o seu e-mail da instituição, escolher o tipo de conta, fornecer as informações do seu perfil e ativar sua conta através das informações que foram enviadas para o seu e-mail.

EPORTFOLIO REGISTRATION


1 **Step 1**
Provide your email address
2 **Step 2**
Choose account
3 **Step 3**
Provide your profile information
4 **Step 4**
Done, activate account
ePortfolio Meter
0%

 **Please enter your email address to begin registration.**
 Please note that when you register for an Epsilon account, you will be sent an Account Activation email from Epsilon.com. If you do not receive this email in your Inbox, please check your spam/junk folder as some email programs will block mail from Epsilon.net as spam.


Email Address: *


[Already registered](#)
[Forgot password](#)

Figura 3.11: Registrando o Portfólio


Hall Gardner Last Modified: 09/19/2011
<http://www.epsilon.com/hgardner>
Access Key
What is Access Key?

Current View: Public


Hall Gardner
 Professor
 Primary Email: hall.gardner@aup.edu
 Secondary Email:
 Office Phone:



September 2001 Interviews
 On September 16, Hall Gardner was interviewed by Al Jazeera on *Inside Story* on the Libyan revolution, along with Nasreldin Bukatif, a member of the International and Political Affairs Committee of the Libyan National Transitional Council and Abdulhamit Bilicic, a columnist for *Today's Zaman* newspaper. <http://www.aljazeera.com/programmes/insidestory/2011/09/201191782733278136.html>
 He was likewise interviewed by France 24 on ten years after September 11, as well as by Russia Today, RT, <http://rt.com/news/war-terror-world-now-257/>.
 One of the major newspapers of Iraqi Kurdistan, the Gulan Weekly, also interviewed him, on the theories and philosophies of revolution. Link forthcoming... <http://www.gulan-media.com/#>

Recent Activities
 Hall Gardner and Oleg Kobtzeff just completed the co-edited book, *The Ashgate Research Companion to War: Origins and Prevention*---- to be published by Ashgate in the coming year. The book features chapters dealing with the theory and history of systemic war. Authors include a number of AUP professors among other top experts in the field: Hall Gardner; Azar Gat; Debra L. DeLaet; Oleg Kobtzeff; Andrew Robinson; Peter Emerson; Troy Davis;

THE AMERICAN UNIVERSITY OF PARIS
 Home | Resume | ShareIt | Showcase | Certifications | Blog | News | Publications | Login

Quick Links

Figura 3.12: Portfólio público do professor Hall Gardner

3.3.8 Comparação entre os portfólios analisados

Com base nos sistemas portfólios analisados (Tabela 3.1), o trabalho proposto tem como principal contribuição o reuso da arquitetura do sistema portfólio-tutor baseado no reuso de componentes, e a redução de esforços, diminuindo o tempo de desenvolvimento de um novo sistema portfólio eletrônico ou portfólio-tutor.

Tabela 3.1: Comparação de funcionalidades entre os portfólios analisados

	1	2	3	4	5	6	7	8	9	10	11	12	13
Portfólio-tutor proposto													
Portfólio eletrônico para web													
Portfólio-tutor													
Poeta													
Grady profile													
Connectcut college e-portfólio													
Pass-Port													
Epsilen													

Tabela 3.2: Legendas de Funcionalidades

Legenda	Funcionalidade
1	Reuso de Software
2	Debate Síncrono
3	Debate Assíncrono
4	Atividade
5	Táticas de Ensino
6	Estratégias Didáticas
7	Sessão de Ensino
8	Páginas Web
9	Leitura de Arquivos pdf
10	Gráficos
11	Vídeos
12	Sons
13	Relatórios

3.4 Resumo

Foram apresentados os conceitos de portfólio eletrônico, os tipos de portfólios eletrônicos existentes que são divididos em portfólio do aluno e portfólio do professor. Foram destacados alguns sistemas portfólios eletrônicos, tais como, Portfólio Eletrônico para Web, Portfólio-Tutor, POETA, Grady Profile, Connecticut College, Webfólio, PASS-PORT e Epsilen e suas respectivas comparações em relação aos serviços disponibilizados por cada um deles.

É grande a necessidade de um acompanhamento detalhado de toda a vida acadêmica de um aluno, bem como sua produção, seu desempenho, entre outras informações que poderiam ser obtidas através da utilização de portfólios eletrônicos, pois com essas informações chegando de forma dinâmica, o professor poderia ajudar individualmente cada aluno e a instituição de forma mais efetiva, conseqüentemente o aluno poderia se sentir mais motivado pelo fato de saber que os portfólios eletrônicos seriam disponibilizados a todos, e produzir trabalhos de qualidade.

4 MODELAGEM DO SISTEMA PORTFÓLIO-TUTOR

No capítulo 4 será abordada a modelagem do sistema proposto. A modelagem do sistema é de suma importância pelo fato de se ter uma visão abstrata dos processos do sistema. Na Seção 4.1 serão apresentados os requisitos do sistema. Na seção 4.2 são apresentados os Casos de Uso Essenciais. Na seção 4.3 serão destacados a análise e o projeto do sistema proposto. Na seção 4.4 será mostrada a arquitetura do sistema portfólio proposto. Na seção 4.5 será mostrado o funcionamento de uma aplicação Portfólio-Tutor. Na seção 4.6 serão apresentados os diagramas de classe dos componentes e na seção 4.7 o resumo do capítulo.

4.1 Modelo do Sistema Portfólio-Tutor

Um sistema portfólio-tutor funciona basicamente organizado em três camadas: Apresentação, Tutor e Portfólio.

Na camada apresentação, nós temos a interface que é vital para o sucesso de qualquer sistema interativo, podemos dizer que a interação é ponto fundamental neste tipo de sistema, pois é onde o sistema tutor exerce duas de suas principais funções:

- Apresentação do material instrucional;
- A monitoração do progresso do estudante através da recepção da resposta do aluno.

Alguns objetivos devem ser cumpridos com base nessas duas funções. Evitar que o estudante se entedie, fazendo-se necessária uma riqueza de recursos no material instrucional.

Na camada tutor, nós podemos dividir em quatro partes: *Base de Domínio, Controle, Perfil do Aluno e Estratégias Didáticas.*

O componente chave do sistema tutor é a base de conhecimento do domínio, onde ali estará representado o material instrucional, ou seja, o conteúdo que será ministrado pelo tutor.

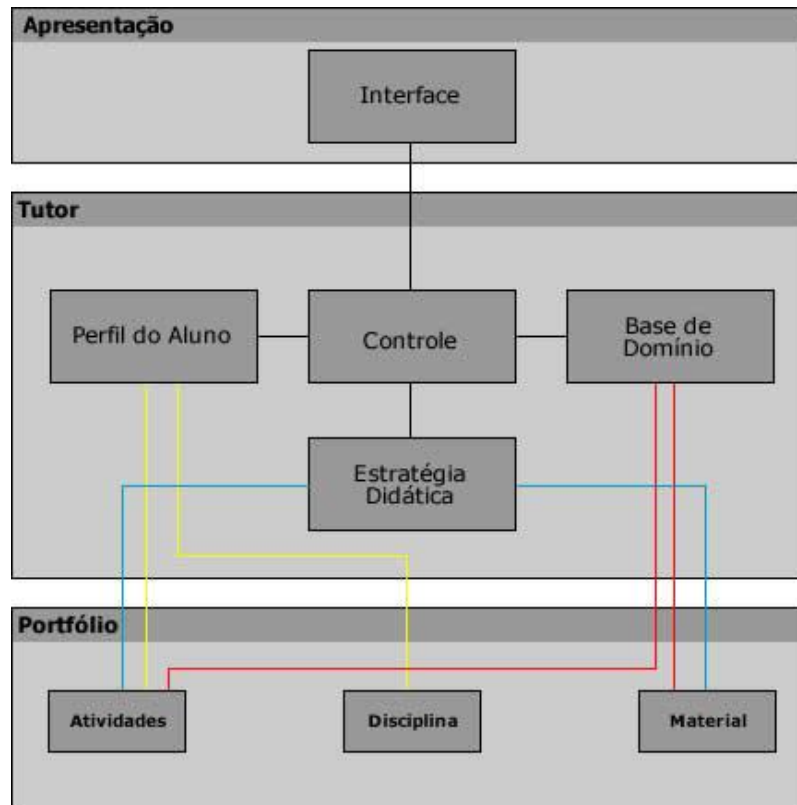


Figura 4.1: Modelo de um Sistema Portfólio-Tutor

O principal fator que determina a diferença entre um sistema convencional de um sistema tutor é o fato de o conteúdo ser armazenado em uma base de conhecimento e não em uma base de dados convencional, como acontecem com os sistemas convencionais. A base de conhecimento permite que o sistema tenha a possibilidade de raciocinar de acordo com o conteúdo a ser ministrado.

O módulo de controle é o responsável pela coordenação geral do tutor, no que diz respeito às suas funções, interfaces, troca de informações e comunicação com outros programas utilitários através do sistema operacional.

No perfil do aluno é representado o conhecimento e as habilidades cognitivas do aluno em um determinado momento. Sua constituição é feita por dados estáticos e dados dinâmicos, que servirão de base para o tutor ter o poder de comprovar hipóteses a respeito do aluno.

Algumas técnicas podem ser utilizadas para construir o modelo do aluno, tais como: colocar as preferências dos alunos e seus objetivos particulares, coisas que geralmente o aluno esquece quando interage com o tutor, indicar seus objetivos particulares e fazer uma comparação entre a conduta de um aluno com a de um especialista, verificando os pontos em comum.

Na estratégia didática, podemos vê-la como um plano, ou seja, uma estrutura de ação visando atingir determinados objetivos. As estratégias didáticas podem ser vistas como esquemas de plano que definem formas de apresentar o material instrucional ao aluno. Elas constituem conhecimento sobre como ensinar, ou seja, sobre como gerar, a partir das informações de diagnósticos, uma seqüência de táticas de ensino capazes de apresentar com sucesso um determinado tópico a um determinado estudante.

A camada portfólio funciona basicamente organizando a camada portfólio eletrônico em duas outras camadas, portfólio do aluno e portfólio do professor, onde o primeiro é subdividido em portfólio de trabalho e portfólio de apresentação. Todas as atividades atribuídas pelo professor aos seus alunos, serão registradas no portfólio de trabalho e são classificadas de acordo com os artefatos utilizados pela instituição de ensino. De acordo com os itens de avaliação pre-estabelecidos, os documentos que foram gerados por cada aluno serão arquivados e avaliados pelo professor. Os melhores trabalhos que serão selecionados em comum acordo entre aluno e professor, estarão disponíveis no portfólio de apresentação.

Tipos de artefatos deverão ser definidos pela instituição de ensino e posteriormente serem utilizados pelos professores. A seguir podemos observar alguns tipos de artefatos dos alunos:

- Projetos de pesquisa;
- Visitas de campo;
- Debates;
- Seminários;
- Provas; entre outros.

Esses artefatos poderiam ser avaliados da seguinte maneira:

- Apropriação de conteúdo;
- Clareza de exposição;
- Capacidade crítica;
- Criatividade;
- Vocabulário diversificado; entre outros.

No portfólio professor, seriam documentados apenas atributos pessoais e qualificações profissionais, certificados, trabalhos de pesquisa e extensão, participações em cursos, entre

outros, que inicialmente não estariam no portfólio-tutor. O portfólio professor também estaria subdividido em portfólio instrucional onde se registraria:

- Disciplinas para as quais está qualificado;
- Disciplinas e turmas lecionadas;
- Estratégias didáticas utilizadas;
- Planejamento de unidades;
- Planejamento de aulas;
- Recursos didáticos empregados.

4.2 Requisitos

Por meio da notação UML foi possível realizar a modelagem do sistema, onde é possível fazer uma abstração do mundo real, simulando toda a funcionalidade do sistema proposto, só que de uma forma muito mais simples do que na realidade, capturando as partes essenciais do sistema.

Os principais requisitos do sistema proposto são:

- Um sistema Portfólio-Tutor Baseado em Componentes;
- Utilização de uma arquitetura de software reutilizável;
- Controle de Acesso
- Cadastro de Perfis
- Realização de Avaliações (Aluno/Professor/Instituição)
- Realizar Testes on-line
- Criar Lista de discussão
- Acompanhar Desempenho
- Realizar Pesquisa
- Controlar Atividades de Disciplinas
- Levantar Questionamentos
- Esclarecer Dúvidas
- Calcular Média

- Fornecer Notas
- Disponibilizar Plano de Ensino

4.3 Casos de Uso Essenciais

Através da análise de casos de uso podemos identificar as interações dos atores com o sistema. No contexto do portfólio-tutor proposto os casos de uso foram divididos da seguinte forma:

- *Diagrama de Caso de Uso para Administrador*: Onde o Administrador terá acesso ilimitado, sendo assim o grande responsável pelo sistema.
- *Diagrama de Caso de Uso para Professor*: Onde o Professor poderá administrar o conteúdo de suas aulas, plano de ensino, avaliações, atividades entre outras funções.
- *Diagrama de Caso de Uso para Aluno*: Onde estarão disponíveis consultas e seleção de disciplinas disponíveis de acordo com o curso que está matriculado.
- *Diagrama de Caso de Uso para Monitor*: Onde o monitor tem basicamente a mesma automação do aluno com alguns acessos privilegiados com base nos privilégios do professor.

4.3.1 Diagrama de Caso de Uso para Administrador

Através da Figura 4.1 é possível perceber que o administrador do sistema será o grande responsável pelo sistema, pois ele poderá analisar uma solicitação de cadastramento de um determinado professor no portfólio, além do acadaastro dos usuários com seus respectivos níveis de acesso, diferenciando-os como professor e aluno, onde esse ultimo poderá também ser um monitor. O administrador também será o responsável pelo cadastramento de Turmas, Debates (Forum) e Salas Virtuais, além de manter todos os dados atualizados, alterando-os ou excluindo-os.

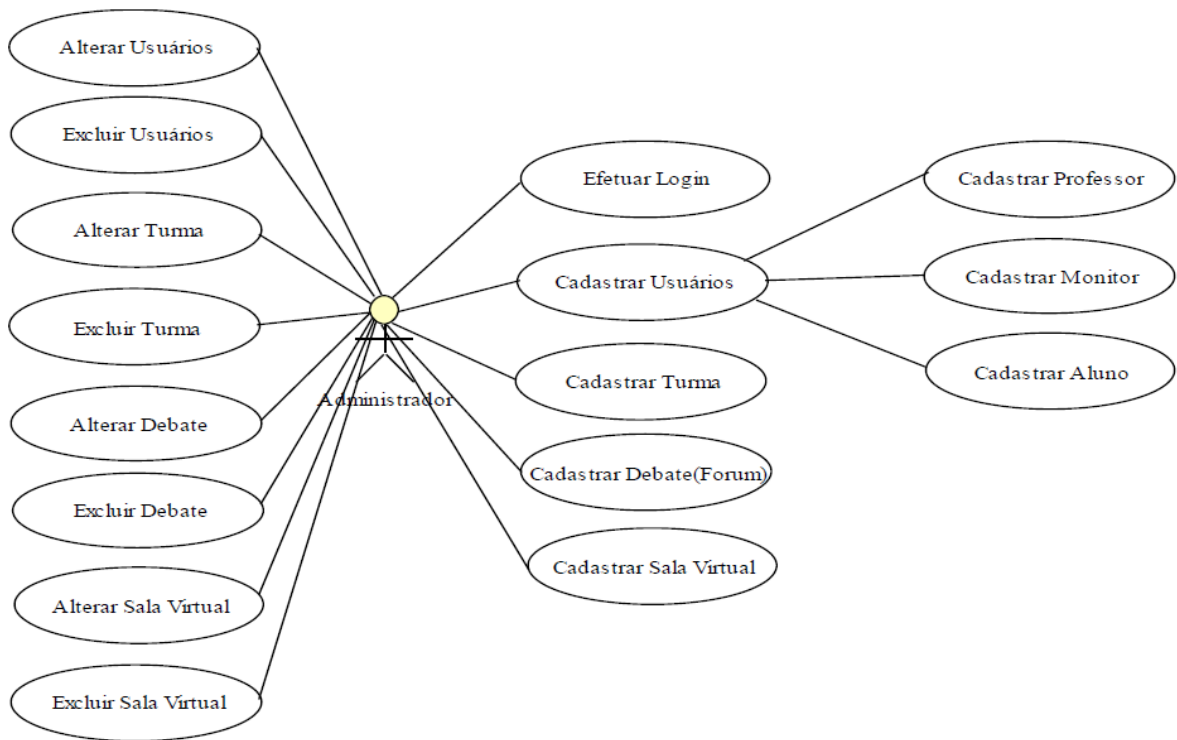


Figura 4.2: Diagrama de Caso de Uso do Administrador

4.3.2 Diagrama de Caso de Uso para Professor

No diagrama de caso de uso do professor (Figura 4.2) o professor fará a solicitação de acesso através do login, onde será retornado todo conteúdo relativo a sua área de administração, como inserir/alterar uma nova aula, inserir/alterar um plano de ensino, consultar as avaliações ou atividades disponibilizadas aos alunos, consultar/atualizar a fórmula, onde esta é a responsável por se chegar a uma determinada média para o aluno de acordo com as avaliações existentes.



Figura 4.3: Diagrama de Caso de Uso do Professor

4.3.3 Diagrama de Caso de Uso para Aluno

No diagrama de caso de uso do aluno (Figura 4.3) o aluno pode consultar e selecionar as disciplinas que estão disponíveis, verificar avisos, consultar suas notas, enviar atividades que foram disponibilizadas por um determinado professor, participar de uma sala virtual ou debates e consultar os planos de ensino.

