

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO**

**PROTÓTIPO DE UM SISTEMA PARA ACOMPANHAMENTO  
DE TRABALHOS DE CONCLUSÃO DE CURSO**

**ELISÂNGELA CRISTINA LOMBARDI KLITZKE**

**BLUMENAU**  
**2009**

**2009/2-06**

**ELISÂNGELA CRISTINA LOMBARDI KLITZKE**

**PROTÓTIPO DE UM SISTEMA PARA ACOMPANHAMENTO  
DE TRABALHOS DE CONCLUSÃO DE CURSO**

Trabalho de Conclusão de Curso submetido à  
Universidade Regional de Blumenau para a  
obtenção dos créditos na disciplina Trabalho  
de Conclusão de Curso II do curso de Ciência  
da Computação — Bacharelado.

Prof. Dalton Solano dos Reis, M. Sc.- Orientador

**BLUMENAU  
2009**

**2009/2-06**

# **PROTÓTIPO DE UM SISTEMA PARA ACOMPANHAMENTO DE TRABALHOS DE CONCLUSÃO DE CURSO**

Por

**ELISÂNGELA CRISTINA LOMBARDI KLITZKE**

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Dalton Solano dos Reis, M. Sc. – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. José Roque Voltolini da Silva, FURB

Membro: \_\_\_\_\_  
Prof. Everaldo Artur Grahl, FURB

Blumenau, 15 de dezembro de 2009

Dedico este trabalho a toda a minha família, pelo incentivo, cooperação e apoio e em especial à minha filha, Brenda Manoela, que compreendeu a minha ausência, durante a realização deste trabalho.

## AGRADECIMENTOS

A Deus, meu refúgio e força, onde sempre encontrei respostas para os meus problemas.

A duas pessoas João e Alma, que em nenhum momento mediram esforços para realização dos meus sonhos, sou extremamente feliz e tenho muito orgulho por chamá-los de pai e mãe.

Ao meu marido Gerson, por estar sempre comigo em cada passo da vida, certo ou errado, por ser meu exemplo de garra, coragem e esperança, é por tudo isso que te amo pra sempre.

Aqueles que direta ou indiretamente contribuíram para que eu chegasse até aqui. Ao amigo Marcelo que voluntariamente dedicou sua atenção, carinho e não mediu esforços para me ajudar.

A empresa, Consistem Sistemas Ltda., por permitir o horário flexível quando fora necessário.

Ao meu orientador, Dalton Solano dos Reis, pela orientação e ajuda na elaboração deste trabalho.

A todos meu carinho e muito obrigado.

Quando se tem sonho grande, a vida se expande. Sonhos grandes impulsionam, motivam, dão energia.

Roberto Shinyashiki

## RESUMO

Este trabalho apresenta o desenvolvimento de uma aplicação Web cuja finalidade é automatizar o gerenciamento dos processos da disciplina de Trabalho de Conclusão de Curso, do curso de Ciência da Computação da Universidade Regional de Blumenau. O foco da aplicação está relacionado ao gerenciamento dos trabalhos de conclusão de curso realizados pelos alunos matriculados nesta disciplina, a qual é responsável pelo cumprimento de todas as etapas finais necessárias para o aluno se formar no curso de Ciência da Computação. Entre as tecnologias utilizadas no desenvolvimento da mesma destacam-se a linguagem *Javascript* e o *framework jQuery*, que possui *plugins* adicionais de controle de dados via *Ajax*. A aplicação utiliza o banco de dados MySQL e as suas páginas foram construídas em *Hypertext PreProcessor* (PHP). Para definir o *layout* foi utilizado *Cascading Style Sheets* (CSS).

Palavras-chave: Aplicação Web. Gerenciamento de processos. *Javascript*. *jQuery*. *Ajax*. MySQL. PHP. CSS. TCC.

## **ABSTRACT**

This work presents the development of a Web application whose purpose is to automatize the management of the processes of disciplines of Trabalho de Conclusão de Curso, do curso de Ciência da Computação. The focus of the application is related to the management of the works of conclusion of course carried through by the academic registered this disciplines, which is responsible for the fulfilment of all the final stages necessary for the academic to form itself in the course of Computer Science. Language enters the used technologies in the development of the same one is distinguished it JavaScript and framework jQuery, that it possess plugins you add of control of data saw Ajax. The application uses the data base MySQL and its pages had been constructed in Hypertext PreProcessor (PHP). To define the layout Style was used Cascading Sheets (CSS).

Word-key: Web application. Management of Processes. JavaScript. jQuery. Ajax. MySQL. PHP. CSS. TCC.

## LISTA DE ILUSTRAÇÕES

Quadro 1 – Visão geral do sistema.....	26
Quadro 2 – Requisito funcional registrar situação do aluno e não funcionais associados .....	27
Quadro 3 – Requisito funcional registrar banca e não funcionais associados.....	27
Quadro 4 – Requisito funcional calcular média e não funcional associado .....	27
Quadro 5 – Requisitos suplementares .....	28
Quadro 6 – Listagem de casos de uso da fase de concepção.....	28
Quadro 7 – Listagem de conceitos e operações de manutenção.....	28
Quadro 8 – Listagem de consulta .....	28
Quadro 9 – Planejamento dos ciclos iterativos.....	29
Quadro 10 – Cronograma para o desenvolvimento do sistema.....	29
Quadro 11 – Caso de uso expandido registrar situação do aluno.....	30
Quadro 12 – Caso de uso expandido registrar banca.....	30
Quadro 13 – Modelo conceitual .....	31
Quadro 14 – Contrato para operação de cadastro de novo aluno .....	32
Quadro 15 – Contrato para operação de alteração de aluno .....	32
Quadro 16 – Contrato para operação de exclusão de aluno .....	32
Quadro 17 – Contrato para operação de consulta de aluno .....	32
Quadro 18 – Diagrama de classes.....	33
Quadro 19 – Exemplo de definição da classe abstrata BD.....	35
Quadro 20 – Exemplo de definição da classe consulta .....	36
Quadro 21 – Exemplo de definição da classe banca .....	37
Quadro 22 – Exemplo de utilização CSS .....	39
Quadro 23 – Tratamento de exceção no preenchimento de campos .....	40
Figura 1 - Tela de <i>login</i> .....	41
Figura 2 – Tela de erro de autenticação.....	41
Figura 3 – Tela de menu .....	42
Figura 4 – Tela de formulário de inclusão de aluno .....	42
Figura 5 – Tela de manutenção de aluno .....	43
Figura 6 – Tela de cadastro da proposta .....	43
Figura 7 – Tela de inclusão de banca .....	44
Figura 8 – Tela de relatório de situação do aluno.....	44
Figura 9 – Caso de teste, primeiro ciclo – usuário .....	45

Figura 10 – Caso de teste, segundo ciclo – situação do aluno.....	45
Figura 11 – Caso de teste, terceiro ciclo – cadastro de banca .....	46
Quadro 24 – Lista de tarefa .....	49

## LISTA DE SIGLAS

ABNT – Associação Brasileira de Normas Técnicas

CASE – *Computer Aided Software Engineering*

CP – Câmara de Pesquisa

CSS – *Cascading Style Sheets*

DSC – Departamento de Sistemas e Computação

FURB – Fundação Universidade Regional de Blumenau

IDS – Intrusion Detecting System

PPP – Plano Político Pedagógico

PHP – *Hipertext PreProcessor*

SSL – *Secure Server Layer*

TCC – Trabalho de Conclusão de Curso

UML – *Unified Modeling Language*

UP – *Unified Process*

XML – *eXtensible Markup Language*

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>13</b>
1.1 OBJETIVOS DO TRABALHO .....	14
1.2 ESTRUTURA DO TRABALHO .....	14
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>16</b>
2.1 TRABALHO DE CONCLUSÃO DE CURSO.....	16
2.1.1 TCC-I.....	16
2.1.2 TCC-II.....	17
2.2 PROCESSO UNIFICADO.....	18
2.2.1 Concepção.....	19
2.2.2 Elaboração.....	20
2.2.3 Construção .....	22
2.3 SEGURANÇA.....	23
<b>3 DESENVOLVIMENTO DO PROTÓTIPO.....</b>	<b>26</b>
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	26
3.1.1 Levantamento de requisitos.....	26
3.1.2 Organização dos requisitos .....	28
3.1.3 Planejamento de desenvolvimento.....	29
3.2 ESPECIFICAÇÃO .....	29
3.2.1 Expansão do caso de uso.....	30
3.2.2 Modelo conceitual.....	31
3.2.3 Contratos .....	31
3.2.4 Diagrama de classes .....	33
3.3 IMPLEMENTAÇÃO .....	34
3.3.1 Geração de código.....	34
3.3.2 Interface.....	38
3.4 OPERACIONALIDADE DA IMPLEMENTAÇÃO.....	41
3.5 TESTES.....	44
3.6 RESULTADOS E DISCUSSÃO .....	46
<b>4 CONCLUSÃO.....</b>	<b>48</b>
4.1 EXTENSÕES .....	48
<b>ANEXO A – AVALIAÇÃO DO(A) ORIENTADOR(A) .....</b>	<b>53</b>
<b>ANEXO B – FICHA DE AVALIAÇÃO DA MONOGRAFIA .....</b>	<b>53</b>