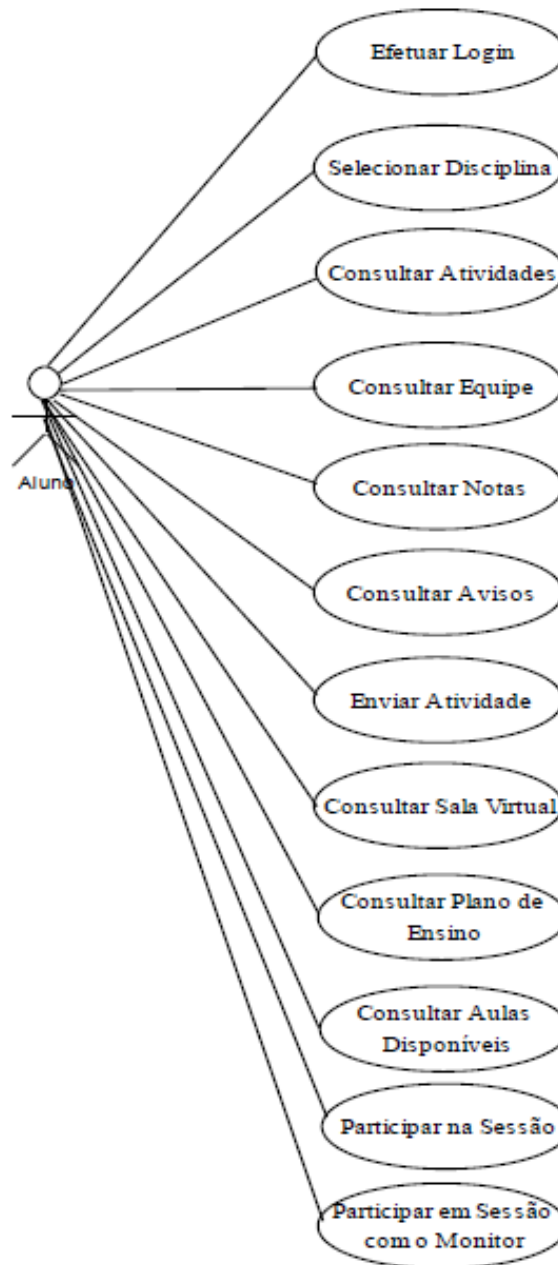


Figura 4.4: Diagrama de Caso de Uso do Aluno

4.3.4 Diagrama de Caso de Uso para Monitor

Através desse diagrama de caso de uso, é possível se notar que o usuário monitor nada mais é que um usuário aluno com algumas “regalias” a mais baseadas nas regalias dos professores. O monitor é o responsável pela atualização das atividades, da fórmula, e das equipes, assim como o professor.

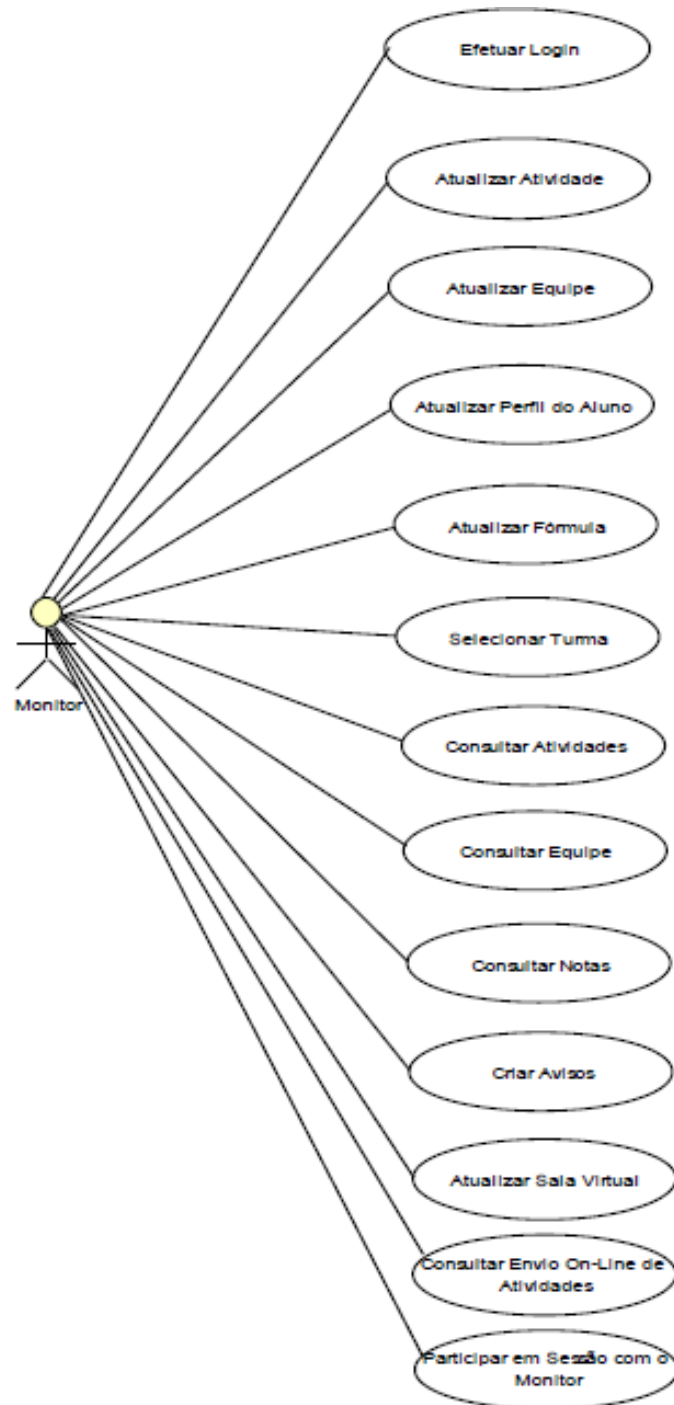


Figura 4.5: Diagrama de Caso de Uso do Monitor

4.4 Arquitetura do Modelo do Sistema Portfólio-Tutor proposto

O sistema portfólio-tutor proposto mostra na sua arquitetura todos os componentes organizados e representados pelas seguintes camadas: camada de apresentação (que possui os

componentes Interface, Administrador e Usuário), a camada portfólio (que possui os componentes Atividades, Disciplina e Material), a camada Tutor (que possui os componentes Controle, Perfil do Aluno, Estratégia Didática e Base de Domínio) e camada Serviços onde temos os componentes de Conexão e Autenticação (Figura 4.5).

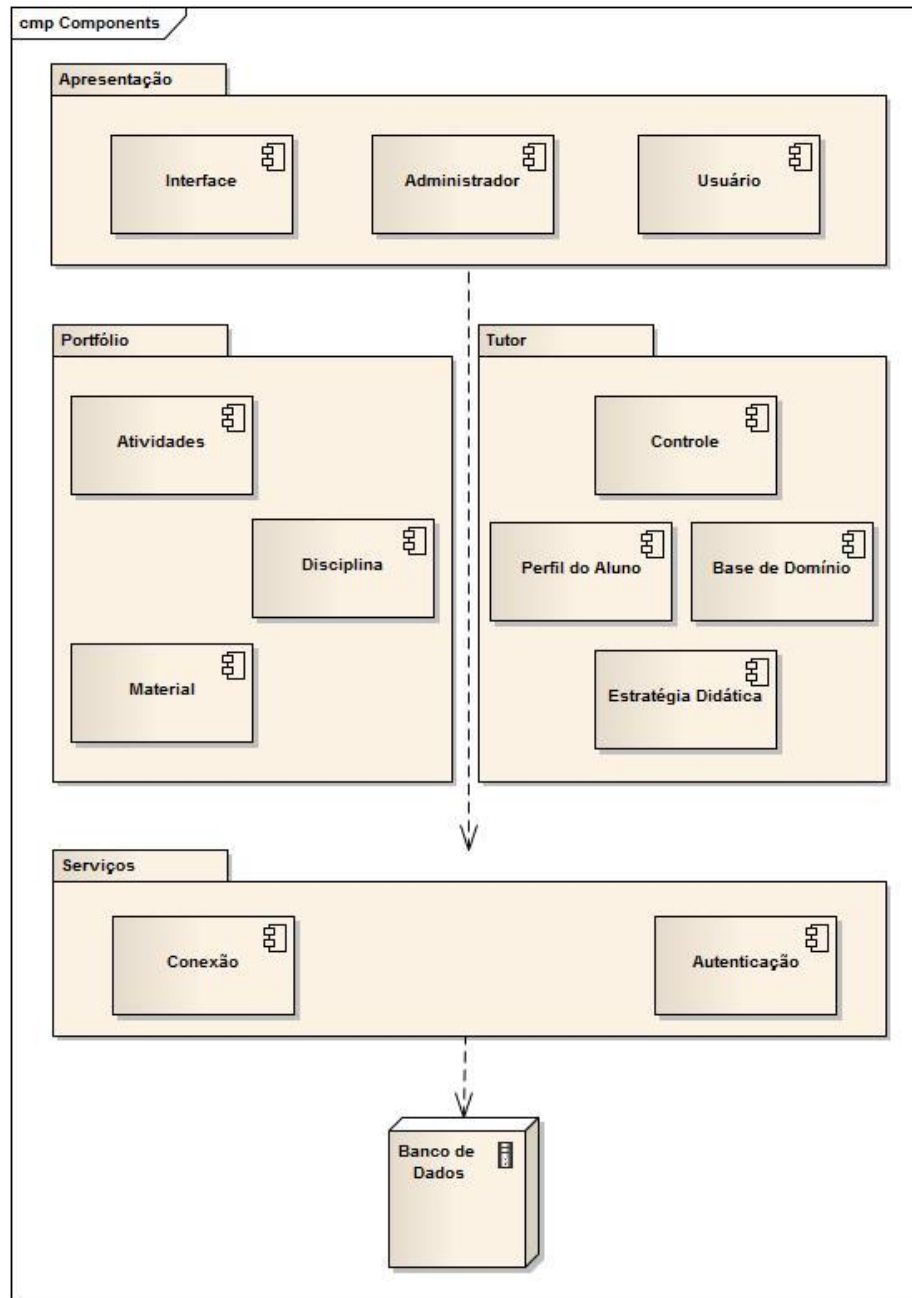


Figura 4.6: Arquitetura do Sistema Portfólio-Tutor Baseado em Componentes

4.5 Como Funciona uma Aplicação Portfólio-Tutor

A primeira coisa que o professor deverá fazer quando quiser utilizar um portfólio-tutor, é entrar em contato com o administrador do sistema e fazer uma solicitação formal para que o mesmo possa cadastrá-lo. Logo em seguida o administrador do sistema fará uma análise e autorizará ou não o cadastro do professor no sistema. No caso de afirmação de cadastro o professor passará para o administrador toda a especificação do curso, onde também será responsável pelo cadastramento dos alunos que participarão do curso. O professor definirá os artefatos e os respectivos itens de avaliação que serão aplicados pela instituição de ensino.

Com a base de domínio preparada e definida uma turma para um determinado professor, o mesmo poderá realizar o seu planejamento didático, definindo as unidades, sessões de ensino e atividades. Para cada unidade serão definidos os conceitos a serem transmitidos de acordo com os níveis básico, intermediário e avançados.

Após o cadastramento dos alunos, será enviada por e-mail a notificação da confirmação, onde todos deverão participar de uma sessão de ensino, sendo que o aluno utilizará os agentes interfaces de sessão para participar.

Serão disponibilizadas sessões de ensino síncronas e assíncronas, de acordo com cada atribuição dada ao grupo, sendo que nas sessões síncronas o controle da seqüência dos conceitos e recursos didáticos a serem aplicados é do sistema, enquanto que nas sessões assíncronas, o aluno tem a livre escolha de qual conceito estudar naquele momento e qual o recurso didático que fará uso.

As atividades poderão ser atribuídas a qualquer hora que o professor queira, e os prazos serão acompanhados pelo sistema automaticamente. Quando existe uma determinada atividade e o aluno a faz no prazo estipulado, o mesmo deverá informar e disponibilizar para o sistema, onde o professor receberá notificação automática de uma atividade a ser avaliada.

4.5.1 Diagrama de Componentes

Na figura 4.6 mostra-se a representação da estrutura do sistema, sendo possível se ver o comportamento através da interação dos componentes. Os componentes foram definidos levando em consideração as entidades em comum identificadas na caracterização do domínio.

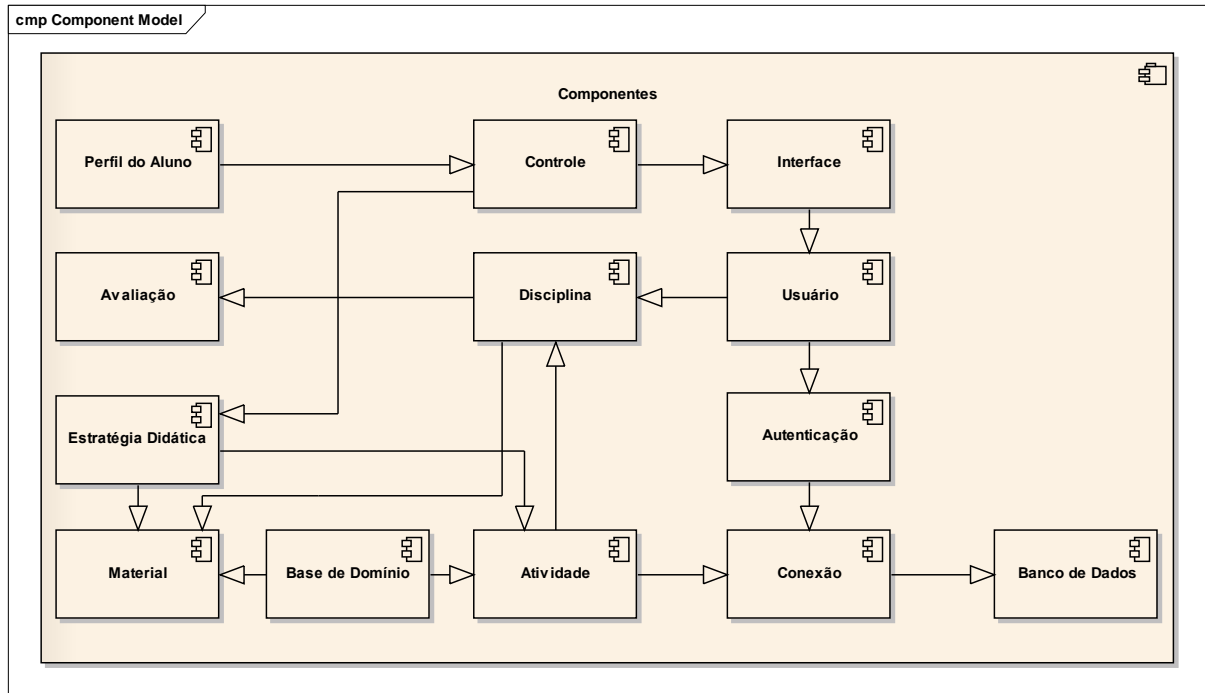


Figura 4.7: Diagrama de Componentes

4.6 Especificação dos Componentes

Para a especificação dos componentes será utilizado os diagramas de classe, onde é mostrado um conjunto de classes, interfaces e colaboração e seus relacionamentos. São utilizados para fazer a modelagem da visão estática do projeto do sistema (Larman 1999).

4.6.1 Diagrama de Classe do Componente Usuário

A figura 4.7 mostra o diagrama de classe do componente usuário. A classe principal é a classe Usuário, três subclasses da classe usuário foram definidas: a classe professor, administrador e aluno, onde essa ultima se relaciona com a classe monitor. A classe Usuário está associada à classe Controlador que implementa os serviços da Interface ControleInterface.

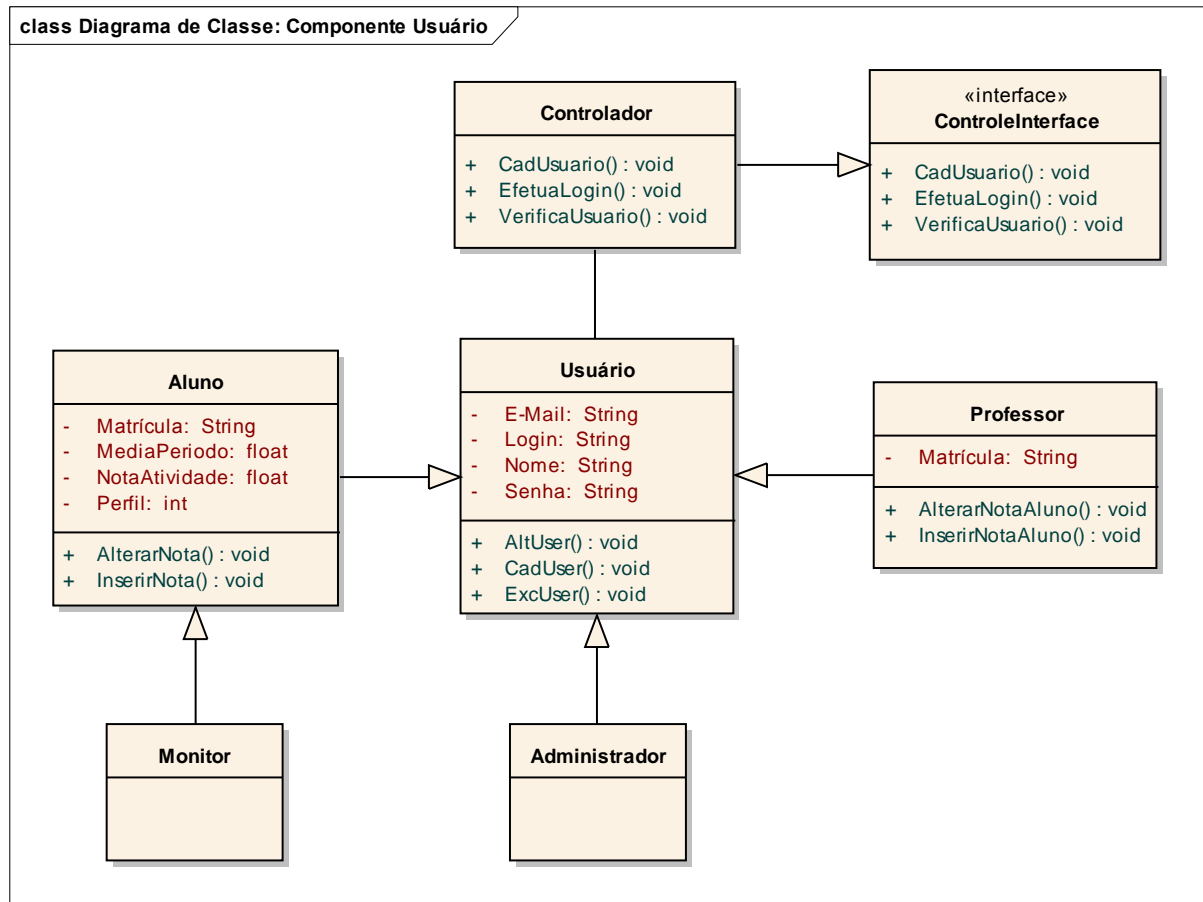


Figura 4.8: Diagrama de Classes do Componente Usuário

4.6.2 Diagrama de Classes do Componente Atividades

A figura 4.8 mostra o diagrama de classe do componente Atividades. A classe principal é a classe *Atividade*, todas as formas de atividades possíveis, tais como, *AvaliacaoTradicional*, *Seminario*, *Projeto*, *ListaExercicio* foram definidas como subclasses da classe *Atividade*. Todas as classes estão associadas à classe *Controlador* que implementa os serviços da *InterfaceControle*.