<b>ANEXO C – FICHA DE AVALIAÇÃO DA APRESENTAÇÃO PÚBLICA.....</b>	<b>54</b>
<b>ANEXO D – FICHA DE AVALIAÇÃO GERAL DO TCC .....</b>	<b>55</b>

## 1 INTRODUÇÃO

A disponibilidade das tecnologias de informação e a popularidade da Web geram novas oportunidades de comunicação, sobretudo pela capacidade em agilizar as informações e aproximar distâncias. Todas as pessoas, empresas e entidades necessitam encontrar caminhos que possibilitem utilizar a tecnologia da informação e a internet em seus negócios e em suas rotinas operacionais (SEVERO, 2006, p. 1). Entretanto, devido à complexidade dos sistemas atualmente disponíveis surge a necessidade de desenvolver tecnologia da informação usando metodologias adequadas.

Uma opção de atender esta necessidade é o método *Unified Process* (UP), Wazlawick (2004, p. 17) descreve um método sistemático orientado a objetos para desenvolvimento de software, que apresenta quatro fases organizadas do seguinte modo: concepção, elaboração, construção e transição. A fase de concepção incorpora o estudo de viabilidade e uma parte da análise de requisitos, na qual se procura levantar os principais requisitos e compreender o sistema de forma abrangente. A fase de elaboração incorpora a maior parte da análise de requisitos, a análise de domínio e o projeto. A fase de construção corresponde a programação e testes. A fase de transição consiste na instalação e manutenção do sistema, substituindo o sistema atual, seja ele manual ou computadorizado.

Sendo assim, este trabalho apresenta a automação do processo de acompanhamento de Trabalhos de Conclusão de Curso (TCCs), no curso de Ciências da Computação da Universidade Regional de Blumenau, usando o método proposto por Wazlawick. Hoje o sistema já controla o processo de gestão de TCCs, amenizando as tarefas do coordenador das disciplinas de TCCs e de todos os envolvidos.

Entre as tarefas que o coordenador de TCCs deve executar estão: controle dos prazos para a entrega de toda a documentação necessária para iniciar o TCC, envio de avisos aos alunos que estão atrasados com o prazo de entrega dos documentos, encaminhamento da proposta, do volume parcial e final para avaliação dos professores da banca, e por fim o agendamento da apresentação.

Ao longo do semestre os alunos matriculados nas disciplinas de TCC, os professores orientadores e os professores avaliadores passam por diversas etapas. Aos alunos cabem as seguintes etapas: indicação de um professor orientador e elaboração de documentos diversos

(proposta, relatórios de andamento do trabalho, volume final). Os professores orientadores devem: aceitar ou não alunos como seus orientados; acompanhar o andamento do trabalho; revisar e aprovar documentos diversos antes da sua distribuição. Já os professores avaliadores que participam de bancas recebem documentos, avaliando-os e emitindo pareceres.

Com a informatização deste sistema o coordenador tem o benefício de poder acompanhar o andamento dos trabalhos em um espaço de tempo menor, minimizando erros, aproveitando melhor as informações, gerando maior produtividade e confiabilidade, melhorando o atendimento e trazendo vantagens para toda a comunidade acadêmica.

Para a implementação do sistema foi utilizada a linguagem *PHP*, usada para criar *sites* dinâmicos, que permite a interação direta do usuário com o mesmo. O banco de dados utilizado foi o MySQL.

## 1.1 OBJETIVOS DO TRABALHO

O objetivo principal deste trabalho é a modelagem e a implementação de um protótipo de um sistema para acompanhamento do desenvolvimento dos TCCs.

Os objetivos específicos do trabalho são:

- a) disponibilizar informações na Web para os alunos e professores sobre o andamento dos TCCs;
- b) possibilitar ao coordenador supervisionar todo o processo;
- c) agilizar o trâmite de informações importantes às pessoas envolvidas.

## 1.2 ESTRUTURA DO TRABALHO

Na primeira seção, encontra-se uma introdução e os objetivos a serem alcançados com o desenvolvimento deste trabalho. Na segunda seção é apresentada a fundamentação teórica, contextualizando o conceito de TCC, UP e segurança. A terceira seção aborda o desenvolvimento do trabalho, apresentando os requisitos, a especificação e as ferramentas

utilizadas na especificação, desenvolvimento e execução do sistema, resultados e problemas encontrados durante a implementação. Finalmente, na quarta seção trata das considerações finais sobre o trabalho e sugestões para extensões.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentados alguns aspectos teóricos relacionados ao trabalho. Primeiramente detalhando os principais processos referentes ao andamento do TCC, descrevendo as principais fases do método UP. Na continuação uma breve explicação sobre como programar com segurança em PHP e na última seção são examinadas as tecnologias usadas.

### 2.1 TRABALHO DE CONCLUSÃO DE CURSO

Segundo Universidade Regional de Blumenau (2004, p. 3), o TCC é uma atividade obrigatória que consiste de trabalho final de graduação, abordando temas das áreas de estudo relacionados ao Plano Político Pedagógico (PPP) do curso de Ciências da Computação da Universidade Regional de Blumenau e às linhas de pesquisa da área de formação.

O objetivo geral do TCC é possibilitar ao acadêmico o desenvolvimento de sua capacidade intelectual, científica e criativa.

Os objetivos específicos são:

- a) dinamizar as atividades de ensino-aprendizagem;
- b) permitir que o acadêmico possa integrar teoria e prática, consolidando a sua formação intelectual e profissional;
- c) proporcionar ao acadêmico a oportunidade de realizar experiências de pesquisa e extensão universitária;
- d) possibilitar ao acadêmico a aplicação dos conhecimentos adquiridos ao longo do curso, através do desenvolvimento do trabalho de pesquisa.

Ao final do curso são oferecidas duas disciplinas de conclusão de curso: TCC-I, oferecida no penúltimo semestre e TCC-II, no último semestre.

#### 2.1.1 TCC-I

Na disciplina de TCC-I é definido um tema, um orientador e elaborada uma proposta.

O acadêmico pode matricular-se na disciplina de TCC-I, desde que tenha cumprido os pré-requisitos exigidos pelo curso.

O tema é o início do processo de desenvolvimento de TCC. Conforme Universidade Regional de Blumenau (2004, p. 5), nas aulas presenciais da disciplina de TCC-I o professor fornece orientações e diretrizes para a definição do mesmo e também para a escolha do orientador.

Para a escolha de um tema o aluno deve levar em conta as áreas de pesquisa que mais chamaram a sua atenção durante o curso ou que teve maior afinidade. Para garantir o efetivo desenvolvimento do trabalho o professor orientador escolhido deverá pertencer à área de pesquisa abordada.

Já o passo seguinte é a elaboração de uma proposta. Nela deve estar definido o que é pretendido fazer no decorrer da disciplina de TCC-II. A proposta deve ser formalizada sob a supervisão do orientador, seguindo um modelo disponibilizado aos alunos, devendo apresentar introdução, objetivos, relevância, metodologia, revisão bibliográfica, trabalhos correlatos, requisitos do sistema a serem desenvolvidos, considerações finais e referências bibliográficas.

A proposta deve ser entregue ao professor de TCC-I até a data estipulada por ele, devidamente assinada pelo orientador e pelo acadêmico para a avaliação que é feita por ele, pelo coordenador de TCC e por outro professor indicado pela Câmara de Pesquisa (CP) do Departamento de Sistemas e Computação (DSC). O acadêmico que não tiver sua proposta aprovada é reprovado na disciplina de TCC-I. A avaliação é feita em um formulário específico, mostrado no ANEXO A.

### 2.1.2 TCC-II

Na disciplina de TCC-II é desenvolvida a monografia e defendida publicamente. O acadêmico pode matricular-se desde que tenha sido aprovado na disciplina de TCC-I, seja formando no semestre corrente e esteja cursando, no máximo, oito créditos acadêmicos de outras disciplinas do currículo pleno.

Deve ser realizada uma reunião a cada semana do aluno com o seu orientador, para discutir assuntos pertinentes ao andamento do trabalho. O acompanhamento é feito através de um relatório, onde são colocados os problemas observados, os assuntos tratados e as pendências para a próxima reunião.

A estrutura e apresentação do TCC seguem as normas técnicas e metodologia do trabalho acadêmico adotadas pela FURB, as quais devem estar em conformidade com o que estabelece a Associação Brasileira de Normas Técnicas (ABNT).

O regulamento do trabalho de conclusão de curso (UNIVERSIDADE REGIONAL DE BLUMENAU, 2004, p. 9) descreve que o acadêmico deve primar pela autenticidade da autoria de seu TCC e pela veracidade técnico-científica dos dados, cuja falsificação é passível de sanções administrativas e legais.

A apresentação pública somente é realizada se o acadêmico obtiver aprovação preliminar de cada um dos membros da banca examinadora (nota igual ou superior a 6,0), que deve ser encaminhada a coordenação de TCC através de formulário conforme ANEXO B e representa 40% da nota final.

A avaliação da banca examinadora com relação à apresentação pública corresponde a 25% da nota final em relação à defesa e 30% em relação à implementação, conforme a ficha de avaliação demonstrada no ANEXO C. Os outros 5% é uma nota atribuída pelo professor coordenador de TCC-II.

A nota final do TCC está condicionada à entrega formal do mesmo, após a apresentação pública, com as devidas correções, se houver. O formulário apresentado no ANEXO D deve ser preenchido, onde são somadas todas as médias para se obter a nota final, e se a nota final for igual ou superior a 6,0 (seis) o aluno é considerado aprovado.

## 2.2 PROCESSO UNIFICADO

Este trabalho utiliza o método UP para atingir seu objetivo, que é a modelagem e a implementação de um protótipo de um sistema para acompanhamento de TCCs. O processo unificado de desenvolvimento de software é o conjunto de atividades necessárias para transformar requisitos do usuário em um sistema de software (MARTINS, 1999, p. 1).

O UP comporta, em suas recomendações, as antigas fases de estudo de viabilidade, análise de requisitos, análise de domínio e o projeto em múltiplas camadas. Contudo, estas fases aparecem no UP organizadas de modo diferente. As fases do UP são: concepção, elaboração, construção e transição (WAZLAWICK, 2004, p. 23).

### 2.2.1 Concepção

O objetivo principal da fase de concepção é delimitar o escopo do projeto, definido como o sistema será utilizado por cada usuário. A partir deste escopo poderá ser definido o que o desenvolvimento do projeto deverá cobrir.

Além disso, o esforço empenhado na fase de concepção poderá evitar o fracasso do projeto através da identificação prévia de riscos. A causa do fracasso de vários projetos é o fato de riscos críticos serem encontrados tarde demais (SEABRA JÚNIOR, 2001, p. 41).

Para Waslawick (2004, p. 32), a fase de concepção consiste em uma etapa na qual o analista vai buscar as primeiras informações sobre o sistema a ser desenvolvido. Nessa etapa assume-se pouco conhecimento do analista sobre o sistema e uma grande interação com o usuário.

As atividades da fase de concepção serão apresentadas em três partes:

- a) levantamento de requisitos;
- b) organização dos requisitos;
- c) planejamento do desenvolvimento.

Segundo Waslawick (2004, p. 33), a etapa de levantamento de requisitos corresponde a buscar junto ao usuário, seus sistemas e documentos, todas as informações possíveis sobre as funções que o sistema deve executar, levando o analista a produzir os seguintes documentos:

- a) visão geral do sistema: deve descrever as principais idéias do cliente sobre o sistema. O analista deve ter em mente que este documento descreve as principais preocupações do cliente, os quais serão mais bem estruturados nas fases posteriores do processo de análise;
- b) requisitos funcionais e não funcionais: os requisitos funcionais correspondem à listagem de todos os tópicos relativos ao que o sistema deve fazer. Os requisitos não funcionais são restrições colocadas sobre como o sistema deve realizar seus requisitos funcionais.

Conforme Rosa (2005, p. 37), os requisitos funcionais especificam as principais características de camada de negócio do sistema. Estes requisitos descrevem os serviços que o sistema deve oferecer, como deve reagir a interação com o usuário e o comportamento do sistema em determinadas situações.

Uma vez que os requisitos tenham sido levantados, cabe agora organizá-los. Conforme Waslawick (2004, p. 44), os requisitos podem ser agrupados então do seguinte modo:

- a) casos de uso: é necessário identificar os grandes processos de negócio, também chamados de casos de uso. Cada caso de uso será associado a um conjunto de requisitos funcionais do sistema;
- b) conceitos: os conceitos envolvidos com o sistema possivelmente sofrerão operações de manutenção ou cadastro. Em geral, estas operações são: inserir; alterar; remover e consultar. Nem sempre estas operações de manutenção de informação vão aparecer nos casos de uso, sugere-se então, que o analista procure elaborar um modelo conceitual preliminar, que é segundo Souza (2005), composto de conceitos com atributos que mostram como a informação está organizada e de que forma pode ser encontrada ou mantida;
- c) consultas: a geração de relatórios não precisa ser considerada caso de uso, basta definir o formato em que a informação deverá ser exibida e definir um modo padrão para o usuário gerar relatórios.

Os ciclos iterativos fazem parte da atividade do planejamento do desenvolvimento e são miniprojetos que se baseiam em casos de uso, conceitos e consultas. Estes miniprojetos devem ser curtos e com tempo predeterminado. Recomenda-se que o cronograma tenha certa folga, porque vários requisitos novos serão fatalmente descobertos ao longo dos ciclos iterativos e deverão ser acomodados à medida que o desenvolvimento se mantém. Portanto, ao longo dos ciclos iterativos o conhecimento sobre o sistema vai aumentando e grande parte da arquitetura já vai sendo definida e implementada. Desta forma, os processos do trabalho vão ficando cada vez mais sistemáticos (WAZLAWICK, 2004, p. 51).

### 2.2.2 Elaboração

Segundo Falbo (2000, p. 4), o propósito desta fase é analisar mais refinadamente o domínio do problema, estabelecer uma arquitetura de fundação sólida, desenvolver um plano de projeto para o sistema a ser construído e eliminar os elementos de projeto que oferecem maior risco. Embora o processo deva sempre acomodar alterações, as atividades da fase de elaboração asseguram que os requisitos, a arquitetura e os planos estão suficientemente estáveis e que os riscos estão suficientemente mitigados, de modo a se poderem prever com precisão os custos e prazos para a conclusão do desenvolvimento.

A fase de elaboração se inicia com a subfase de análise e prossegue com a subfase de projeto. A subfase de análise em si comporta três atividades:

- a) expansão dos casos de uso;
- b) construção do modelo conceitual;
- c) elaboração dos contratos das operações de sistema.

Quando se está expandindo um caso de uso é preciso proceder a um exame detalhado do processo de negócio, descrevendo-se o caso de uso passo a passo: como ele ocorre; como é a interação entre o usuário e o sistema.

A atividade de expansão dos casos de uso consiste, segundo Wazlawcik (2004), em realizar o detalhamento dos casos de uso associados ao ciclo iterativo em andamento. Estes casos de uso, que na fase de concepção foram apenas listados ou descritos em alto nível, agora devem ser detalhados em uma seqüência de passos capaz de incluir todas as possibilidades de interações possíveis de serem identificadas pelos analistas.

Esta expansão constitui-se basicamente de:

- a) identificar a seqüência de passos principais (fluxo principal);
- b) identificar as seqüências alternativas associadas às possíveis exceções, ou seja, os fluxos específicos para tratamento de exceções.

Para representar a seqüência dos eventos de sistema em um cenário de um caso de uso a *Unified Modeling Language* (UML) possui o diagrama de seqüência que tem como elementos, instâncias de atores, representados por figuras humanas esquematizadas, e instâncias de objetos constituintes do sistema.