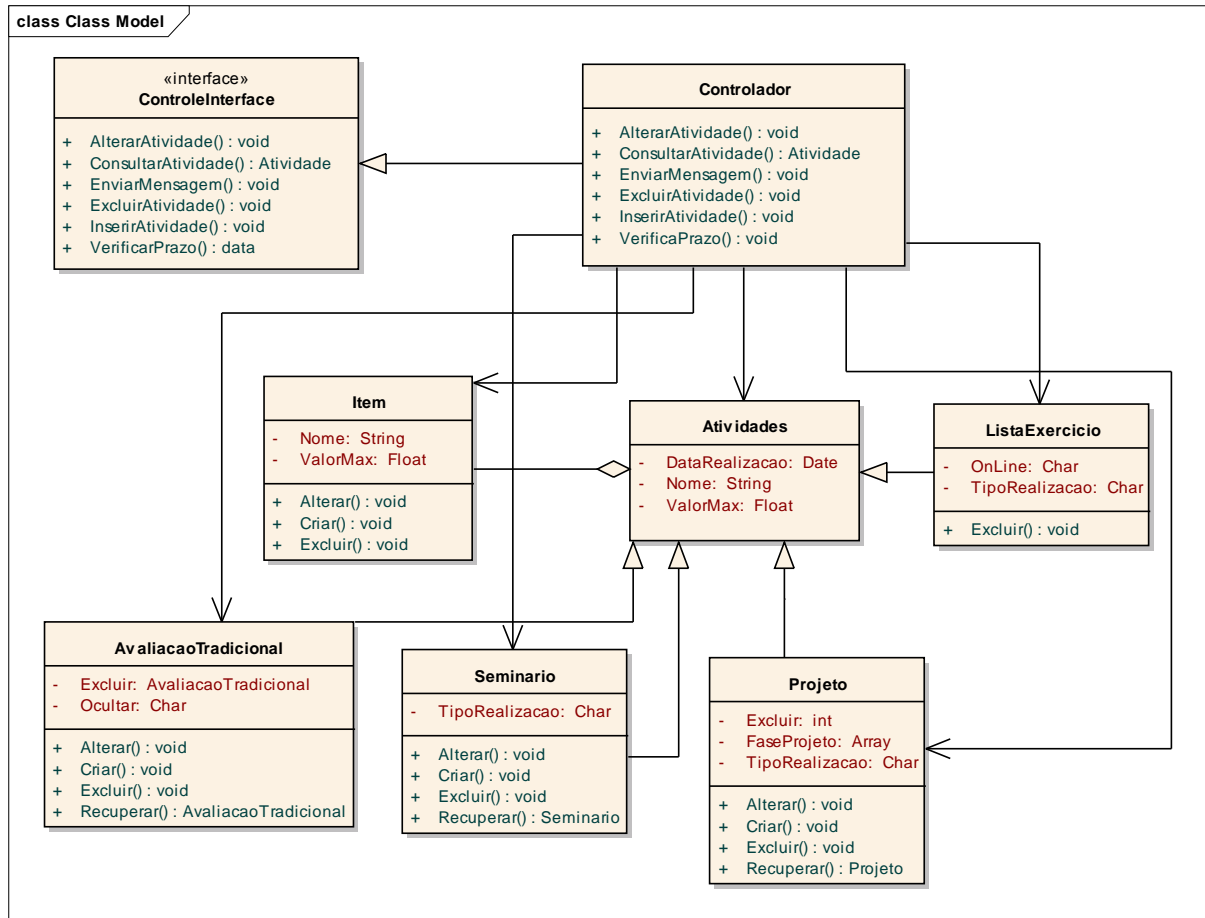


Figura 4.9: Diagrama de Classes do Componente Atividades

4.6.3 Diagrama de Classes do Componente Disciplina

A figura 4.9 mostra o diagrama de classe do componente Disciplina. A classe principal é a classe Disciplina, que está relacionada com Professor e Aluno. A classes principal está associada à classe Controlador que implementa os serviços da InterfaceControle.

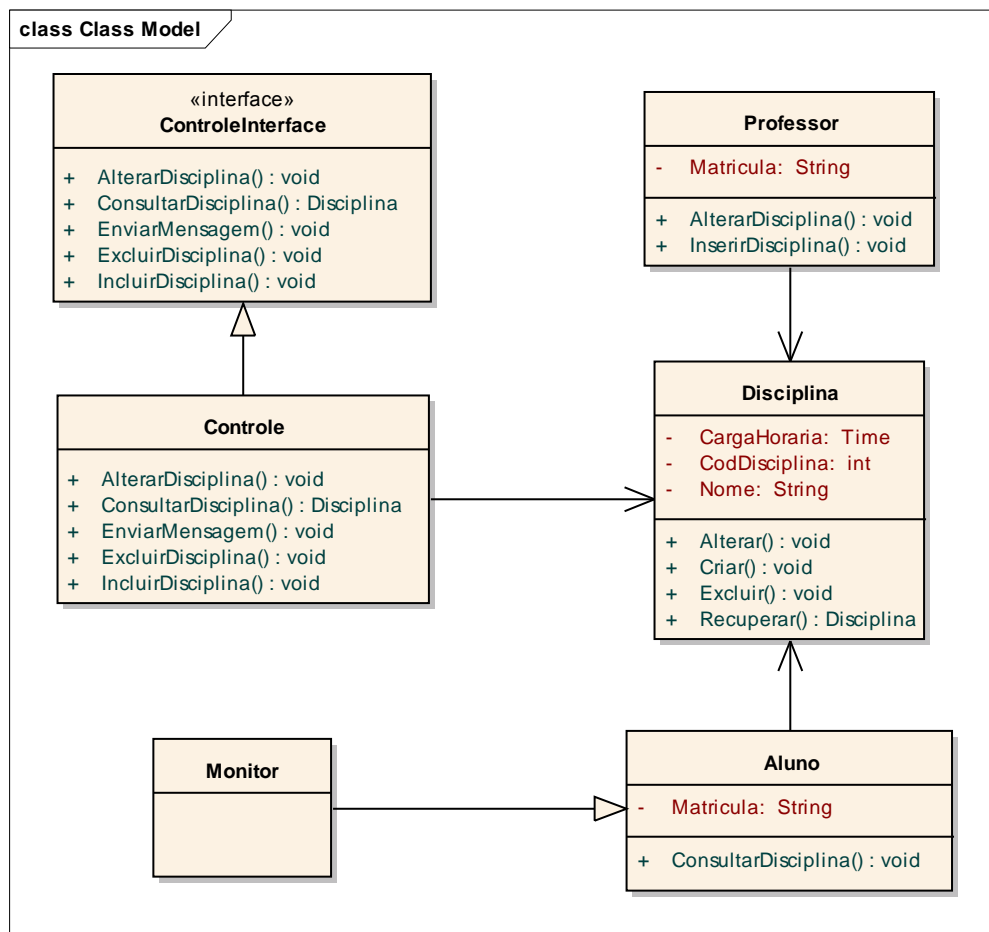


Figura 4.10: Diagrama de Classes do Componente Disciplina

4.6.4 Diagrama de Classes do Componente Avaliação

A figura 4.10 mostra o diagrama de classe do componente Avaliação. A classe principal é a classe Avaliação, que está associada com AvaliacaoTradicional, Seminario, Projeto e ListaEercicio. A classe principal está associada à classe Controle que implementa os serviços da InterfaceControle.

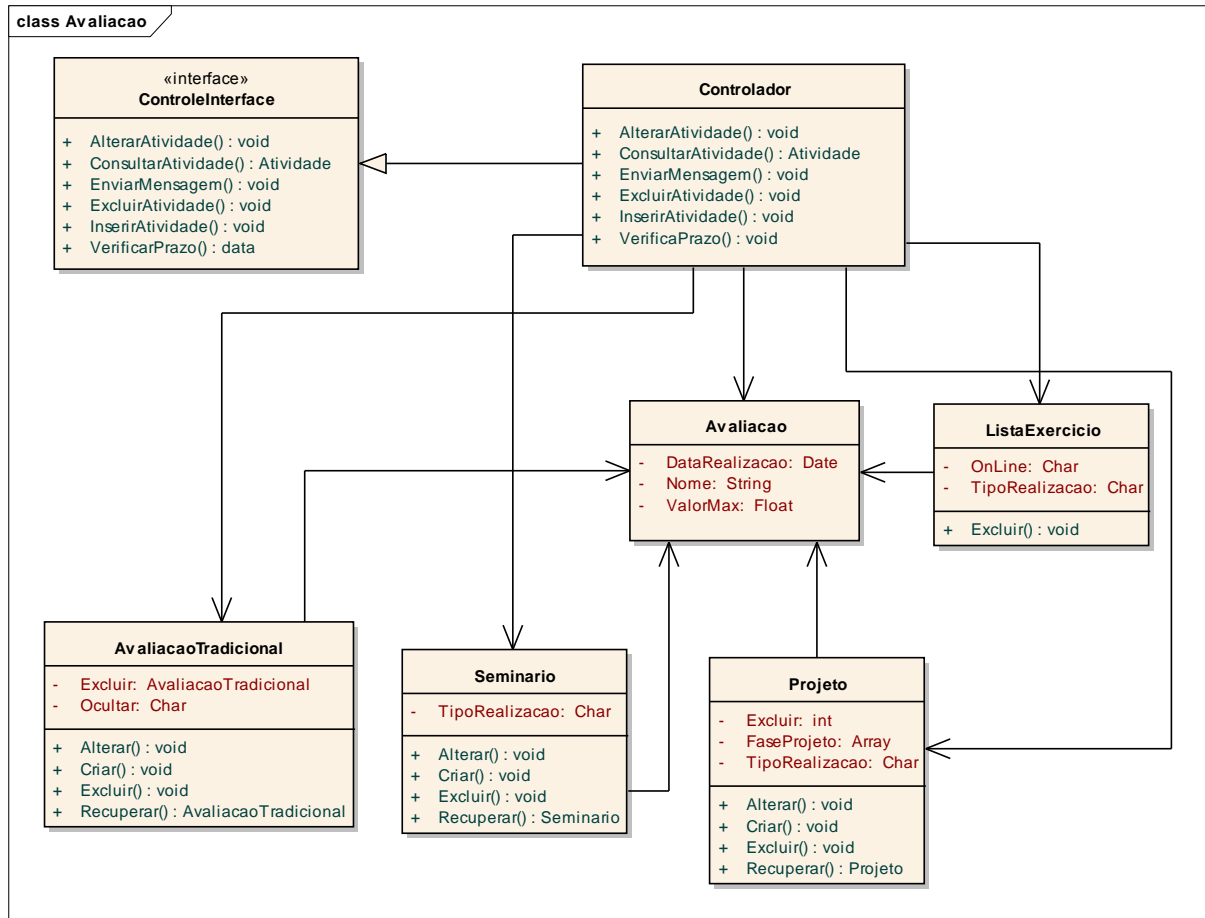


Figura 4.11: Diagrama de Classes do Componente Avaliação

4.7 Resumo

Através dos modelos elaborados é possível se ter uma visão do sistema proposto em duas perspectivas, a estrutural e a comportamental.

Foram apresentados os diagramas de casos de uso, a descrição dos casos de uso, a arquitetura do sistema portfólio-tutor baseado em componentes e o diagrama de classes de vários componentes.

Esses modelos em conjunto capturam as decisões mais importantes do sistema proposto, e servirão de base para o desenvolvimento do sistema proposto, detalhado no próximo capítulo.

5 IMPLEMENTAÇÃO DO SISTEMA

No capítulo 5 serão mostrados os elementos técnicos da implementação e a parte funcional do sistema será representada de forma detalhada através das telas do sistema.

5.1 Tecnologias Utilizadas na Implementação

O protótipo do sistema portfólio-tutor baseado no reuso de software foi desenvolvido com o uso da plataforma Delphi 7, onde a definição da interface e até mesmo de parte da estrutura de um aplicativo pode ser realizada com o auxílio de ferramentas visuais, facilitando e agilizando o desenvolvimento do sistema proposto. É uma linguagem orientada a objetos e orientada a eventos, abrangendo os conceitos de classe, herança e polimorfismo. O banco de dados utilizado foi o Firebird versão 2.1, ele tem código aberto e não precisa de licença para sua utilização. Abaixo podemos observar algumas características desse SGBD:

- Suporte total a Stored Procedure e Triggers;
- Integridade Referencial;
- Consome poucos recursos de processamento;
- Possui diversas formas de acesso ao banco de dados: Nativo/API, dbexpress, ODBC, .Net Provider, JDBC Nativo, etc.

5.2 Disposição Física dos Módulos do Sistema

A tecnologia utilizada para o desenvolvimento do protótipo do Sistema Portfólio-Tutor Baseada no Reuso de Software possibilita a disposição física dos seus módulos em três camadas:

- **Camada Interface:** é identificada como a camada de apresentação e responsável pela interação dos usuários.

- **Camada Lógica de Negócios:** controla as regras de negócio e determina como processar, o que vai persistir e o que vai recuperar.
- **Camada de Persistência:** controla a persistência dos dados, que são presentes temporariamente nas instâncias das classes e são armazenados em um meio físico permanente.



Figura 5.1: Disposição Física em Camadas

5.3 Estudo de Caso

Devido a necessidade de um melhor controle e o acompanhamento do desempenho dos alunos nas atividades de aprendizagem, foi desenvolvida uma aplicação que atendesse a maioria desses requisitos.

As funcionalidades da camada Tutor são especificadas pelo professor, onde o mesmo cria uma sessão de ensino, e o sistema irá informar a criação dessa sessão de ensino, onde o professor poderá criar uma ou várias estratégias didáticas, ou até mesmo utilizar uma estratégia didática existente, que são compostas por táticas de ensino.

5.3.1 Interface do Sistema Portfólio-Tutor Baseado no Reuso de Software

Nesta sessão será mostrado o funcionamento do sistema proposto, detalhando todas as suas interfaces e funcionamento das mesmas.

A Figura 5.2 mostra a tela que representa o início da utilização do sistema proposto.



Figura 5.2: Tela Inicial do Sistema Proposto

Ao iniciar o programa será solicitado um usuário e senha válidos para poder acessar os módulos do sistema, onde os mesmos serão carregados de acordo com o perfil de cada usuário cadastrado. Nesse caso iremos iniciar a sessão com o usuário administrador, pois esse terá controle total de todos os módulos do sistema.



Figura 5.3: Tela Principal do Sistema

Todo usuário após se logar no sistema terá a opção de alterar sua senha por questões de segurança. O usuário digitará a senha atual e depois colocará sua nova senha e confirmará a alteração pressionando o botão Gravar. Para verificar se as alterações foram feitas com sucesso, basta efetuar Logoff no sistema e se Logar novamente, digitando seu usuário e sua nova senha.

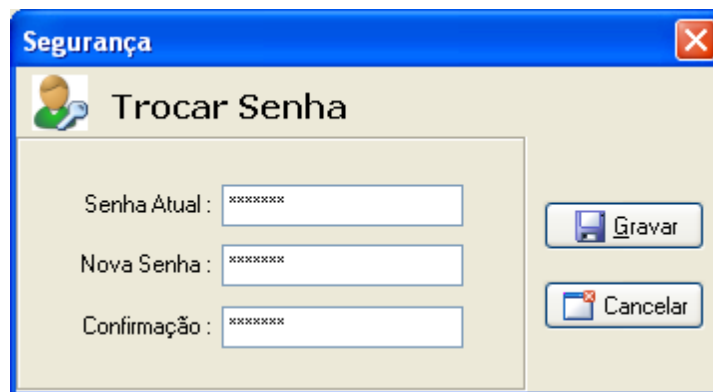
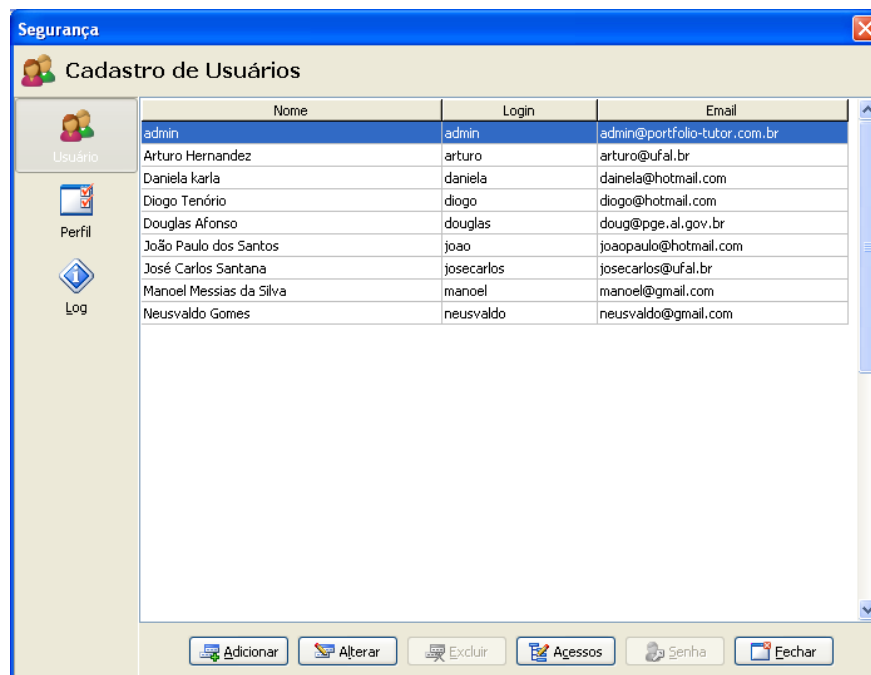


Figura 5.4: Tela para Mudança de Senha

5.3.1.1 Interface do Administrador

Nesta sessão serão mostradas as telas quando logado com o perfil de administrador.