Uma coisa a se ter em mente sempre quando se constrói tais diagramas é que a informação normalmente não é criada durante estes processos, mas apenas transferida ou transformada.

O modelo conceitual deve descrever, conforme Wazlawcik (2004, p. 102), a informação que o sistema vai gerenciar, visando representar a compreensão da informação e não a sua representação física. O modelo conceitual é uma representação da visão que o usuário tem das situações gerenciadas pelo sistema.

O processo de descoberta dos elementos do modelo conceitual é olhar para os casos de uso expandidos e tentar descobrir todos os elementos textuais que eventualmente referenciam informações a ser guardada e/ou processada.

O processo de criação de contratos está diretamente ligado ao caso de uso expandido e ao modelo conceitual. Deve-se usar o modelo conceitual como referência em todos os momentos porque ele é a fonte de informação, os contratos devem ser sempre escritos como expressões interpretáveis.

Segundo Seabra Júnior (2001, p. 36), ao contrário da fase de análise, cujo produto é

um modelo que descreve características comportamentais e estruturais do sistema em um nível conceitual, a fase de projeto desenvolve o modelo de projeto que descreve o sistema em um nível físico, tendo como principal função obter uma compreensão detalhada dos requisitos do sistema, levando em consideração fatores como linguagem de programação, sistemas operacionais e tecnologias de banco de dados.

Enquanto a análise se interessa por o que o sistema deve fazer, o projeto diz respeito a como os requisitos serão implementados. Desta forma, o modelo de projeto funciona como uma abstração da implementação.

A subfase de projeto, dividi-se em primeiro lugar na camada de domínio do software, depois na camada de persistência, interface e segurança, que são derivadas ou dependentes da camada de domínio. O projeto da camada de domínio compõe-se basicamente de duas atividades:

- a) diagrama de colaboração;
- b) diagrama de classes.

### 2.2.3 Construção

A fase de construção corresponde à implementação e testes, visando produzir uma solução para o problema agora claramente identificado pela análise. O problema está especificado nos diagramas de seqüência dos casos de uso e no modelo conceitual (WAZLAWICK, 2004, p. 23).

Enquanto as fases de concepção e elaboração estão ligadas diretamente a modelagem, a fase de construção é caracterizada pelo desenvolvimento, isso é a construção concreta de um sistema (SEABRA JÚNIOR, 2001, p. 47).

Uma vez definidos os diagramas de colaboração e o diagrama de classes de projeto, a geração de código é uma tarefa passível de automatização. Trata-se aqui de gerar o código das classes correspondentes a camada de domínio da aplicação, ou seja, as classes que realizam toda a lógica do sistema a partir das operações e consultas de sistema. Uma vez gerada a camada de domínio, as demais camadas serão derivadas e dependentes desta (WAZLAWICK, 2004, p. 233).

A camada de interface pode ser dividida em duas subcamadas:

- a) apresentação: uma camada com as classes que representam os objetivos gráficos de interface e cujas responsabilidades se resumem a receber dados e comandos do

usuário e apresentar resultados a ele;

- b) aplicação: uma camada que controla a lógica de interface, abrindo e fechando janelas, habilitando e desabilitando botões entre outros.

Até aqui, tanto a camada de domínio quanto a camada de interface foram especificadas como se não houvesse restrições de segurança de acesso e como se todos os usuários pudessem realizar todas as funções do sistema. Contudo, isso normalmente não é o caso. É preciso, portanto, retornar ao documento de requisitos para verificar quais são os níveis de segurança existentes no sistema e que funções podem ser realizadas em cada nível.

A fase de transição ocorre quando o sistema, depois de pronto, será implementado na empresa, substituindo o sistema atual, seja ele manual ou computadorizado. Segundo Martins (2005), o objetivo desta fase é transferir o produto para a comunidade de usuários, garantir que tenha o nível de qualidade esperado e atenda aos requisitos especificados, corrigir erros, fazer a migração de dados, treinar usuários, fazer ajustes no sistema e adicionar elementos que foram esquecidos.

## 2.3 SEGURANÇA

O ambiente Web está longe de ser considerado seguro e, nem com todos os esforços concentrados em segurança na Web, a Internet pode ser considerada 100% segura, mas assumir que a Internet não é um ambiente propício para a disseminação de sistemas de informação é não acompanhar a evolução e seus benefícios (SICA;REAL, 2007).

É neste ambiente adverso que se torna necessário o estudo de técnicas de programação segura, visando diminuir ao máximo uma invasão.

Todas as falhas conhecidas se apóiam em falhas de programação ou na engenharia social. As falhas causadas por engenharia social (SILVA, 2008), não se baseiam em conceitos técnicos e por isso são mais difíceis de serem barradas. A melhor maneira de ficar longe de problemas é ter cautela enquanto se está navegando pela internet, não dar informações a qualquer pessoa e passar a adotar regras e políticas de segurança mesmo em seus computadores pessoais.

As falhas de programação são exploradas por pessoas com grande conhecimento técnico, com o único objetivo de conseguir acesso não autorizado aos programas ou danificar o seu banco de dados.

Para que os recursos de segurança sejam maximizados, os fatores que envolvem o sistema operacional, o servidor HTTP, o servidor de banco de dados e os *scripts* PHP, devem ser feitos de forma conjunta.

Alguns fatos relacionados à segurança na camada da rede ou plataforma de execução do PHP podem comprometer a execução e a segurança do PHP em si. Para o bom funcionamento do sistema como um todo, os seguintes cuidados são citados como básicos (SICA;REAL,2007):

- a) manter o sistema operacional sempre atualizado seja ele Windows ou Linux;
- b) manter ativo e devidamente configurado *firewall*, antivírus e *Intrusion Detecting System* (IDS);
- c) não manter o sistema operacional ativo como administrador (*root*), salvo em casos de manutenção ou configuração;
- d) limitar ao máximo o usuário para a utilização do sistema operacional.

Alguns métodos básicos para aumentar sensivelmente a segurança do banco de dados MySQL são:

- a) permitir apenas acessos locais ao MySQL, evitando conexões remotas;
- b) a conta *root* do MySQL deve possuir uma senha difícil de ser quebrada;
- c) a conta administrador deve ser renomeada;
- d) o acesso anônimo (*via nobody*) deve ser desativado;
- e) todos os bancos e tabelas de exemplos devem ser apagados.

Ao se preparar o ambiente, outros fatores são importantes. O PHP precisa estar configurado para que as variáveis superglobais fiquem desativadas (*register\_globals = off*) e executar somente no modo de segurança (*safe\_mode = on*). Estas configurações têm que ser alteradas de forma manual no arquivo *php.ini*. Deve-se também alterar o acesso do PHP aos arquivos (*open\_basedir*), configurando a árvore de diretório.

Para arquivos de include, extensão *.inc*, qualquer navegador cliente (*browser*) que fizer uma requisição ao servidor por este arquivo terá todo o conteúdo do mesmo exibido na tela, porque o servidor só irá interpretar aquilo para o que foi configurado. Para que isso não ocorra, deve-se renomear com a extensão *.php*, desta forma o arquivo sempre será processado antes de ser entregue para os usuários.

Para criptografia, o PHP oferece um grande suporte por meio de funções internas, como por exemplo, *base64()* e *mdecrypt()* que trabalham em parceria, permitindo a criptografia e a descryptografia de dados: chaves públicas e chaves secretas.

Fornecer suporte também ao algoritmo de *hashing* utilizando a biblioteca pública

`mhash`. Como exemplo, existe a função `crypt()` e `md5()`, que não traz uma parceria para descriptografar os dados e se aplica rotineiramente na criptografia de senhas. Esta técnica é utilizada porque para validar senha, não é necessário descriptografar, é só comparar a senha fornecida criptografada com a armazenada.

Quando se faz necessário gravar algum dado muito sigiloso, por exemplo, o número de um cartão de crédito, a aplicação de criptografia ou *hashing* não é suficiente, pois o PHP é uma linguagem executada do lado servidor e, naturalmente, o envio destes dados do computador cliente até o servidor estará desprotegido. Para solucionar este problema o ideal é utilizar, para esse tipo de comunicação, o protocolo *Secure Server Layer* (SSL).

### 3 DESENVOLVIMENTO DO PROTÓTIPO

Nesta seção é abordado o desenvolvimento do protótipo proposto, seguindo as quatro fases do processo unificado. Na primeira seção é apresentado os requisitos principais do problema a ser trabalhado, que faz parte da fase de concepção.

A segunda seção trata da especificação que é incluída na fase de elaboração, Na terceira seção a implementação ou fase de construção, depois é apresentada a operacionalidade que faz parte do projeto de interface e por ultimo a fase de transição.

#### 3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O processo unificado na fase de concepção apresenta três sub-etapas: levantamento de requisitos, organização dos requisitos e planejamento do desenvolvimento.

##### 3.1.1 Levantamento de requisitos

Os artefatos da fase de levantamento de requisitos são: visão geral do sistema, requisitos funcionais e não funcionais. No quadro 1 é apresentada a visão geral do sistema para acompanhamento de TCCs, que será usado no processo de desenvolvimento ao longo deste trabalho.

#### **Protótipo de um sistema para acompanhamento de TCCs**

##### Visão Geral do Sistema

É proposto o desenvolvimento de um protótipo de um sistema para acompanhamento de TCCs para a FURB, que vai informatizar as funções de administrar e supervisionar a elaboração dos TCCs do curso de ciência da computação e coordenar as apresentações. O objetivo do sistema é possibilitar ao coordenador supervisionar todo o processo de forma eficiente, ao mesmo tempo agilizar o trâmite de informações importantes as pessoas envolvidas, possibilitar ainda, um melhor controle das informações por parte da coordenação e garantir maior segurança. Deverão ser gerados relatórios de situação dos alunos, lista de presença, publicação das propostas e informações sobre a apresentação. O sistema deverá calcular automaticamente a nota final do aluno.

**Quadro 1** – Visão geral do sistema

Os requisitos deste projeto foram levantados e definidos com a participação do professor e coordenador de TCC do curso de ciência da computação da FURB, onde foram identificadas as necessidades do professor em relação á aplicação do sistema.

O quadro 2 representa o requisito funcional registrar situação do aluno e seus requisitos não funcionais associados. O quadro 3 representa o requisito funcional registrar banca e seus requisitos não funcionais associados. O quadro 4 representa o requisito funcional calcular média e seus requisitos não funcionais associados.

F1 Registrar situação do aluno		<b>Oculto ( )</b>		
<b>Descrição:</b> O sistema deve registrar a atividade que o aluno esta desenvolvendo e em que situação se encontra até o momento.				
<b>Requisitos Não Funcionais</b>				
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Desejável</b>	<b>Permanente</b>
NF1.1 Controle de Acesso	A função só pode ser acessada por usuário com perfil de administrador.	Segurança	( )	( )
NF1.2 Identificação do aluno	O aluno deverá ser identificado a partir de seu curso, disciplina, turma e nome	Interface	( )	( )
NF1.3 Janela única	Todas as funções relacionadas a situação do aluno devem ser efetuadas em uma única janela	Interface	(x)	( )
...	...	...	...	...

Quadro 2 – Requisito funcional registrar situação do aluno e não funcionais associados

F2 Registrar banca		<b>Oculto ( )</b>		
<b>Descrição:</b> O sistema deve registrar a data a hora e a sala da apresentação, bem como os professores que avaliaram a banca e suas respectivas notas.				
<b>Requisitos Não Funcionais</b>				
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Desejável</b>	<b>Permanente</b>
NF2.1 Controle de Acesso	A função só pode ser acessada por usuário com perfil de administrador.	Segurança	( )	( )
NF2.2 Identificação do aluno	A banca deverá ser identificada a partir da monografia	Interface	( )	( )
NF2.3 Janela única	Todas as funções relacionadas a situação do aluno devem ser efetuadas em uma única janela	Interface	(x)	( )
...	...	...	...	...

Quadro 3 – Requisito funcional registrar banca e não funcionais associados

F3 Calcular média		<b>Oculto ( x )</b>		
<b>Descrição:</b> O sistema deve calcular média final do TCC.				
<b>Requisitos Não Funcionais</b>				
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Desejável</b>	<b>Permanente</b>
NF3.1 Regras do Regulamento	Para calcular a média deve se levar em consideração as regras do regulamento.	Especificação	( )	( )
...	...	...	...	...

Quadro 4 – Requisito funcional calcular média e não funcional associado

O quadro 5 representa os requisitos suplementares.

Nome	Restrição	Categoria	Desejável	Permanente
S1 Tipo de Interface	As interfaces do sistema devem ser implementadas como formulários acessíveis em um browser html.	Interface	( )	( )
S2 Armazenamento de dados	A camada de persistência deve ser implementada de forma que diferentes tecnologias de bancos de dados possam vir a ser utilizadas no futuro	Persistência	( )	( x )
S3 Perfis de usuário	Os perfis de usuário para acesso ao sistema são: 1. Administrador - pode efetuar todas as operações. 2. Usuário - pode efetuar apenas consultas.	Segurança	( )	( )
...	...	...	...	...

Quadro 5 – Requisitos suplementares

### 3.1.2 Organização dos requisitos

Uma vez que os requisitos tenham sido levantados, agora podem então serem agrupados em casos de uso, em conceitos e em consultas. O Quadro 6 representa a listagem de casos de uso da fase de concepção. O Quadro 7 representa uma tabela com os principais conceitos identificados, indicando se estes conceitos devem sofrer inserção (I), alteração (A), exclusão (E) ou consulta (C). O Quadro 8 representa uma consulta que combina informações da tabela de alunos, de atividade e de situação para posterior construção.

Nome	Atores	Descrição	Referências Cruzadas
Registrar Situação do aluno	Coordenador	O coordenador atualiza situação conforme atividade realizada.	
Registrar Banca	Coordenador	O coordenador registra os dados da apresentação e atualiza as notas dadas pelos avaliadores.	
Calcular Média	Sistema	O sistema calcula a média final.	F2

Quadro 6 – Listagem de casos de uso da fase de concepção

Conceito	I	A	E	C	Observação	Ref. Cruzadas
Aluno	x	x	x	x		F1
Banca	x	x	x	x		F2,F3

Quadro 7 – Listagem de conceitos e operações de manutenção

Nome	Referências Cruzadas
Situação do aluno	F1
...	...

Quadro 8 – Listagem de consulta

### 3.1.3 Planejamento de desenvolvimento

Tomou-se uma parte importante do sistema e fez-se análise, projeto, implementação e testes, dividindo cada parte em ciclos iterativos, sendo cada parte um subconjunto dos casos de uso, operações de manutenção de informações e/ou consultas. Para o trabalho proposto, o processo de desenvolvimento foi planejado de acordo com o Quadro 9.

Ciclo	Casos de Uso	Manutenção de Informações	Consultas	Observações
1	Usuário, Permissão			-
2	Registrar situação do aluno	Curso, Disciplina, Turma Atividade, situação, Aluno		-
3	Registrar Banca	Professor, Proposta Monografia	Alunos por atividade e situação	-

Quadro 9 – Planejamento dos ciclos iterativos

Para este caso foi propostos 3 ciclos. Supondo, por exemplo, que cada ciclo tenha um esforço estimado de 400 horas de trabalho, e que se deseja desenvolver este projeto em cerca de 150 dias, entre os meses de agosto e dezembro, então chegou-se a um cronograma conforme mostrado no Quadro 10. Neste cronograma supõe-se que a fase de implantação, ocorrerá no final do projeto.

Dias:	1-25	26-50	51-75	76-100	101-125	126-150	151-175
Ciclo 1	Análise	Projeto	Impl.	Testes			
Ciclo 2		Análise	Projeto	Impl.	Testes		
Ciclo 3			Análise	Projeto	Impl.	Testes	
Implantação							Implant.

Quadro 10 – Cronograma para o desenvolvimento do sistema

## 3.2 ESPECIFICAÇÃO

Para a especificação deste trabalho, foram utilizados os diagramas da UML: de casos de uso, modelo conceitual e diagrama de classes apresentados nas seções seguintes, através das ferramentas *Computer Aided Software Engineering (CASE)*, *DBDesigner* na sua versão 4.0.5.6 Beta e *StarUML* (versão 5.0.2.1570).

De acordo com a necessidade do processo unificado que na fase de elaboração realiza três atividades: expansão dos casos de uso, construção do modelo conceitual e elaboração dos contratos das operações do sistema, na subfase de análise e na subfase de projeto definiram-se

o diagrama de classes.