	Nome	Login	Email
admin	admin	admin	admin@portfolio-tutor.com.br
Usuário	Arturo Hernandez	arturo	arturo@ufal.br
	Daniela Karla	daniela	dainela@hotmail.com
	Diogo Tenório	diogo	diogo@hotmail.com
Perfil	Douglas Afonso	douglas	doug@pge.al.gov.br
	João Paulo dos Santos	joao	joaopaulo@hotmail.com
	José Carlos Santana	josecarlos	josecarlos@ufal.br
Log	Manoel Messias da Silva	manoel	manoel@gmail.com
	Neusvaldo Gomes	neusvaldo	neusvaldo@gmail.com

Figura 5.5: Cadastro de Usuários (Informações Básicas dos Usuários)

É nesta tela (Figura 5.5) que o usuário administrador terá acesso a todas as informações relativas ao cadastro de usuário, perfil e permissões de acesso. Podemos observar que ao lado esquerdo da tela temos três botões, onde o primeiro representa as informações básicas que são mostradas de cada usuário cadastrado, o segundo representa o perfil do usuário (Figura 5.6), e o terceiro o Log de acesso, ou seja, o que cada usuário acessou no sistema em uma determinada data (Figura 5.7).

Ainda na Figura 5.5, podemos observar também na parte inferior, todas as ações que o usuário administrador poderá ter, ou seja, adicionar um novo usuário, alterar as informações desse usuário, excluir um usuário existente, alterar a senha ou fechar essa tela.

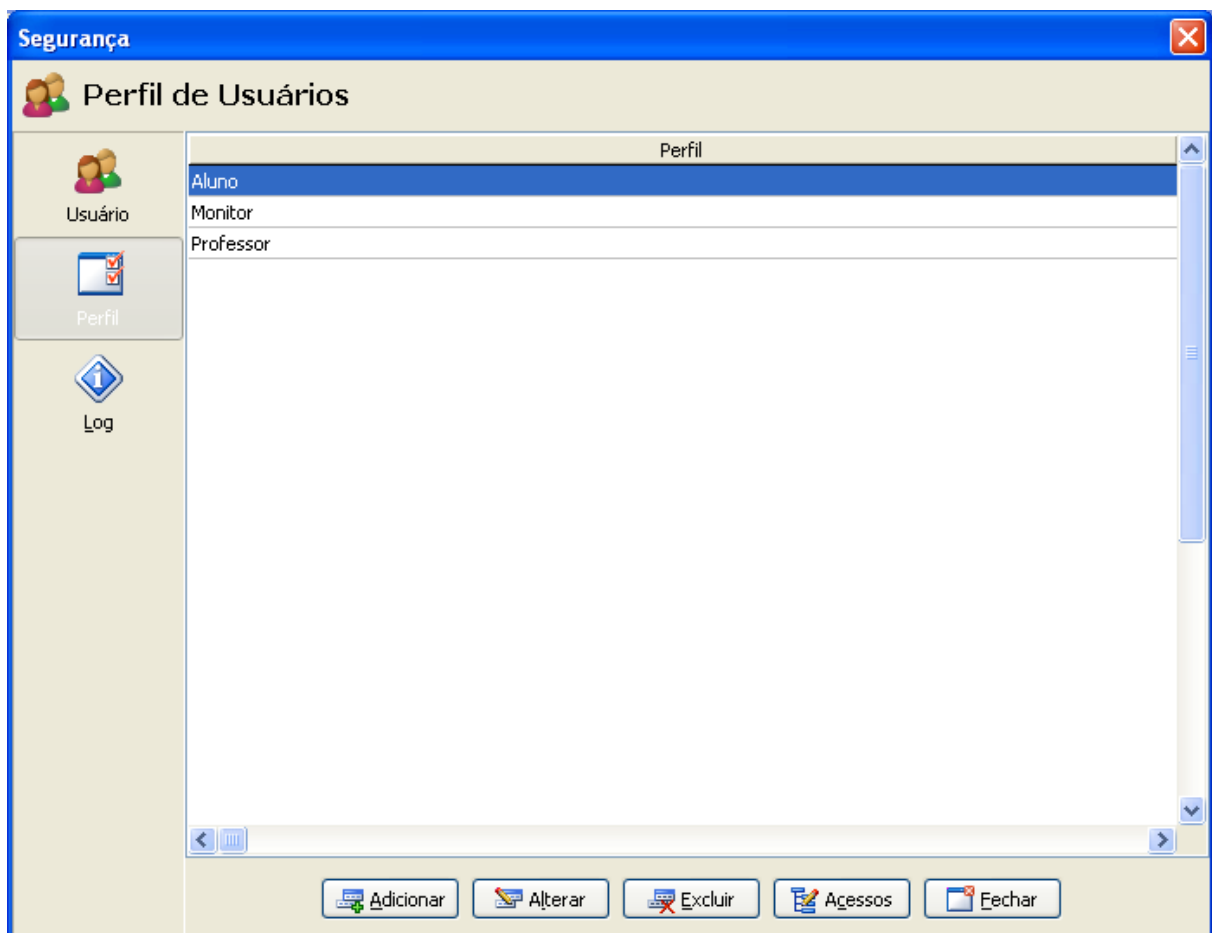


Figura 5.6: Cadastro de Usuários (Perfil dos Usuários)

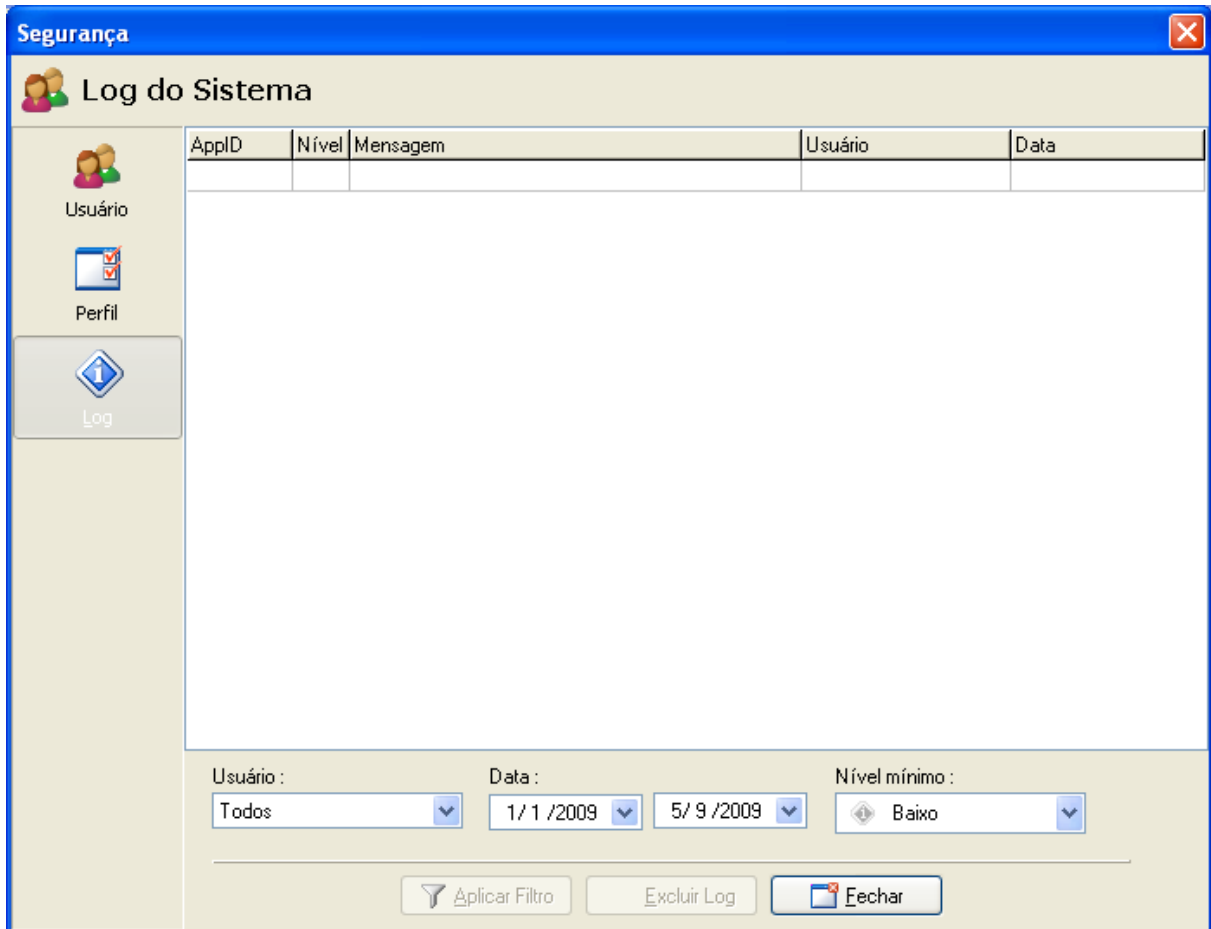


Figura 5.7: Cadastro de Usuários (Log de Acesso)

O usuário administrador deverá primeiramente criar os perfis para posteriormente cadastrar os usuários e atribuir a cada um deles seus respectivos perfis.



Figura 5.8: Perfil de Usuários

Feito o cadastro do perfil, o próximo passo é o cadastro do usuário com sua respectiva atribuição de perfil. No exemplo a seguir foi feito o cadastro de um usuário com o perfil de Aluno.

Cadastro de Usuários

Alterar Usuário

Nome : Douglas Afonso

Login : douglas

Email : doug@pge.al.gov.br

Status : Ativo

Perfil : Aluno

Expira em : 10 Dias

Senha do usuário não expira

Gravar

Cancelar

Figura 5.9: Cadastro de Usuários

Efetuamos o cadastro do usuário informando o Nome, Login, E-mail, Status, Perfil, e determinamos se a senha do usuário poderá ter um período de validade, onde será informada a quantidade de dias que esse usuário terá acesso.

Após a finalização do cadastro dos usuários, serão dadas as devidas permissões de acessos a cada um deles.

Segurança

Permissões do Usuário : douglas

Ações | Controles

- Logoff
- Alterar Senha
- Usuários

Liberar

Bloquear

Gravar

Cancelar

Figura 5.10: Tela de Permissão de Acesso (Aba Ações)

Tomando como base o cadastro do usuário com perfil de aluno, daremos ao mesmo as permissões de acesso. Na tela de segurança temos duas abas, a de Ação e a de Controle. A primeira são as ações que cada usuário deverá ter. Nesse exemplo podemos ver que foi desmarcada a opção Usuário, pois é nessa opção que o usuário administrador terá total controle sobre os usuários cadastrados, e somente ele poderá ter esse acesso.

Na aba de Controles será dado acesso às áreas de controle proveniente a cada usuário. Na Figura 5.11 podemos observar as permissões para um usuário com perfil de Aluno.

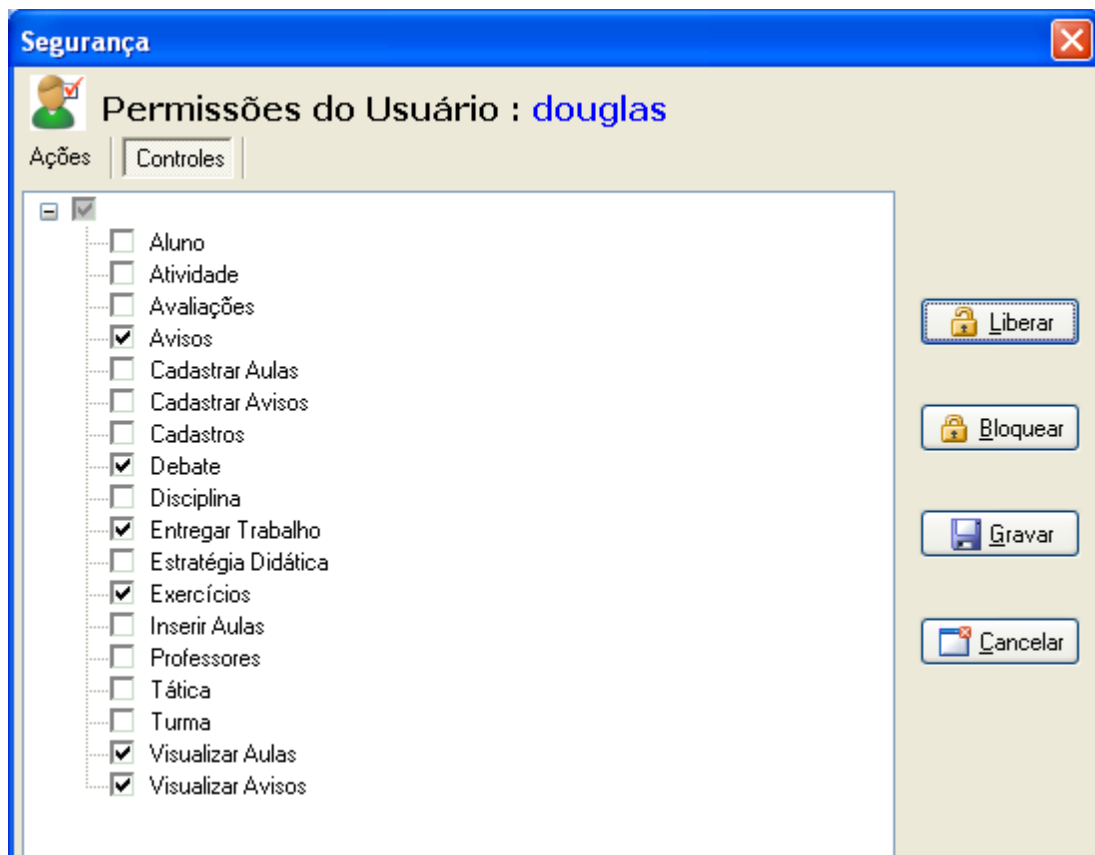


Figura 5.11: Tela de Permissão de Acesso (Aba Controles)

Na figura 5.11 serão detalhadas todas as funções relacionadas aos cadastros necessários para o início da utilização do sistema proposto. Na tela inicial temos todos os controles necessários para o funcionamento do sistema, e estão disponibilizados ao lado esquerdo da tela inicial. Ao pressionar o botão de Cadastros será habilitado um menu superior com todas as opções do mesmo. Selecionando a opção Professores, podemos observar na Figura 5.12 que abrirá uma tela de cadastro do mesmo.

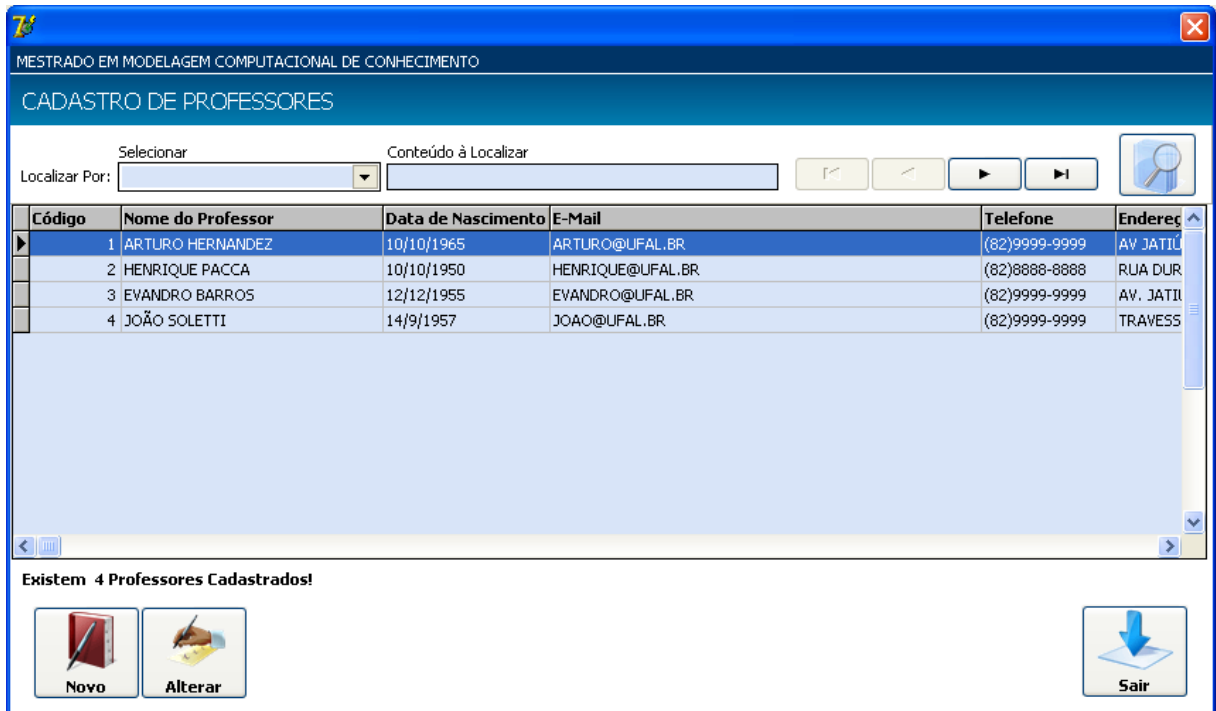


Figura 5.12: Tela de Controle de Professores

Nesta tela o usuário terá total controle de todos os professores cadastrados, podendo localizar o mesmo por Nome ou pela Turma que o mesmo está lotado. Ao ser pressionado o botão Novo, será aberta uma nova janela correspondente ao cadastro das informações do professor.

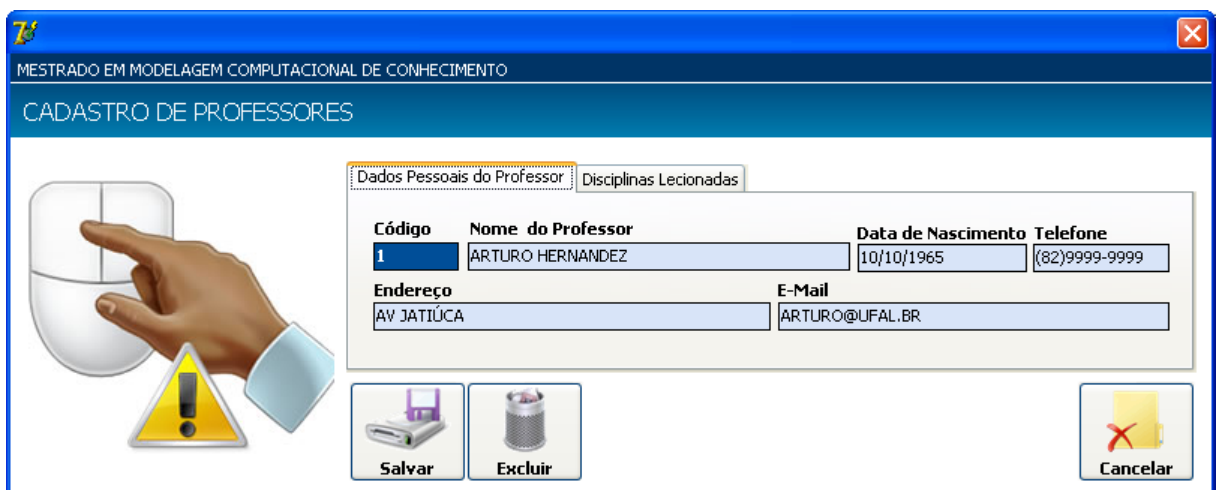


Figura 5.13: Cadastro de Professores (Dados Pessoais)

Podemos observar que nessa tela temos duas abas, uma com os dados pessoais do professor (Figura 5.13) e outra com as disciplinas lecionadas por esse professor que será

cadastrado (Figura 5.14). Primeiramente iremos informar os dados pessoais do professor que se encontram na primeira Aba.

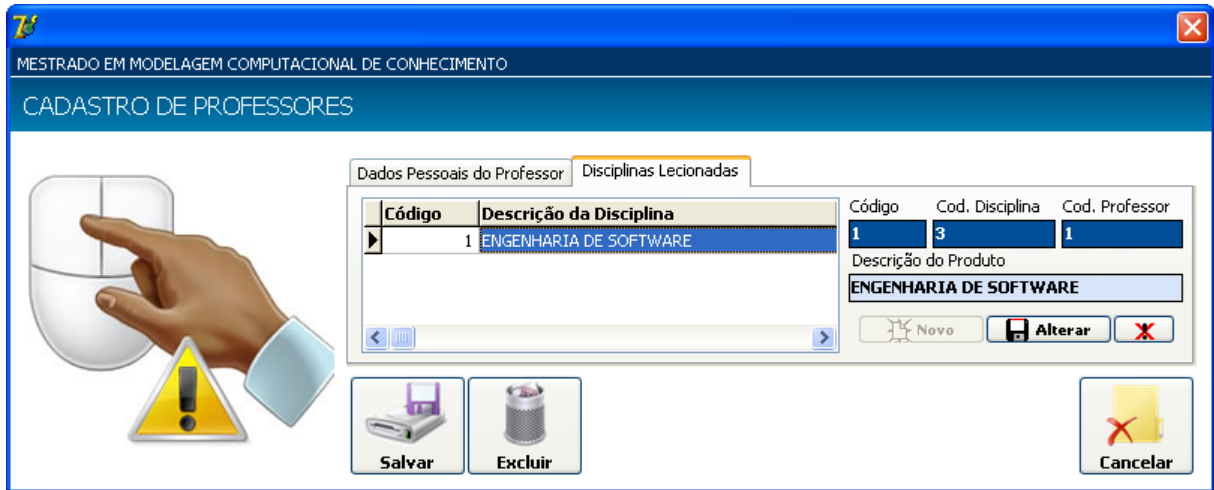


Figura 5.14: Cadastro de Professores (Disciplinas Lecionadas)

Na segunda aba iremos adicionar as disciplinas que este professor irá lecionar. Para isto basta pressionar o botão Novo e aparecerá uma lista de disciplinas cadastradas no sistema (Figura 5.15).

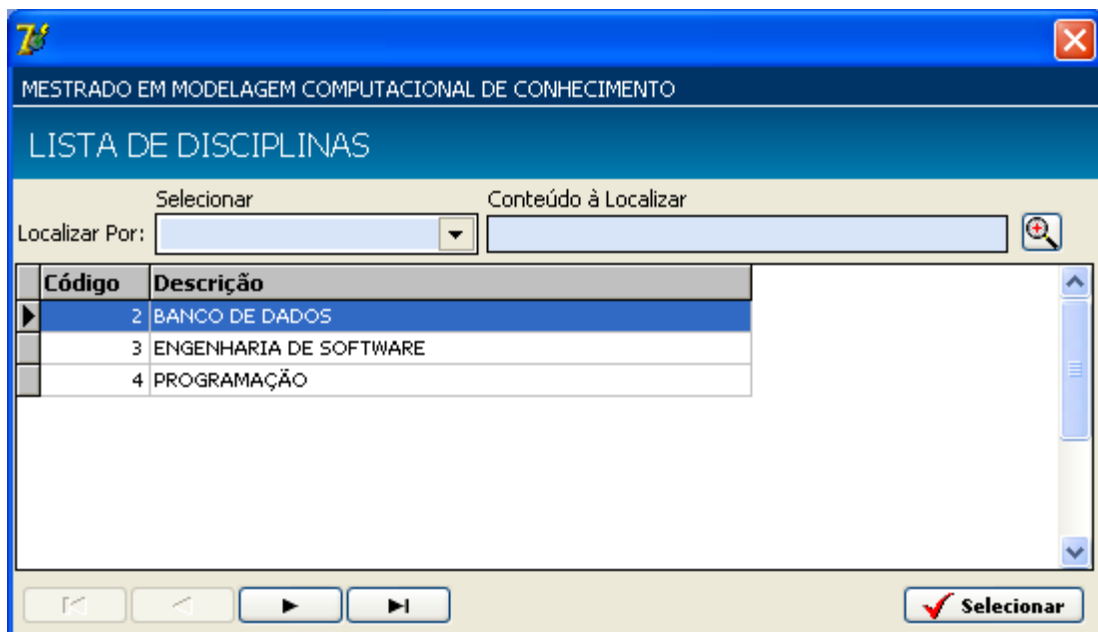


Figura 5.15: Lista de Disciplinas Cadastradas no Sistema

Localize a disciplina que deverá ser adicionada ao cadastro do professor e pressione o botão selecionar, onde serão inseridas as informações necessárias para o cadastro. Pressione o

botão salvar e a disciplina estará anexada aos dados do respectivo professor. Se desejar fazer alguma alteração nas informações inseridas, basta pressionar o botão Alterar que serão carregadas todas as informações referentes ao professor.

O próximo cadastro a ser detalhado é o de Alunos, e o processo utilizado é o mesmo para o cadastro de Professores.

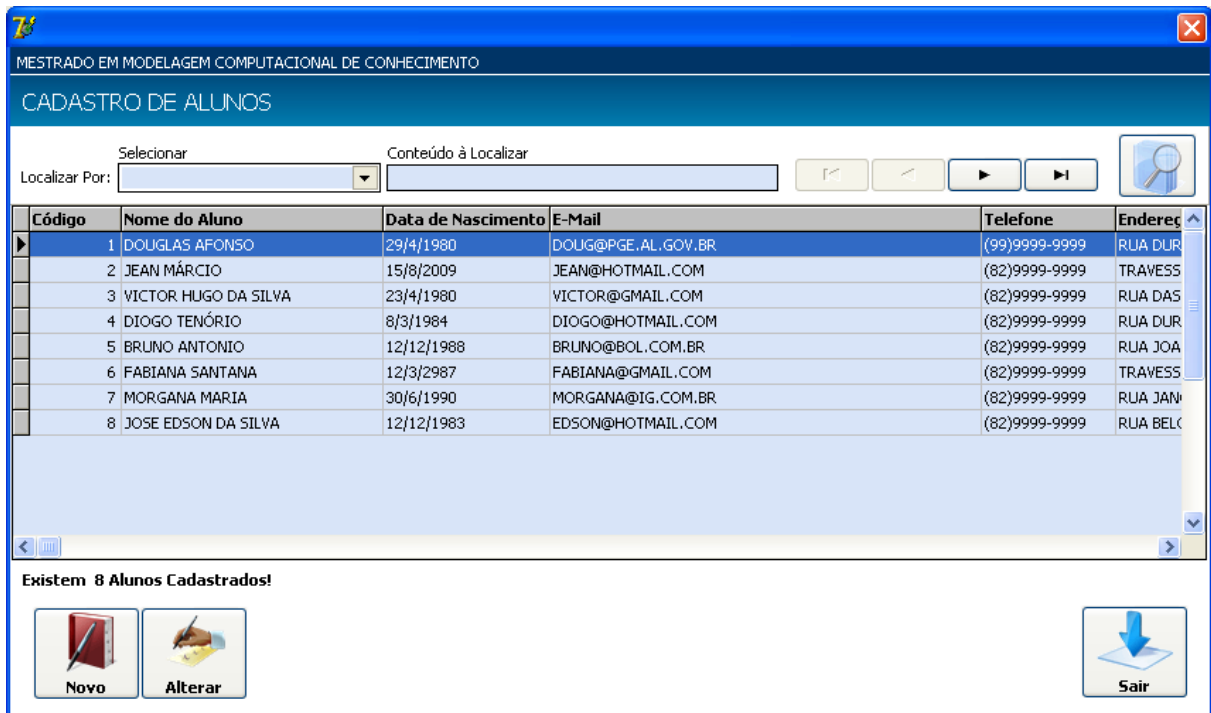


Figura 5.16: Tela de Controle de Alunos

Ao ser pressionado o botão Novo (Figura 5.16), será aberta uma nova janela correspondente ao cadastro das informações do Aluno Figura 5.17.

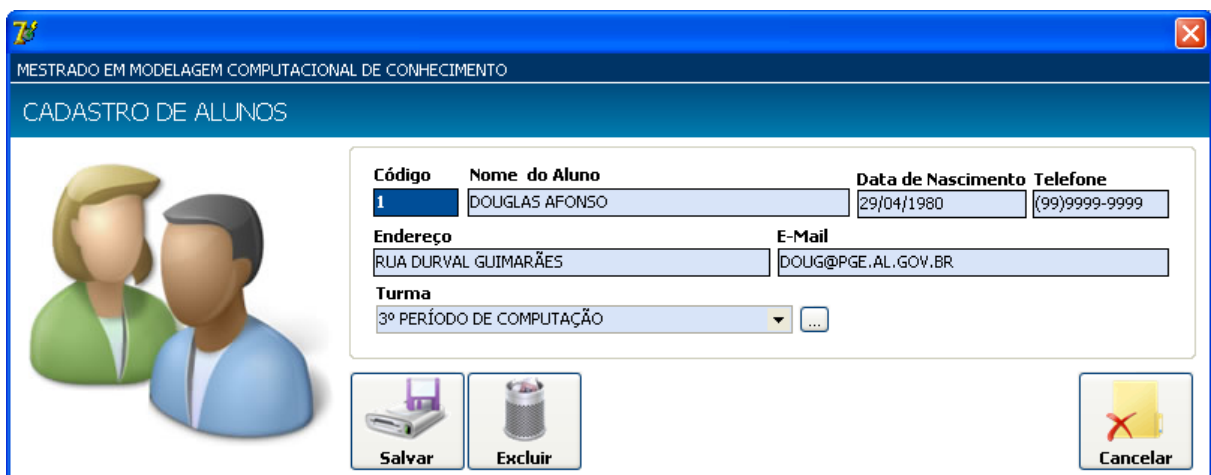


Figura 5.17: Cadastro de Alunos

No espaço reservado para o preenchimento da informação relativa à turma, o usuário tem a opção de selecionar uma turma já cadastrada, caso não exista a turma na lista que será carregada, o mesmo poderá fazer o cadastro pressionando o botão ao lado, onde será carregada uma nova tela de cadastro relativo à turma (Figura 5.18).

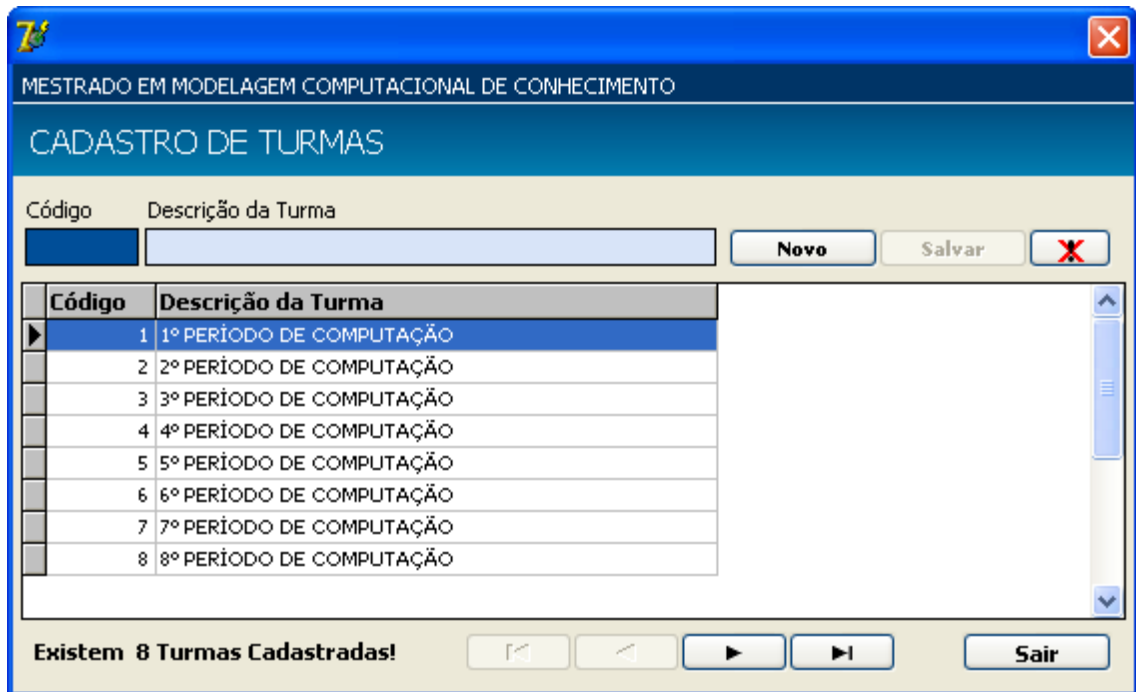


Figura 5.18: Tela de Cadastro de Turmas

Feito os cadastros e concedidas às respectivas permissões de acesso, o sistema estará configurado e pronto para sua utilização.

5.3.1.2 Interface do Professor

A Figura 5.19 representa a tela inicial quando logado com o perfil de Professor.



Figura 5.19: Tela inicial para o usuário com o perfil de Professor

Ao lado esquerdo da tela inicial podemos observar todos os botões de controle do sistema, onde será expandido um menu superior com várias funções de acordo com cada perfil logado. Nessa tela podemos observar que após o acionamento do botão Cadastros foi habilitado um menu superior com as opções de Cadastro de Professores, Alunos, Disciplina, Turma, Atividade, Tática, Estratégia Didática e Sessão de Ensino, onde só ficou disponibilizado para o usuário de perfil professor as últimas três opções de cadastro, devido às suas restrições de acesso, que foram dadas anteriormente no momento do cadastro dos usuários e seus respectivos perfis. A mesma coisa ocorre para todos os outros botões, habilitando apenas aqueles que lhe foi dado permissão.

A seguir serão detalhadas todas as funções relacionadas aos cadastros que estão disponíveis para o Professor.

5.3.1.2.1 Cadastro de Atividades

Esta é a tela inicial que o professor terá para o início do cadastro e a visualização de todas as atividades que o mesmo criou. Ao se pressionar o botão Novo, iremos para a tela de cadastro da atividade (Figura 5.21), onde serão inseridas todas as informações necessárias.

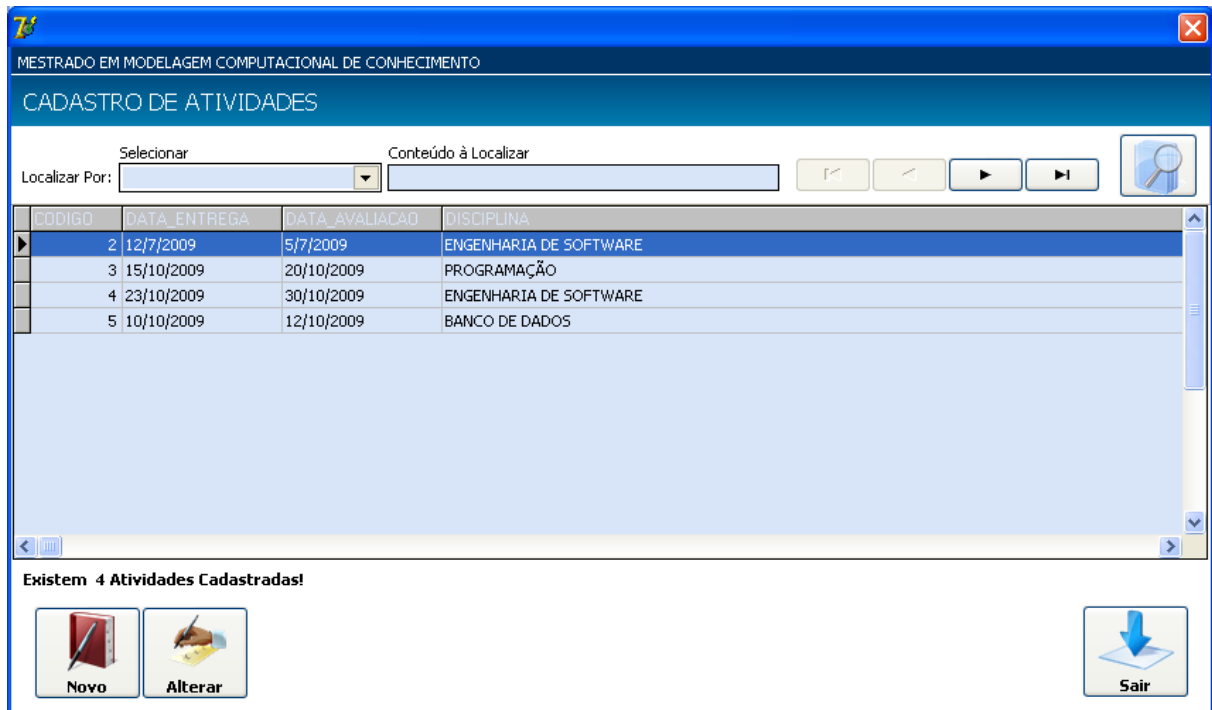


Figura 5.20: Tela de Controle de Atividades

Professor: Arturo Hernandez

Código: 3 Data de Entrega: 15/10/2009 Data da Avaliação: 20/10/2009 Turma: 1º PERÍODO DE COMPUTAÇÃO

Título da Atividade: ATIVIDADE DE PROGRAMAÇÃO

Descrição da Atividade: FAÇA UM PROGRAMA QUE RECEBA O NOME DE UM ALUNO, TRÊS NOTAS E CALCULE A MÉDIA ARITMÉTICA.