### 3.2.1 Expansão do caso de uso

O Quadro 11 apresenta a expansão do caso de uso Registrar situação do aluno. O Quadro 12 apresenta a expansão do caso de uso Registrar banca, que estão relacionados ao perfil Administrador.

<b>Caso de uso:</b> Registrar situação do aluno
<b>Atores:</b> Coordenador
<b>Precondições:</b> O usuário deverá possuir um cadastro ativo no sistema e deverá estar logado. O usuário deverá selecionar uma atividade, um aluno e uma situação já existente para o registro.
<b>Pós-condições:</b> Situação do aluno registrada
<b>Fluxo principal:</b> 1. O sistema apresenta o menu de opções. 2. O usuário clica na opção Cadastros/Situação do aluno. 3. O usuário informa o aluno, a atividade e em que situação ele se encontra e clica no botão confirmar. 4. O sistema registra uma nova situação referente a uma atividade de um aluno.
<b>Tratamento de exceções:</b> 3a. O aluno não está cadastrado. 3a.1. O usuário deverá registrar o aluno. 3b. Atividade não está cadastrada. 3b.1. O usuário deverá registrar atividade. 3c. Situação não está cadastrada. 3c.1. O usuário deverá registrar atividade.

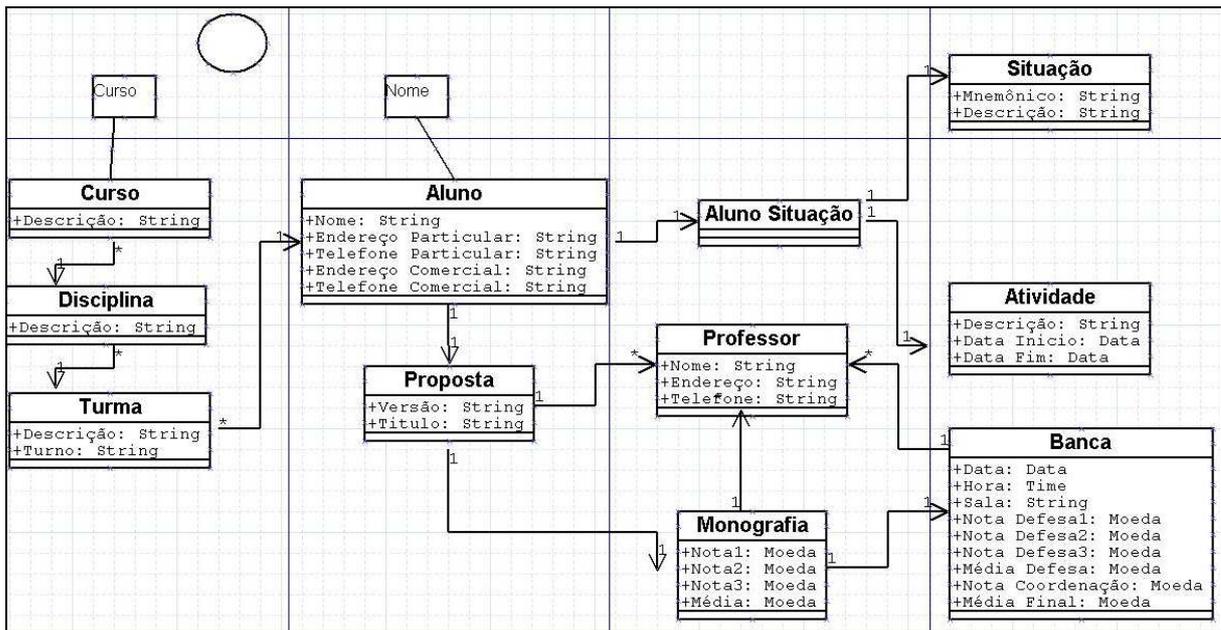
Quadro 11 – Caso de uso expandido registrar situação do aluno

<b>Caso de uso:</b> Registrar banca
<b>Atores:</b> Coordenador
<b>Pré-condições:</b> O usuário deverá possuir um cadastro ativo no sistema e deverá estar logado. O usuário deverá selecionar monografia e professor já existente para o registro.
<b>Pós-condições:</b> Banca registrada
<b>Fluxo principal:</b> 1. O sistema apresenta o menu de opções. 2. O usuário clica na opção Cadastros/Banca. 3. O usuário preenche as informações conforme a necessidade, monografia, data, hora, sala, avaliadores e notas. Clica no botão calcular média e clica no botão confirma. 4. O sistema calcula média. 5. O sistema registra uma nova banca.
<b>Tratamento de exceções:</b> 3a. A monografia não está cadastrada. 3a.1. O usuário deverá registrar a monografia. 3b. Professor não está cadastrado. 3b.1. O usuário deverá registrar o professor.

Quadro 12 – Caso de uso expandido registrar banca

### 3.2.2 Modelo conceitual

As informações trocadas entre o sistema e o mundo externo, conforme a expansão do caso de uso, serão usadas para construir a base do modelo conceitual. O modelo conceitual do trabalho é representado como no Quadro 13.



Quadro 13 – Modelo conceitual

### 3.2.3 Contratos

Cada operação ou consulta, implica a existência de uma intenção por parte do usuário, a qual é capturada pelos contratos de operação de sistema e nos contratos de consulta de sistema. O quadro 14 apresenta um contrato para cadastramento do aluno, referente operação de inserção. O quadro 15 apresenta um contrato para alteração do aluno. O quadro 16 apresenta um contrato para exclusão do aluno. O quadro 17 apresenta um contrato para operação de consulta de aluno.

Classe Aluno

operação: cadastraAluno(nomeAluno, enderecoAluno, telefoneAluno:String )

pré:

Não existe nenhum Aluno com nome = nomeAluno.

pós:

Foi criado um aluno e adicionada ao cadastro.

Os atributos nome, endereço, telefone foram alterados para nomeAluno, enderecoAluno e telefoneAluno.

Quadro 14 – Contrato para operação de cadastro de novo aluno

Classe Aluno

operação: alteraAluno(nomeAluno, enderecoAluno, telefoneAluno:String)

pré:

Existe um Aluno com nome = nomeAluno.

pós:

Os atributos endereço e telefone do Aluno foram alterados para enderecoAluno e telefoneAluno.

Quadro 15 – Contrato para operação de alteração de aluno

Classe Aluno

operação: excluirAluno(nomeAluno:String)

pré:

Existe um Aluno com nome = nomeAluno.

pós:

O aluno com nome = nomeAluno foi destruído.

Quadro 16 – Contrato para operação de exclusão de aluno

Classe Aluno

operação: consultaAluno(nomeAluno:String)

pré:

Existe um Aluno com nome = nomeAluno.

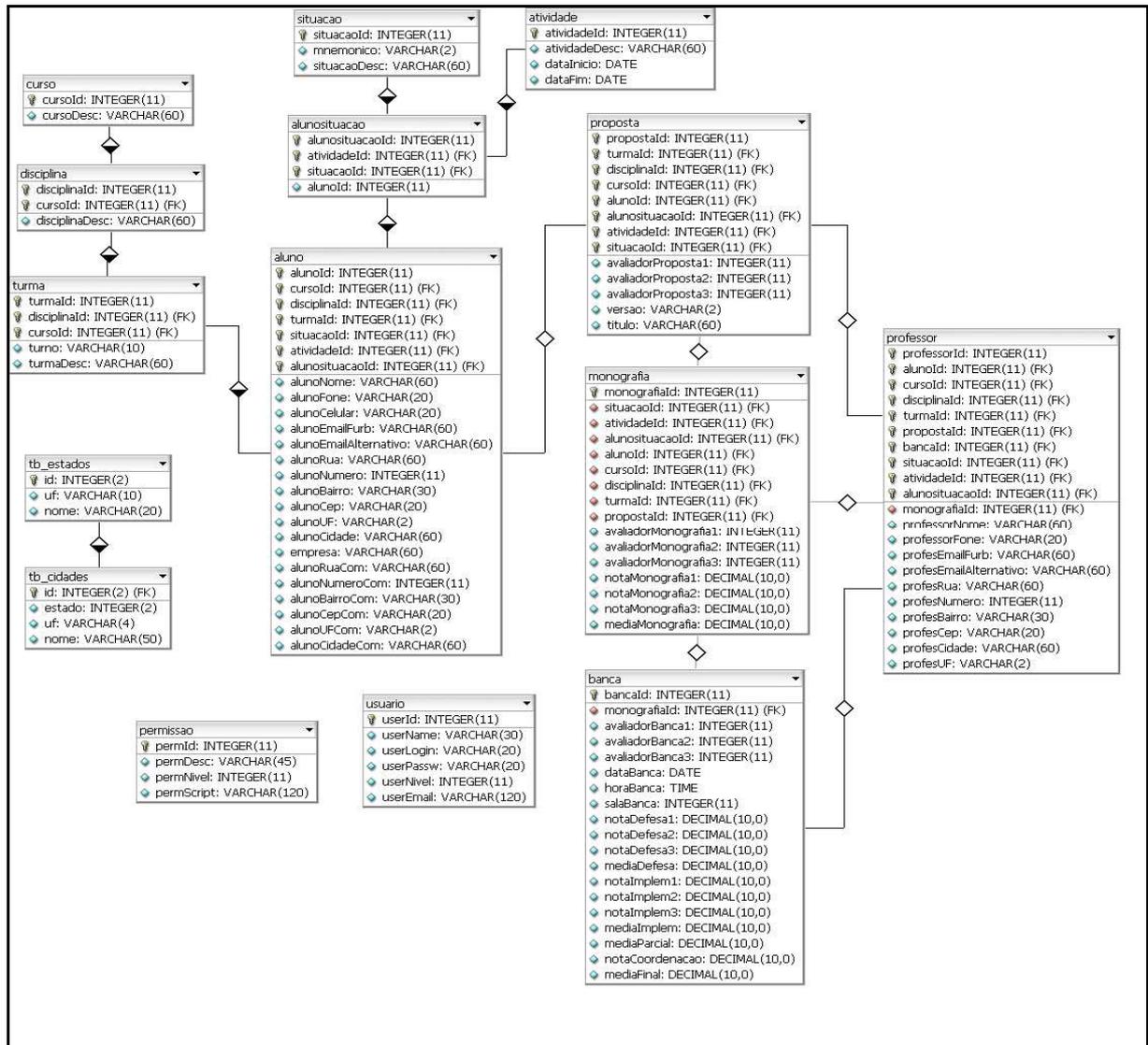
pós:

Retorna o endereço e o telefone do Aluno.

Quadro 17 – Contrato para operação de consulta de aluno

### 3.2.4 Diagrama de classes

As interações realizadas pelos usuários com o sistema através da camada de aplicação são controladas internamente através das classes, as quais também interagem entre si. A seguir são representadas as interações entre as classes do sistema, bem como, os seus atributos e métodos. O quadro 18 representa o diagrama de classes, referentes à camada de modelo do sistema.



Quadro 18 – Diagrama de classes

### 3.3 IMPLEMENTAÇÃO

O sistema aqui desenvolvido consiste em uma aplicação *Web*, a qual foi concebida utilizando-se a linguagem PHP, na sua versão 5.3.0, funções *JavaScripts*, *framework* JQuery e estilo de página CSS. O servidor *Web* utilizado foi o Apache na versão 2.2.11.

A implementação das classes e das páginas do sistema foram realizadas através da ferramenta PHP Editor 2.22, enquanto a base de dados foi construída a partir das ferramentas phpMyAdmin e HeidiSQL.

Conforme o processo unificado na fase de construção é gerado o código das classes correspondentes à camada de domínio da aplicação.

#### 3.3.1 Geração de código

Conforme já citado no início desta seção, a base de dados da aplicação foi construída em plataforma MySQL. A sua escolha deve-se, entre outros motivos, a sua portabilidade, confiabilidade, desempenho e multiplataforma (em nível de sistema operacional); além de ser um banco de dados *open source*.

Para facilitar o desenvolvimento criou-se uma interface genérica de acesso ao banco de dados, desta forma qualquer alteração ou implementação de um novo tipo de banco de dados pode ser realizada em um único lugar.

Foram criadas duas classes, a primeira responsável somente pela conexão, será abstrata, não pode ser instanciada, deve ser herdada por outra classe e ter os métodos construtor da classe, de conexão e de retorno dos dados da conexão. Já a segunda classe, deve conter toda a interação com o banco de dados, sendo responsável por preparar a consulta, enviá-la ao servidor e gerenciar o resultado.

A conexão como o banco de dados pode ser realizada quando a classe filha for instanciada (`new consulta("mysql")`), ou então após a classe filha ser instanciada, por meio da chamada ao método `conecta()` da classe. O único atributo que é obrigatoriamente utilizado na criação do objeto é o tipo de banco de dados, ou seja, `new consulta()` instância a classe para o tipo de banco de dados `mysql`, conforme mostra o Quadro 19 referente ao fragmento de código responsável pela definição desta.

A classe `consulta` herda a classe `bd`, implementando a função abstrata `setAffectedRows`, além de implementar as funções pertinentes a esta classe, bem como seus atributos. O Quadro 20 demonstra um fragmento deste código.

A classe `banca`, convertida em classe na linguagem de programação e seus atributos convertidos em variáveis de instância (sempre privadas). O Quadro 21 demonstra um fragmento deste código.

```
<?php
// Classe Abstrata BD
abstract class bd {
    protected $_bd = "mysql"; // mysql, mysqli, postgresql
    protected $_bdtype = 0; // 0=MySQL, 1=PostgreSQL, 2 = mysqli
    protected $_connection = NULL;
    protected $_affected_rows = 0;
    protected $_servidor = "localhost";
    protected $_bdname = NULL;
    protected $_usuario = "root";
    protected $_senha = "";
    protected $_porta = NULL;
    protected $_extras = NULL;
    protected $_last_error = -1;
// Funções abstratas
abstract protected function setAffectedRows();
// Contrutor
function __construct($_bdtype="mysql", $_srv="", $_port="", $_bd="", $_usr="", $_
_pwd="", $_extras="") {
    $_bdtype = strtolower($_bdtype);
    if($_bdtype!="mysql"&&$_bdtype!="postgresql"&&$_bdtype!="mysqli") {
        return FALSE; }
    $this->_bd = $_bdtype;
    switch($_bdtype) {
        case 'mysql':           $this->_bdtype = 0;
        break;
        case 'postgresql':     $this->_bdtype = 1;
        break;
        case 'mysqli':         $this->_bdtype = 2;}
    if($_srv!="") {
        // banco, usuario devem ser informados tb...
        if($_bd!=""&&$_usr!="") {
            $_r = $this->conecta($_srv, $_port, $_bd, $_usr, $_pwd, $_extras);
            if($_r===FALSE) {
                return FALSE;
            }
        }
    }
}
} ...
```

Quadro 19 – Exemplo de definição da classe abstrata BD

```

// Classe Consulta
class consulta extends bd {
    protected $_res = NULL;
    protected $_sql = "";
    protected $_dados = Array();
    protected $_tipo_res = 0; // 0 - array, 1 - objeto
    protected $_num_rows = -1;
    private $_atual = -1;
    // Construtor da classe

function__construct($_bdtype="mysql",$_srv="",$_port="",$_bd="",$_usr="",$_
_pwd="",$_extras="", $_sql="") {
    parent::__construct($_bdtype,$_srv,$_port,$_bd,$_usr,$_pwd,$_extras);
    if($_sql!="") {
        $this->executa($_sql);
    }
}
// Funções privativas
private function moveponteiro($_ponteiro) {
    switch($this->_bdtype) {
        case 0: $_r = mysql_data_seek($this->_res,$_ponteiro);
            break;
        case 1: $_r = pg_result_seek($this->_res,$_ponteiro);
            break;
        case 2: $_r = $this->_res->data_seek($_ponteiro);
            break;
    }
    if($_r!==FALSE) {
        $this->_atual = $_ponteiro;
    } else {
        return FALSE;
    }
}
// Funções protegidas
protected function setAffectedRows() {
    switch($this->_bdtype) {
        case 0: $this->_affected_rows = mysql_affected_rows($this->_connection);
            break;
        case 1: $this->_affected_rows = pg_affected_rows($this->_res);
            break;
        case 2: $this->_affected_rows = $this->_connection->affected_rows;
            break;
    }
}
protected function setNumRows() {
    if($this->_res===NULL) {
        return FALSE;
    }
    switch($this->_bdtype) {
        case 0: $this->_num_rows = mysql_num_rows($this->_res);
            break;
        case 1: $this->_num_rows = pg_num_rows($this->_res);
            break;
        case 2: $this->_num_rows = $this->_res->num_rows;
            break;
    }
}
}
}