Caminho do Arquivo: Procurar... Disciplina: PROGRAMAÇÃO

Buttons: Salvar, Excluir, Cancelar

Figura 5.21: Tela de Cadastro de Atividade

Nesta tela o professor irá informar a data que será disponibilizada a atividade e a data de entrega da mesma, depois deverá selecionar a turma à quem será disponibilizada a atividade, informar o título e a descrição da mesma, depois deverá selecionar uma das disciplinas que é lecionada. O professor também terá uma opção de anexar um arquivo a essa atividade.

5.3.1.2.2 Cadastro de Táticas

Na Figura 5.22 podemos observar a tela responsável pelo cadastro das táticas que serão utilizadas pelo professor na sessão de ensino posteriormente criada. Sua interface é de fácil compreensão, basta pressionar o botão novo e no campo descrição, informar a tática a ser cadastrada. No Grid em destaque, podemos observar todas as táticas já inseridas. Para alterar alguma informação referente à tática, basta dar um duplo click na informação desejada e fazer as alterações necessárias.

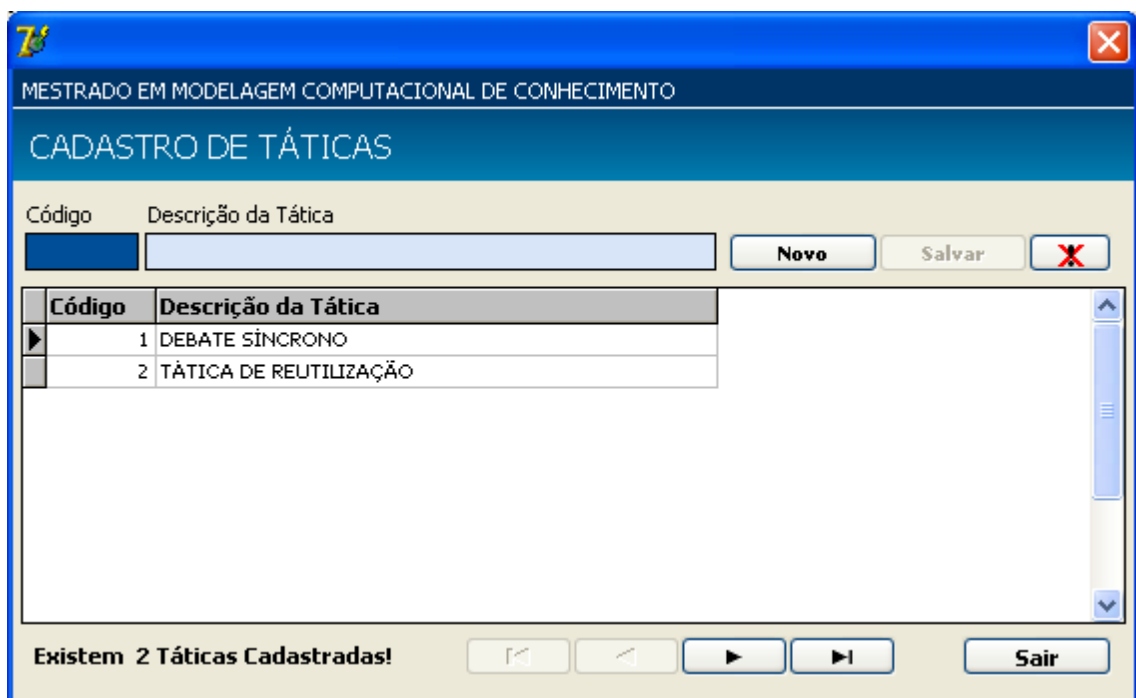


Figura 5.22: Tela de Cadastro de Táticas

5.3.1.2.3 Cadastro de Estratégias Didáticas

Quando o professor pressiona no link de Cadastro das Estratégias Didáticas, ele é direcionado para a tela de Controle de Estratégia (Figura 5.23). Nesta tela o professor poderá visualizar todas as informações referentes às Estratégias através de consultas especializadas. Ele também terá o total controle das informações, tanto em relação ao cadastro como em relação a alteração das informações.

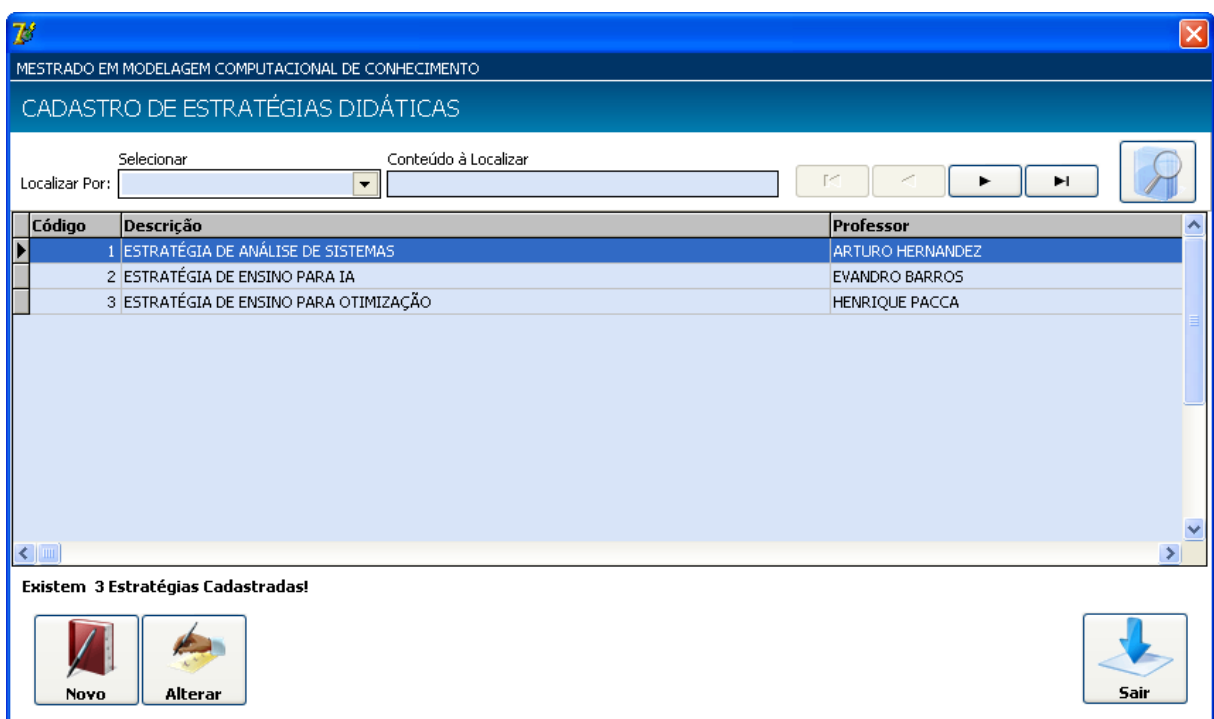


Figura 5.23: Tela de Controle das Estratégias Didáticas

Quando pressionado o botão novo, será aberta uma nova tela, que é a responsável pelo cadastro das informações referentes às Estratégias Didáticas (Figura 5.24) que o professor irá utilizar em uma determinada sessão de ensino.

MESTRADO EM MODELAGEM COMPUTACIONAL DE CONHECIMENTO

CADASTRO DE ESTRATÉGIA DIDÁTICA

Código: 1 | Descrição da Estratégia: ESTRATÉGIA DE ANÁLISE DE SISTEMAS | Professor: HENRIQUE PACCA

Código: 3 | Cód. Estrat.: 1 | Descrição da Tática: TÁTICA DE REUTILIZAÇÃO | Tempo: 00:30:00

Código	Cód. Estratégia	Descrição	Tempo
1	1	DEBATE SÍNCRONO	16/9/2009 00:10:00
2	1	TÁTICA DE REUTILIZAÇÃO	4/10/2009 00:30:00

Buttons: Salvar, Excluir, Cancelar

Figura 5.24: Tela de Cadastro das Estratégias Didáticas

Nesta tela o professor irá informar todos os dados referentes ao cadastro das Estratégias Didáticas. A princípio, ele irá informar a descrição da estratégia, e a informação referente ao professor, pois a mesma será preenchida automaticamente, devido às informações carregadas quando feito o login no sistema. O próximo passo é inserir quais as táticas que serão utilizadas para essa estratégia didática. Ao se pressionar o botão Novo, serão preenchidas automaticamente as informações referentes ao código do cadastro da tática e o código da estratégia didática, depois o professor selecionará uma tática e posteriormente informará o tempo de duração para a mesma. Depois é só pressionar o botão salvar e inserir outra tática, caso seja necessário para essa estratégia.

5.3.1.2.4 Cadastro de Sessão de Ensino

Ao ser pressionado o link de Sessão de Ensino, será aberta uma nova tela com todas as informações referentes à Sessão de Ensino. O funcionamento desta tela é basicamente o mesmo referente ao Cadastro de Estratégias Didáticas, onde o professor terá o controle das informações relativo ao cadastro e alteração das informações da Sessão de Ensino.

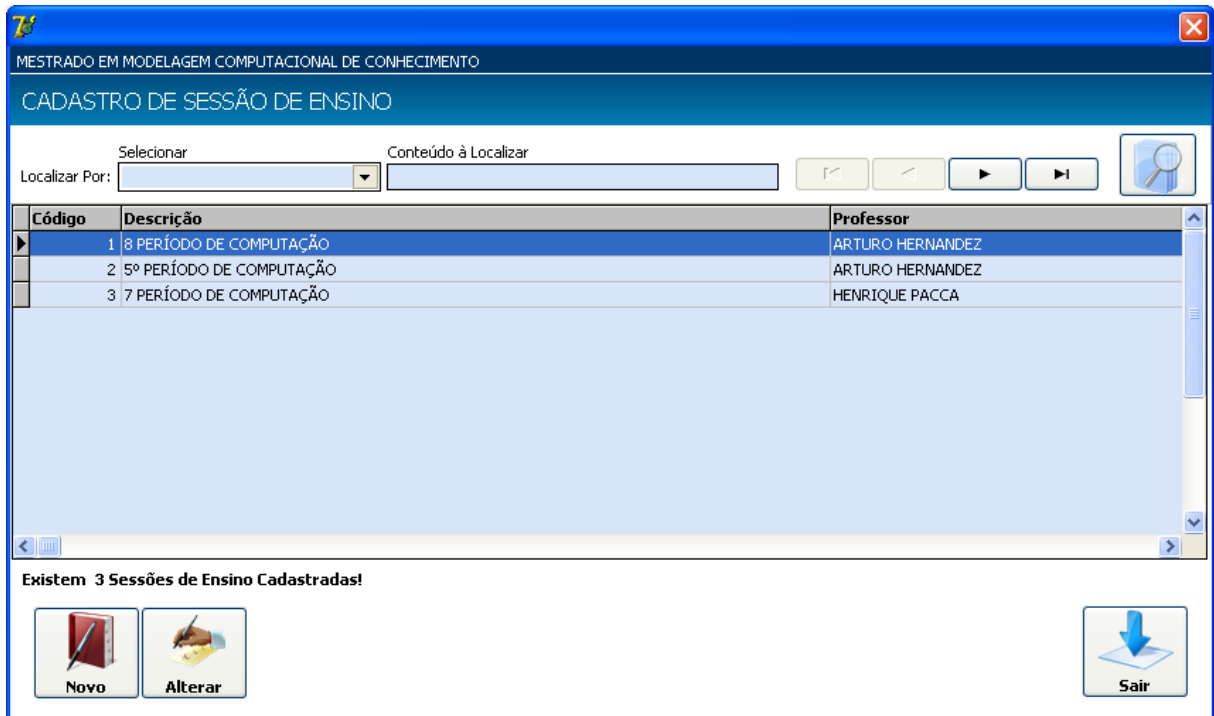


Figura 5.25: Tela de Controle das Sessões de Ensino

Quando pressionado o botão novo, será aberta uma nova tela, que é a responsável pelo cadastro das informações referentes às Sessões de Ensino (Figura 5.26).

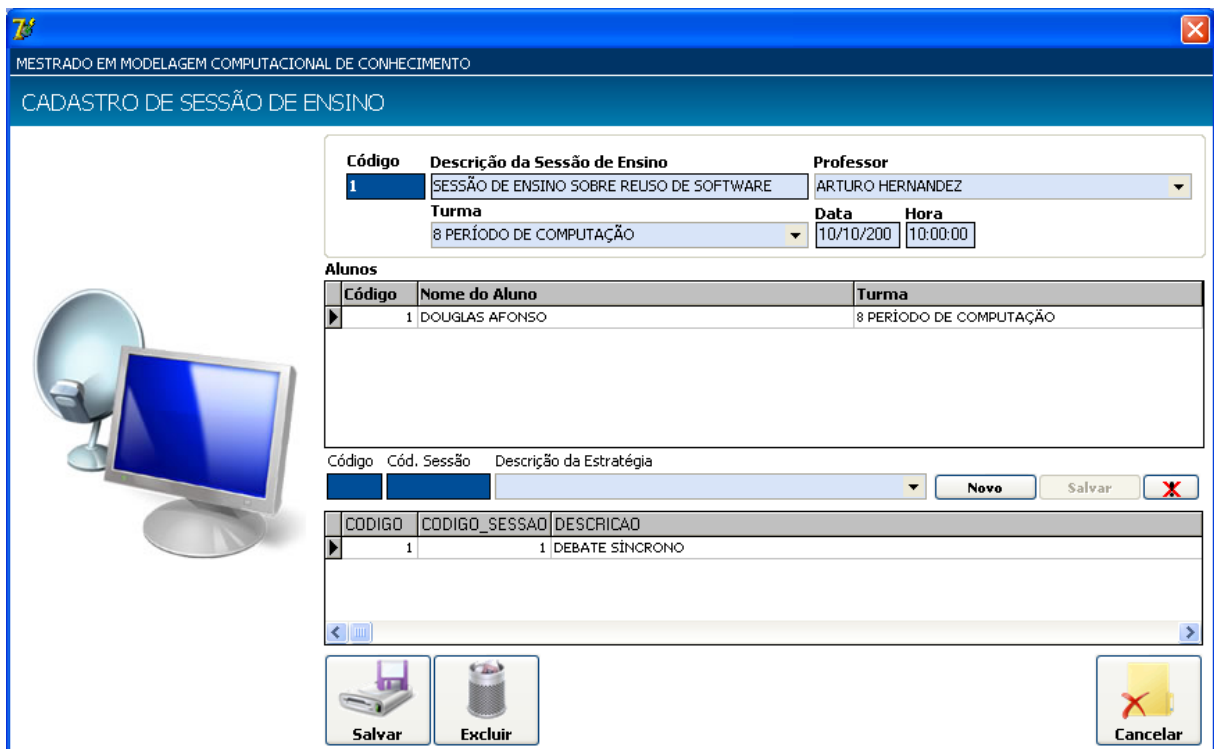


Figura 5.26: Tela de Cadastro das Sessões de Ensino

Nesta tela o professor irá preencher todas as informações pertinentes a criação da sessão de ensino. A princípio será informada a descrição da sessão, o professor responsável pela mesma, a turma que fará parte, a data e a hora de início da sessão. Note que ao selecionar a turma que fará parte da sessão de ensino, o Grid abaixo selecionará automaticamente todos os alunos que fazem parte da mesma. O próximo passo é inserir todas as estratégias que farão parte desta sessão de ensino. Para isso basta pressionar o botão Novo e selecionar uma estratégia que já foi cadastrada anteriormente pelo próprio professor, finalizando o cadastro pressionando o botão Salvar.

5.3.1.3 Interface do Aluno

Assim como foi mostrado anteriormente na tela inicial para o usuário logado com o perfil de Professor, também podemos observar ao lado esquerdo da tela todos os botões de controle do sistema que estarão disponíveis para o usuário Aluno. Quando pressionado o botão Atividades, será habilitado um menu superior com as opções de visualização das atividades disponíveis e da entrega das atividades.



Figura 5.27: Tela inicial para o usuário com perfil de Aluno

Ao pressionar o botão Avisos, será habilitado um menu com as opções de Cadastrar Avisos (nesse caso ficará desabilitado pelo fato de existir restrição de acesso à esse usuário) e a opção de Visualização dos Avisos existentes. Na opção de Debate podemos usar de duas maneiras, a primeira é quando um ou mais usuários se conectam em um determinado horário combinado com o professor para debater sobre algum assunto relevante a sua disciplina sem limite de tempo para o mesmo. A segunda maneira é quando o professor cria uma sessão de ensino, e nessa sessão adicionar uma estratégia de debate síncrono, onde o mesmo determinará o início e a duração do debate. A seguir serão detalhadas todas essas funcionalidades.

5.3.1.3.1 Visualização das Atividades

Após ser pressionado o link de visualização de atividades, a tela acima (Figura 5.28) será disponibilizada para o usuário. Nela será possível visualizar todas as atividades que estão disponíveis para o usuário logado, ou seja, só serão disponibilizadas as atividades destinadas ao usuário em questão de acordo com a turma pela qual a atividade foi encaminhada, podendo assim, existirem atividades de várias disciplinas.

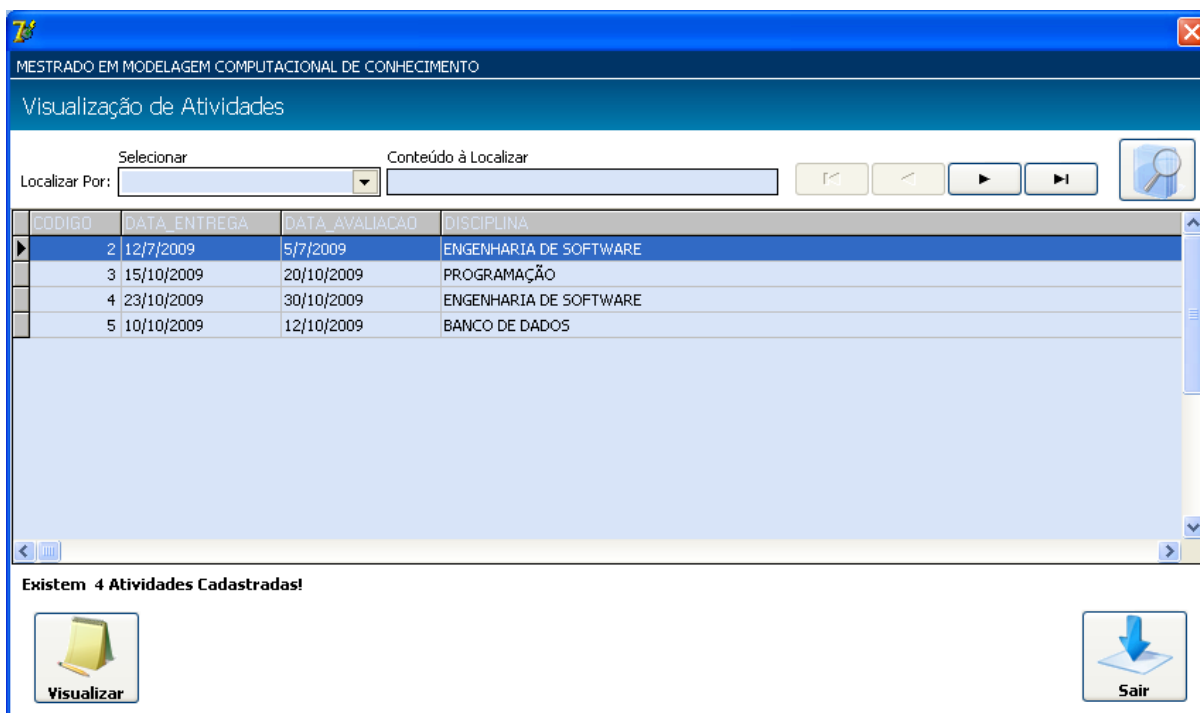


Figura 5.28: Tela de Visualização de Todas as Atividades Disponíveis

O usuário poderá fazer consultas por disciplinas ou até mesmo por intervalos de datas, para chegar a uma determinada atividade. Após a consulta, o usuário deverá selecionar uma atividade e pressionar no botão Visualizar, onde o mesmo será encaminhado para uma nova tela onde serão disponibilizadas todas as informações relativas a essa atividade.

MESTRADO EM MODELAGEM COMPUTACIONAL DE CONHECIMENTO

CADASTRO DE ATIVIDADES

Professor:

Código: 3 Data de Entrega: 15/10/2009 Data da Avaliação: 20/10/2009 Turma: 1º PERÍODO DE COMPUTAÇÃO

Título da Atividade: ATIVIDADE DE PROGRAMAÇÃO

Descrição da Atividade: FAÇA UM PROGRAMA QUE RECEBA O NOME DE UM ALUNO, TRÊS NOTAS E CALCULE A MÉDIA ARITMÉTICA.

Caminho do Arquivo: Procurar... Disciplina: BANCO DE DADOS

Salvar Excluir Cancelar

Figura 5.29: Tela de Visualização de uma Atividade Específica

É nesta tela que o aluno terá todas as informações necessárias em relação ao conteúdo da atividade.

5.3.1.3.2 Entregar Atividades (Atividades Finalizadas)

Nesta de tela de controle, temos todas as atividades que foram finalizadas por um determinado aluno. Ao se pressionar o botão Novo (Figura 5.30), o usuário será direcionado a uma tela de cadastro para a finalização de uma determinada atividade.

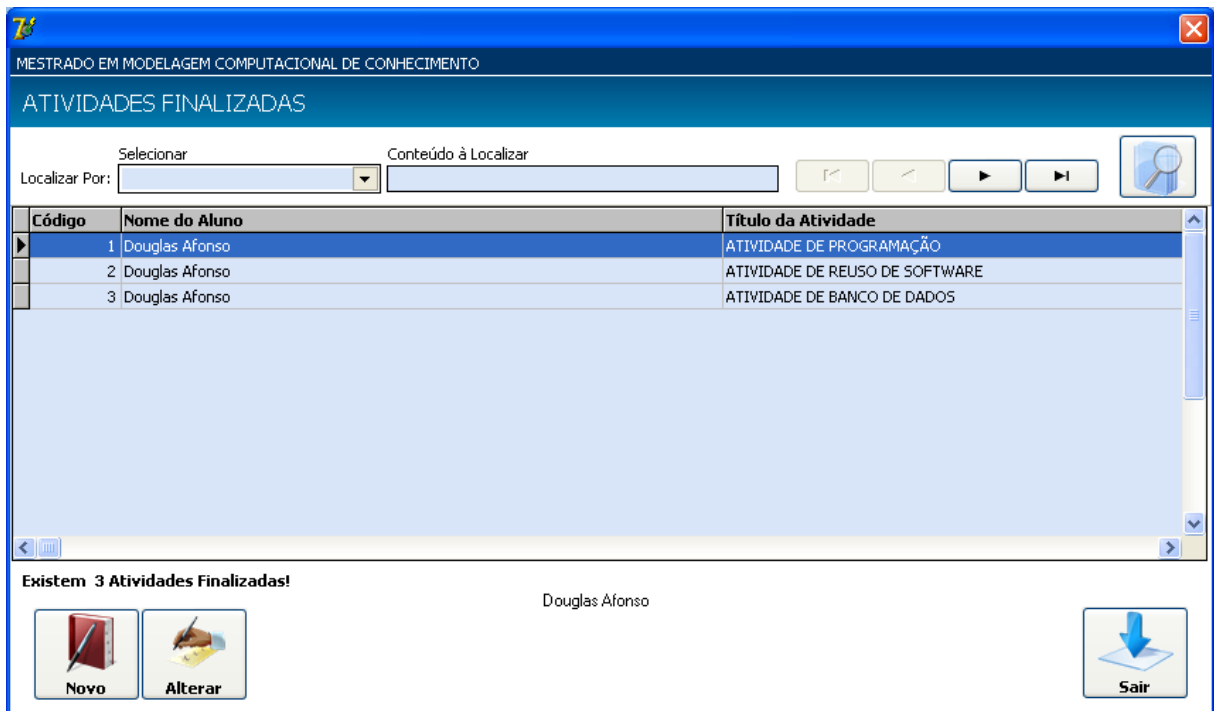
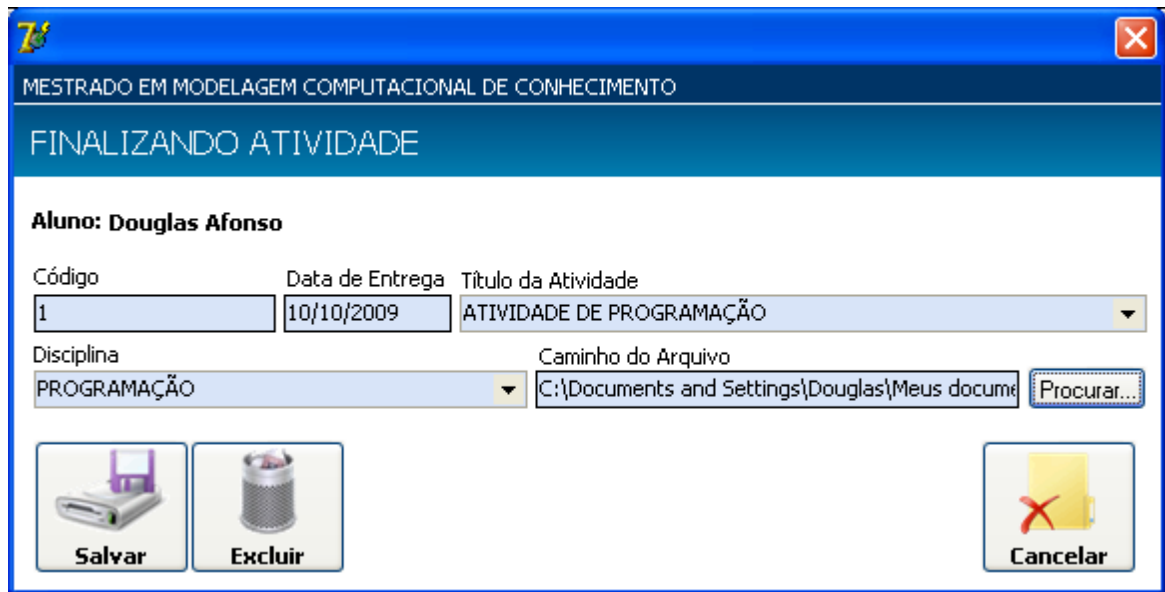


Figura 5.30: Tela de Controle das Atividades Finalizadas

Nesta tela serão preenchidas as informações necessárias para a finalização de uma determinada atividade (Figura 5.31). O aluno deverá informar a data da entrega da atividade, selecionar a atividade que deseja finalizar, selecionar a disciplina relacionada a essa atividade e por fim anexar o arquivo relativo à realização da atividade em questão. Quando o botão Salvar for pressionado, automaticamente todas as informações serão enviadas ao professor responsável pela atividade.

O aluno também tem a possibilidade de visualização de avisos que foram criados pelos professores. Para ter acesso aos avisos basta o usuário com perfil de aluno se direcionar a tela principal e pressionar o botão avisos que fica ao lado esquerdo da tela principal. Ao selecionar essa opção será habilitada um menu superior com a opção de visualização de avisos (Figura 5.32).



MESTRADO EM MODELAGEM COMPUTACIONAL DE CONHECIMENTO

FINALIZANDO ATIVIDADE

Aluno: Douglas Afonso

Código: 1 Data de Entrega: 10/10/2009 Título da Atividade: ATIVIDADE DE PROGRAMAÇÃO

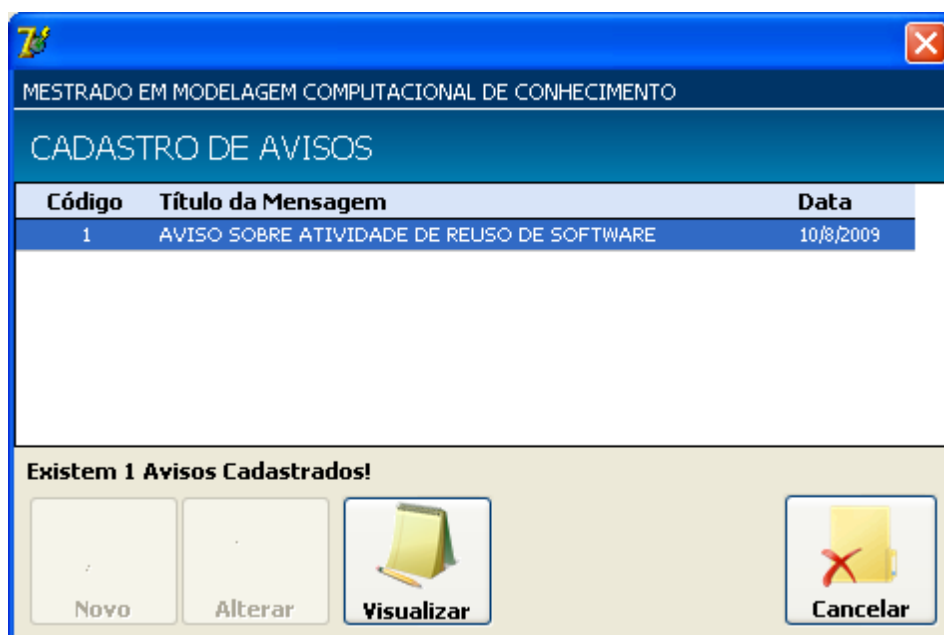
Disciplina: PROGRAMAÇÃO Caminho do Arquivo: C:\Documents and Settings\Douglas\Meus docume Procurar...

Salvar Excluir Cancelar

Figura 5.31: Tela de Finalização de uma Atividade

5.3.1.3.3 Visualização de Avisos

Basta selecionar o aviso desejado e pressionar o botão Visualizar. O aluno será direcionado à seguinte tela (Figura 5.33).



MESTRADO EM MODELAGEM COMPUTACIONAL DE CONHECIMENTO

CADASTRO DE AVISOS

Código	Título da Mensagem	Data
1	AVISO SOBRE ATIVIDADE DE REUSO DE SOFTWARE	10/8/2009

Existem 1 Avisos Cadastrados!

Novo Alterar Visualizar Cancelar

Figura 5.32: Tela de Avisos

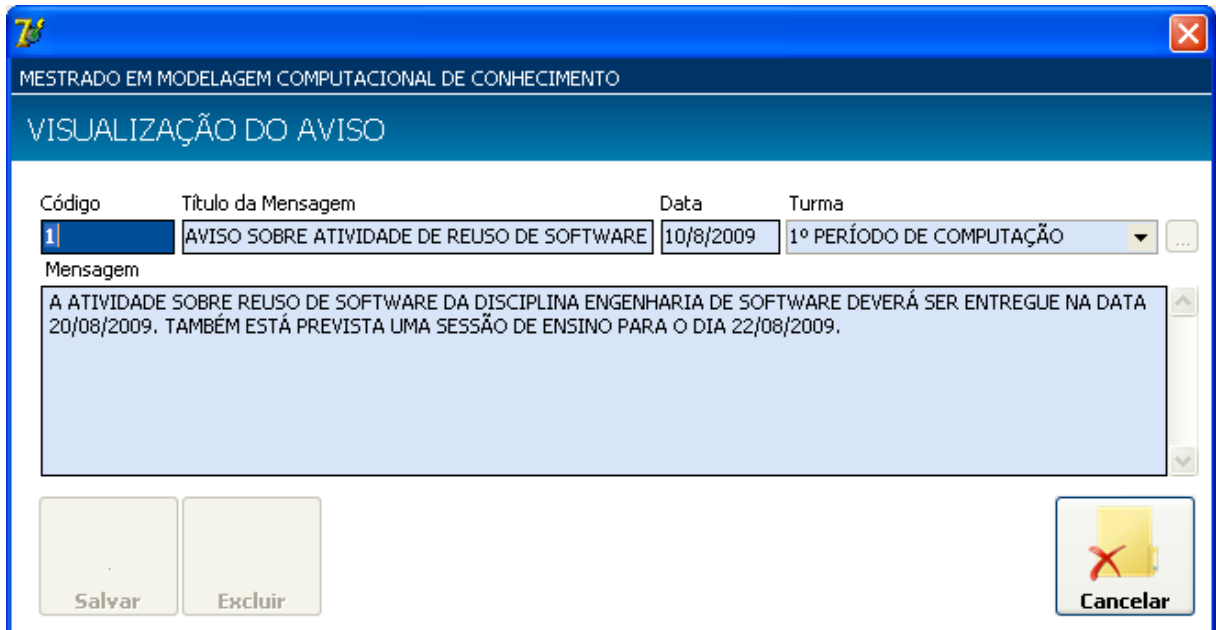


Figura 5.33: Tela de Visualização de Avisos

5.4 Execução de uma Sessão de Ensino

Para a execução de uma sessão de ensino, é necessário primeiramente criar a estratégia didática referente a esta sessão de ensino, posteriormente o professor deverá criar a sessão de ensino com os alunos que irão participar, informar a estratégia que será utilizada, a data e a hora de início da sessão. Podemos observar na seção 5.1.4 que isso já foi feito na demonstração da utilização do sistema, quando o usuário com perfil de Professor, fez os cadastros das táticas, das estratégias que poderiam ser utilizadas e posteriormente criou a sessão de ensino. No exemplo a seguir foi utilizado apenas o debate síncrono para a sessão de ensino sobre Reuso de Software. Nesta sessão o debate deverá começar às 10:00 do dia 10/08/2009 e terá uma duração de 10 minutos. Para que a sessão tenha início, o usuário deverá se logar no sistema às 10:00 ou antes.

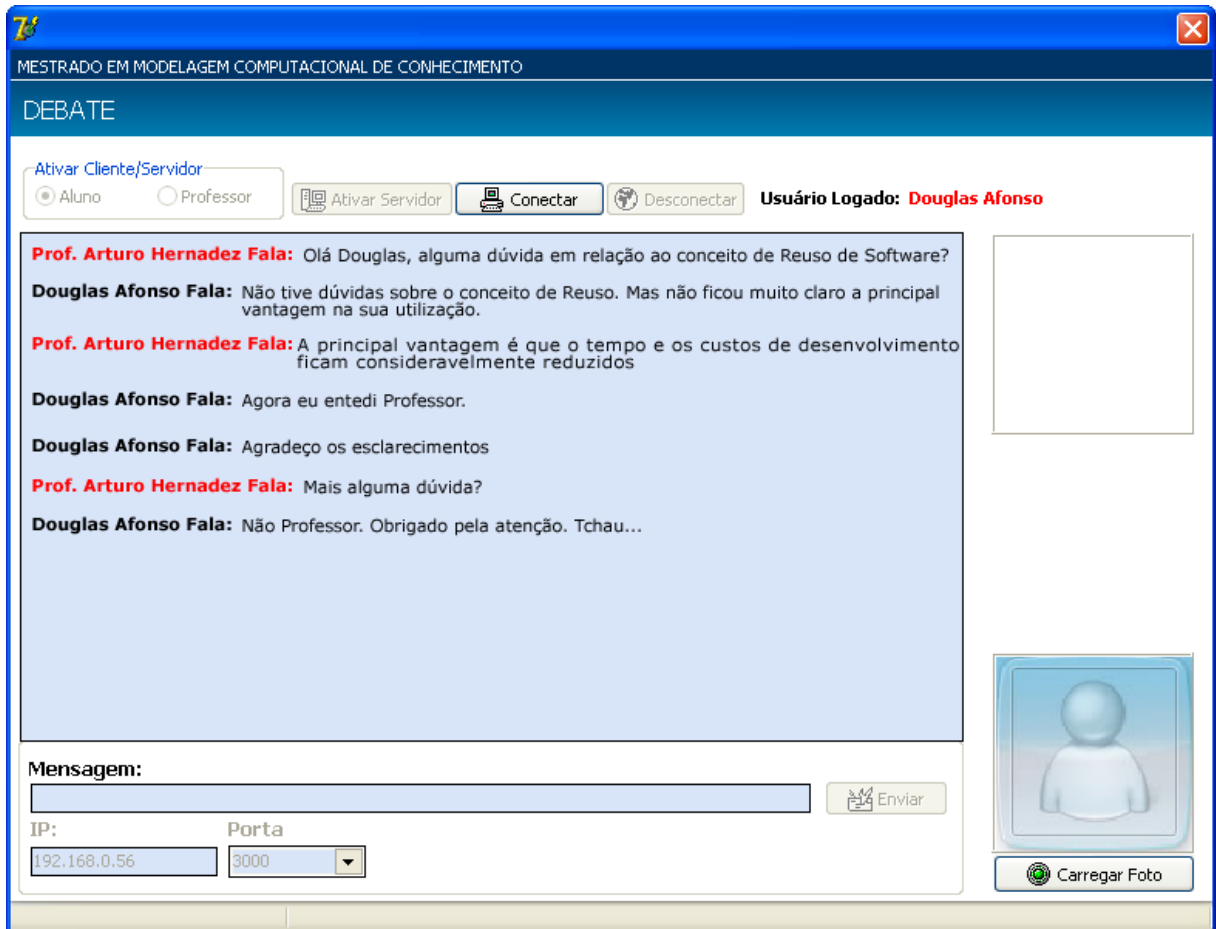


Figura 5.34: Tática de Ensino Utilizando Debate Síncrono

5.5 Resumo

Neste capítulo foi apresentado um protótipo do sistema Portfólio-Tutor baseado em componentes.

Através de uma aplicação Portfólio-Tutor o professor tem informações sobre o rendimento de cada aluno individualmente, vendo sua evolução ao longo do tempo, no contexto de uma determinada disciplina.

Com as táticas de ensino previamente implementadas, o funcionamento do Portfólio-Tutor vai depender das estratégias didáticas que serão definidas pelo professor.

6 CONSIDERAÇÕES FINAIS

Neste capítulo serão apresentadas as considerações finais, os resultados obtidos e as perspectivas para futuros trabalhos.

Foi apresentado nesta dissertação o desenvolvimento de um Sistema Portfólio-Tutor. Para isso foram analisados vários sistemas de tipo portfólio eletrônico, tais como: Portfólio Eletrônico para Web (Silva, 2002), Portfólio-Tutor (Nascimento, 2002), POETA (Sistêlos, 1999), Grady Profile (EportConsortium 2009), Connecticut College (EportConsortium 2009), Webfólio (EportConsortium 2009), PASS-PORT (EportConsortium 2009) e Epsilon(EportConsortium 2009).

Com a crescente utilização de sistemas portfólio eletrônico, muitos vêm sendo desenvolvidos, mas poucos utilizam reuso de software e um Sistema Tutor acoplado a um portfólio eletrônico. O sistema proposto faz uso de componentes de softwares, reduzindo de forma considerável o tempo e o custo de desenvolvimento de um sistema Portfólio-Tutor. Isto representa a principal contribuição deste trabalho.

A seguir serão apresentados os resultados obtidos neste trabalho e as perspectivas para trabalhos futuros.

6.1 Resultados Obtidos

De acordo com as funcionalidades disponibilizadas pelo sistema Portfólio-Tutor Baseado no Reuso de Software, podemos destacar algumas contribuições:

- Participação mais ativa e autônoma do aluno;
- Colaboração entre alunos através de mensagens e debates e, entre alunos e professor;
- Estimula a criatividade, capacidade crítica e iniciativa dos alunos;
- Gerencia o acompanhamento individual do aluno ao longo do tempo;
- Acompanha o rendimento da turma através de sessões de ensino;

- A possibilidade que o professor tem de assumir o papel de facilitador, orientador e acima de tudo incentivador.

6.2 Trabalhos Futuros

Com base nos resultados obtidos e visualizando a continuidade desse trabalho, sugerem-se as seguintes atividades:

- Implementar o restante das funcionalidades do sistema proposto;
- Implementação de interface para o acompanhamento de outros tipos de táticas, tais como, debates assíncronos, envio de mensagens com arquivo anexado;
- Aplicação de técnicas de Inteligência Artificial, aumentando a capacidade inteligente do tutor;
- Implementação de consultas mais detalhadas em relação à base histórica armazenada.

REFERÊNCIAS BIBLIOGRÁFICAS

- BACHMANN, F. (2000), 'Technical concepts of component-based software engineering', Relatório Técnico CMU/SEI-2000-TR-008. Software Engineering Institute, Carnegie Mellon 2.
- BATORY, D. *Product Line Architectures*. Erfurt, Germany. Smaltalk and Java in Industrie and Ausbildung. 1998.
- BASS, Len et al. Product Line Practice Workshop Report. Pittsburgh: [s. n.], 1997. 36 p. (Technical Report CMU/SEI-97).
- BIANCHINI, CABELE P., ALMEIDA, EDUARDO S., FONTES, DIOGO S. E ANDRADE, ROSSANA M. C., Um Padrão Para Gerenciamento de Redes, Universidade Anhembi Morumbi, Universidade Federal de Pernambuco, Universidade Paulista, Universidade Federal do Ceará.
- BOOCH, GRADY, RUMBAUGH, JAMES, JACOBSON, IVAR. (2000) UML – Guia do Usuário. Tradução de Fábio Freitas da Silva. Rio de Janeiro: Campus, 2000.
- CAMPBELL, C. (1997), 'How to develop a professional portfolio - a manual for teachers', Allyn & Bacon, USA.
- CLEMENTS, P. et al. Software Product Lines – Practices and Patterns. [S.l.]: Addison Wesley, 2001.
- COHEN, S. G.; HESS, J. A.; KANG, K. C.; NOVAK, W. E.; PETERON, A. S. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Pittsburgh, Pennsylvania, USA, nov. 1990.
- D'SOUZA, D. E WILLS, A., Objects, Components and Frameworks with UML: The Catalysis Approach, Massachusetts, Addison-Wesley Longman, 1998.
- Eportconsortium (2009). Electronic portfolio white paper. <http://eportconsortium.org/>.
- FAYAD, MOHAMED; SCHMIDT, DOUGLAS; JOHNSON, RALPH. Building Applications Frameworks. Editora John Willey & Sons, 1999.
- F. R. LAZILHA, I. M. S. GIMENES, E. A. OLIVEIRA JUNIOR. *Uma Arquitetura de Linha de Produto para Sistemas de Gerenciamento de Workflow de Acordo com a Abordagem de Desenvolvimento Baseado em Componentes*, in 2003 Proc. 3as Jornadas Ibero-Americanas de Engenharia de Software e Engenharia de Conhecimento, Nov. 2003).
- FONTOURA, MARCUS; PREE, WOLFGANG; RUMPE, BERNHARD. UML-F: A Modeling Language for Object-Oriented Frameworks. Cannes: European Conference On Object Oriented Programming (ECOOP 2000), 2000.
- FROZZA, R. (2000), 'Uma arquitetura para acompanhamento pedagógico em ensino a distância', XI Simpósio Brasileiro de Informática na Educação - SBIE - Maceió, Alagoas.

FURTADO, M. E. S. (1993), 'Uma metodologia para projeto de banco de dados temporal orientado a objetos', Dissertação de Mestrado, Universidade Federal da Paraíba _ COPIN/DSC, Campina Grande, Paraíba, Brasil.

GAMMA I. M. S et al. Designer Patterns: elements of reusable object-oriented software. Massachusetts: Addison-Wesley, 1995.

GIRAFFA, LUCIA MARIA MARTINS. (1999) Uma Arquitetura de Tutor utilizando Estados Mentais. 1999. 177 f. Tese (Doutorado em Ciência da Computação) - Programa de Pós-Graduação em Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre.

GIMENES, ITANA M. S., NISHIMURA, RUY., OLIVEIRA, EDSON A., LAZILHA, FABRÍCIO R., KULESZA, UIRÁ., LUCENA, CARLOS J. P., A Component-based Product Development Process for a Workflow Management System Product Line, 2004.

GIMENES, I. M. S., BARROCA, L., HUZITA, E. H. M., CARNIELLO, A., “*O Processo de Desenvolvimento Baseado em Componentes Através de Exemplos*”, In: ERI 2000 – VIII Escola Regional de Informática da SBC-Sul, Foz do Iguaçu-PR, 2000.

GIMENES, ITANA M. S., LAZINLHA, FABRÍCIO, R., HALMEMAN, RADAMES J., Uma Arquitetura de Linha de Produto Baseada em Componentes para Sistemas de Gerenciamento de Workflow, 2000.

GIMENES, ITANA M. S., O Processo de Desenvolvimento Baseado em Componentes Através de Exemplos. In: Escola de Informática da SBC-Sul, 8., 2000, Foz do Iguaçu. Livro Texto. Santa Maria: UFSM, 2000.

GIMENES, ITANA M. S.; TRAVASSOS, G. H. O Enfoque de Linha de Produto para Desenvolvimento de Software. Florianópolis, SC: SBC, 2002. Trabalho apresentado na Jornada de Atualização em Informática, 2002.

GRISS, M. et al. Integrating Feature Modeling with RSEB. In: International Conference on Software Reuse, ICSR-5, 5. 1998, Victoria.

GODOY, ADRIANO. *Matriz: a arte de controlar reações e ser uma pessoa eficaz*. Virtual Book. Vivali Editora Eletrônica Ltda (Setembro de 2006).

HART, D. (1994), Authentic Assessment - A Handbook for Educations, Addison Wesley, USA.

HERNÁNDEZ-DOMÍNGUEZ, A. (1997), ‘Specification and Implementation of an adaptable virtual class’, In: *ED-MEDIA 97, World Conferences on Educational Multimedia and Hypermedia on Educational Telecommunications, Calgary, Canada*.

HAINES, G. CARNEY, D. E FOREMAN, J. Component – Based Software Development/COTS Integration, Software Engineering Institute, Carnegie Mellon University, USA. Janeiro, 1997.

JACOBSON, I., GRISS, M. & JOHNSON, P. (1997), *Software Reuse-architecture process and organization for business success*, Addison Wesley, New York: Springer.

JOHNSON, R. E. & FOOTE, B. (2001), 'Designing reusable classes', *Journal of Object Oriented Programming, JOOP*.

KANG, K.; LEE, J.; DONOHOE, P., 2002, *Feature-Oriented Product Line Engineering*. IEEE Software, July, 58-65p, August 2002.

KNABBEN, ANDRÉ LUÍS; ROBERT, THIAGO. *Framework para Jogos de Cartas*. Porto Alegre: Universidade Federal do Rio Grande do Sul, 2002.

LANKES, A. M. D. (1995), 'Electronic portfolios: A new idea in assessment', *Clearinghouse on Information & Technology*, Syracuse University, Center for Science and Technology, Syracuse, New York .

LARMAN, CRAIG. (2000) *Utilizando UML e Padrões: uma introdução à análise e ao projeto orientados a objetos*. Tradução de Luiz Augusto Meirelles Salgado. Porto Alegre: Bookman, 2000.

LAZILHA, F. R., PRICE, R. T., GIMENES, I. M. S., “*Uma Proposta de Arquitetura de Linha de Produto para Sistemas de Gerenciamento de Workflow*”, *Dissertação de Mestrado*, Universidade Federal do Rio Grande do Sul, Porto Alegre-RS, 2002.

LAZILHA, F. R., GIMENES, I., PRICE, R.T: *Uma Proposta de Arquitetura de Linha de Produto para Workflow Management Systems*. SBES 2002 (Simpósio Brasileiro de Engenharia de Software).

MADEIRO, J. M., MEDEIROS, F. N. & HERNÁNDEZ-DOMÍNGUEZ, A. (2005), 'Um framework para portfólios eletrônicos', XVI Simpósio Brasileiro de Informática na Educação, Workshop de Arquiteturas Pedagógicas para Suporte à Educação a Distância mediada pela Internet, Juiz de Fora/MG, Brasil .

MEDEIROS, F. N., MEDEIROS, F. M. & HERNÁNDEZ-DOMÍNGUEZ, A. (2006), 'Fa_port: Um framework para sistemas portfólio-tutor utilizando agentes', XVII Simpósio Brasileiro de Informática na Educação, Brasília/DF, Brasil .

MEDEIROS, F. N. (2006), 'FA_Port: Um Framework para sistemas Portfólio-Tutor Baseado em Agentes', *Dissertação de Mestrado em Modelagem Computacional de Conhecimento*, Universidade Federal de Alagoas – UFAL – Instituto de Computação – IC, Maceió, Alagoas, Brasil.

MENDES NETO, FRANCISCO MILTON. (2000) *E-Grupo - Um Ambiente para Suporte à Aprendizagem colaborativa baseada na Web*. 2000. 145 f. *Dissertação (Mestrado em Informática) – Coordenação de Pós-Graduação em Informática*, Universidade Federal da Paraíba, Campina Grande.

MOKHTARI, K. & YELLIN, D. (1996), 'Portfolio assessment in teacher education: impact on preservice teachers' knowledge and attitudes', *Journal of Teacher Education*.

NASCIMENTO, D. M. C. (2002), 'Um Sistema Tutor Acoplado a um Portfólio Eletrônico no Contexto da Educação a Distância – Portfólio-Tutor', *Dissertação de Mestrado, Universidade Federal de Campina Grande – UFCG – Centro de Ciências e Tecnologia, Campina Grande, Paraíba, Brasil.*

NASCIMENTO, D. M. C., HERNÁNDEZ-DOMÍNGUEZ, A. & SCHIEL, U. (2002), 'Portfolio-tutor: Um tutor acoplado a um portfolio eletrônico no contexto da educação a distância', XIII Simpósio Brasileiro de Informática na Educação, São Leopoldo, Rio Grande do Sul.

OLIVEIRA, EDSON A., GIMENEZ, ITANA M. S., Especificação de um Sistema Gerenciador de Workflow de Acordo com a Abordagem de Desenvolvimento Baseado em Componentes, Universidade Estadual de Maringá, 2003.

OLIVEIRA, EDSON A. (2002), 'Especificação do Ambiente ExpSEE de acordo com a abordagem de desenvolvimento baseado e componentes', *Trabalho de Conclusão de Curso, Universidade Estadual de Maringá, Centro de Tecnologia, Departamento de Informática, CTC, DIN, Maringá, Paraná, Brasil.*

POULIN, J., Software Architectures, Products Lines, and DSSAs: Choosing the Appropriate Level of Abstraction . In: International Workshop on Software Reuse, 1997, Columbus.

SILVA, ALEKSANDRA DO SOCORRO DA.(2000) TUTA – Um Tutor Baseado em Agentes no Contexto do Ensino a Distância. 2000. 141 f. Dissertação (Mestrado em Informática) – Coordenação de Pós-Graduação em Informática, Universidade Federal da Paraíba, Campina Grande.

SILVA, E. P. (2002), 'Desenvolvimento de um Portfólio Eletrônico para Web baseado nos princípios de banco de dados ativo e temporal', *Trabalho de Conclusão de Curso, Universidade Federal de Alagoas – UFAL – Departamento de Tecnologia da Informação – TCI, Maceió, Alagoas, Brasil.*

SILVA, RICARDO P., PRICE, ROBERTO T. (1999), 'Suporte ao Desenvolvimento e Uso de Componentes Flexíveis, Universidade Federal de Santa Catarina, Universidade Federal do Rio Grande do Sul'.

SILVEIRA, R. A. (1995), 'Inteligência Artificial em educação: Um Modelo em Sistema Tutorial Inteligente para Microcomputadores', *Dissertação de Mestrado - Porto Alegre - FATED/PUCRS.*

SISTÊLOS, A. J. C. M. (1999), 'Um Ambiente Computacional de Apoio ao Método de Avaliação Autêntica: Projeto POETA (Portfólio Eletrônico Temporal e Ativo)', *Dissertação de Mestrado em Informática – Coordenação de Pós-Graduação em Informática – Universidade Federal da Paraíba – UFPB – João Pessoa, Paraíba, Brasil.*

SISTÊLOS, A. J. C. M., SCHIEL U., HERNÁNDEZ-DOMÍNGUEZ, A. (1998) Um Sistema de Apoio ao Método de Avaliação Autêntica: Projeto POETA (Portfolio Eletrônico Temporal de Ativo). In: Simpósio Brasileiro de Informática na Educação, 9., 1998, Fortaleza. Anais... Fortaleza: UFC, 1998.

SOMMERVILLE, IAN. (2007), ' Engenharia de Software. 8ª Edição. São Paulo: Pearson Education'.

VALENTE, J. A. (1998), 'Formação de Profissionais na Área de Informática em Educação. in: Valente (org.) computadores e conhecimento: repensando a educação', Campinas: UNICAMP/NIED pp. 139_164.

VICCARI, R. M. (1990), 'Um tutor inteligente para a programação em lógica-idealização, projeto e desenvolvimento', Tese de Doutorado - Universidade de Coimbra.

WEBER, KIVAL; ROCHA, ANA R. C.; NASCIMENTO, CÉLIA J. Qualidade e Produtividade em Software. São Paulo, Makron Books, 2001.

WERNER, CLAUDIA M. L., BRAGA, REGINA M. M., Desenvolvimento Baseado em Componentes. In SBES Workshops, 14., 2000, João Pessoa. Anais... João Pessoa: CEFET – PB, 2000. p. 297-329.

WERNER, C. M. L., BRAGA, R. M. M. Desenvolvimento Baseado em Componentes. In: SBES WORKSHOPS, 14., 2000, João Pessoa. Anais. João Pessoa: CEFET – PB, 2000.

WOLF, D.P. (1989), *Portfolio assessment: Sampling student work*, Educational Leadership.