```

Quadro 20 – Exemplo de definição da classe consulta

```

<?php
// Definição das interfaces
require_once("interfaces.inc.php");
// Classe banca
class banca implements ITabela {
    private $_bd = NULL;
    private $_bancaId = NULL;
    private $_monografiaId;
    private $_avaliadorBanca1;
    private $_avaliadorBanca2;
    private $_avaliadorBanca3;
    private $_dataBanca;
    private $_horaBanca;
    private $_salaBanca;
    private $_notaDefesa1;
    private $_notaDefesa2;
    private $_notaDefesa3;
    private $_notaImplement1;
    private $_notaImplement2;
    private $_notaImplement3;
    private $_notaCoordenacao;
    private $_mediaDefesa;
    private $_mediaImplement;
    private $_mediaFinal;
    // Construtor da classe
    function __construct($_bd) {
        if($_bd===NULL) {
            return FALSE;
        } else {
            $this->_bd = $_bd;
            return TRUE;
        }
    }
    private function preenche() {
        $_dados = $this->_bd->getDados();
        if($_dados===FALSE) {
            return FALSE;
        }
        $_dados = array_change_key_case($_dados,CASE_LOWER);
        $novaData = explode("-",$_dados["databanca"]);
        $bdData = $novaData[2].'/'.$novaData[1].'/'.$novaData[0];
        $this->_bancaId = (int) $_dados["bancaid"];
        $this->_monografiaId = $_dados["monografiaid"];
        $this->_avaliadorBanca1 = (int) $_dados["avaliadorbanca1"];
        $this->_avaliadorBanca2 = (int) $_dados["avaliadorbanca2"];
        $this->_avaliadorBanca3 = (int) $_dados["avaliadorbanca3"];
        $this->_dataBanca = $bdData;
        $this->_horaBanca = $_dados["horabanca"];
        $this->_salaBanca = $_dados["salabanca"];
        $this->_notaDefesa1 = $_dados["notadefesa1"];
        $this->_notaDefesa2 = $_dados["notadefesa2"];
        $this->_notaDefesa3 = $_dados["notadefesa3"];
        $this->_notaImplement1 = $_dados["notaimplement1"];
        $this->_notaImplement2 = $_dados["notaimplement2"];
        $this->_notaImplement3 = $_dados["notaimplement3"];
        $this->_notaCoordenacao = $_dados["notacoordenacao"];
        $this->_mediaDefesa = $_dados["mediadefesa"];
        $this->_mediaImplement = $_dados["mediaimplement"];
        $this->_mediaFinal = $_dados["mediafinal"];
        return TRUE;
    }
}

```

Quadro 21 – Exemplo de definição da classe banca

### 3.3.2 Interface

A etapa de implementação da interface foi transformar um *layout* previamente definido em páginas navegáveis, através de *links* que se relacionam entre si, composta por HTML (conteúdo), mais CSS (apresentação) e JavaScript (comportamento) .

Apesar da maioria dos navegadores dar suporte ao padrão W3C, pode haver diferença na visualização da mesma página entre navegadores diferentes, devido à falta de padronização, particularidades que cada navegador tem, e também pelas diversas resoluções de tela que o usuário pode estar utilizando.

Através do Quadro 22, onde são demonstrados os estilos aplicados aos botões e aos títulos pode-se perceber também a padronização resultante da utilização do CSS, uma vez que, a partir destas configurações, estes botões serão idênticos em todas as páginas da aplicação. Além da praticidade e agilidade em relação a sua manutenção, caso necessário, pois, bastaria alterar o arquivo de configuração CSS e esta alteração seria aplicada a todo o sistema.

Os campos: curso, disciplina, turma, aluno, fone, celular e email FURB são de preenchimento obrigatório, sendo que, para garantir isso, o sistema informa através de mensagens de alertas o campo que faltou preencher e se não foi preenchido corretamente (no caso do campo email). Esse tratamento de exceção foi feito através de função javascript, exemplificado no Quadro 23.

```

...
#topo{
  width:80%;
  float:left;
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 21px;
  color: #FFFFFF;
  border-top-width: 1px;
  border-right-width: 1px;
  border-bottom-width: 1px;
  border-left-width: 1px;
  border-top-style: solid;
  border-right-style: solid;
  border-bottom-style: solid;
  border-left-style: solid;
  border-top-color: #9FB6CD;
  ...
  font-weight: bold;
  background-color: #9FB6CD;
  font-style: normal;}
#topodir{
  width:20%;
  float:left;
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 21px;
  color: #FFFFFF;
  border-top-width: 1px;
  border-right-width: 1px;
  border-bottom-width: 1px;
  border-left-width: 1px;
  border-top-style: solid;
  border-right-style: solid;
  border-bottom-style: solid;
  border-left-style: solid;
  border-top-color: #9FB6CD;
  border-right-color: #9FB6CD;
  border-bottom-color: #9FB6CD;
  border-left-color: #9FB6CD;
  font-weight: bold;
  background-color: #9FB6CD;
  font-style: normal;
}
#titulo {
  width:100%;
  float:left;
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 21px;
  color: #36428b;
  font-weight: bold;
  font-style: normal;
}
.botaoOk {
  width : 42px;
  height : 28px;
  font : bold 13px/15px calibri;
  text-decoration : none;
  text-align : center;
  background-position : left;
  background : url('../imagens/ok27.gif') no-repeat;
}...

```

Quadro 22 – Exemplo de utilização CSS

```

function valida_form(form,campos,nomescampos,tipos,status) {
  /*
  form = posição do formulário (0,1,...)
  campos = campos a verificar (0,1,2,...)
  tipos = tipo de cada campo:
          4-email
          5-data
          8-string
          9-login/senha
          10-confirmação de senha
  status = 0 - não obrigatório, 1 - obrigatório
  */
  var mensagem = "Os seguintes campos estão incorretos\n\n";
  var erro = false;
  for(var i=0;i<campos.length;i++) {
    resultado=true;
    valor = document.forms[form].elements[campos[i]].value;
    switch(tipos[i]) {
      case 4:
        resultado = valida_email(valor);
        break;
      case 5:
        resultado = valida_data(valor);
        break;
      case 8:
        if(status[i]<=1) {
          resultado = (valor.length==0) ? false : true;
        }
        else {
          resultado = (valor.length<status[i]) ? false : true;
        }
        break;
      case 9:
        resultado = valida_string(valor);
        break;
      case 10:
        resultado = valida_string(valor);
        if(resultado)
          resultado = (valor==document.forms[form].elements[campos[i-
1]].value);
        break;
    }
    if(!resultado && (status[i]>=1 || (status[i]==0 && valor.length!=0)))
    {
      mensagem+= "- " + nomescampos[i] + "\n";
      erro = true;
    }
  }
  if(erro) alert(mensagem)
  return !erro;}

```

Quadro 23 – Tratamento de exceção no preenchimento de campos

### 3.4 OPERACIONALIDADE DA IMPLEMENTAÇÃO

Nesta seção é abordada a operacionalidade da aplicação desenvolvida, representam a idéia central da aplicação, bem como, o problema inicial que motivou o seu desenvolvimento.

Neste primeiro momento é representado a tela de *login*, onde um usuário realiza a autenticação para ter acesso ao sistema. A administração do sistema é realizada por um usuário do tipo “Administrador” e que possui as permissões necessárias para a realização dos cadastros e configuração de permissões. A tela de *login*, apresentada na Figura 1, é comum a todos os usuários do sistema.

Figura 1 - Tela de *login*

Caso o nome ou senha estejam incorretos, o sistema informará em uma janela de mensagem, o erro de autenticação, mostrado na Figura 2.

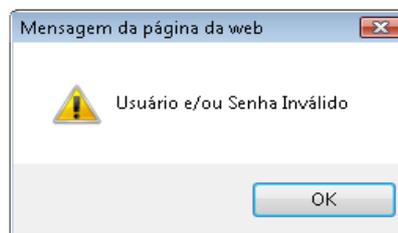


Figura 2 – Tela de erro de autenticação

Caso as credenciais informadas estejam corretas, o usuário acessará a página contendo um menu personalizado (de acordo com o nível de permissão). Na Figura 3 é apresentada a tela da página que o usuário *Administrador* tem acesso.

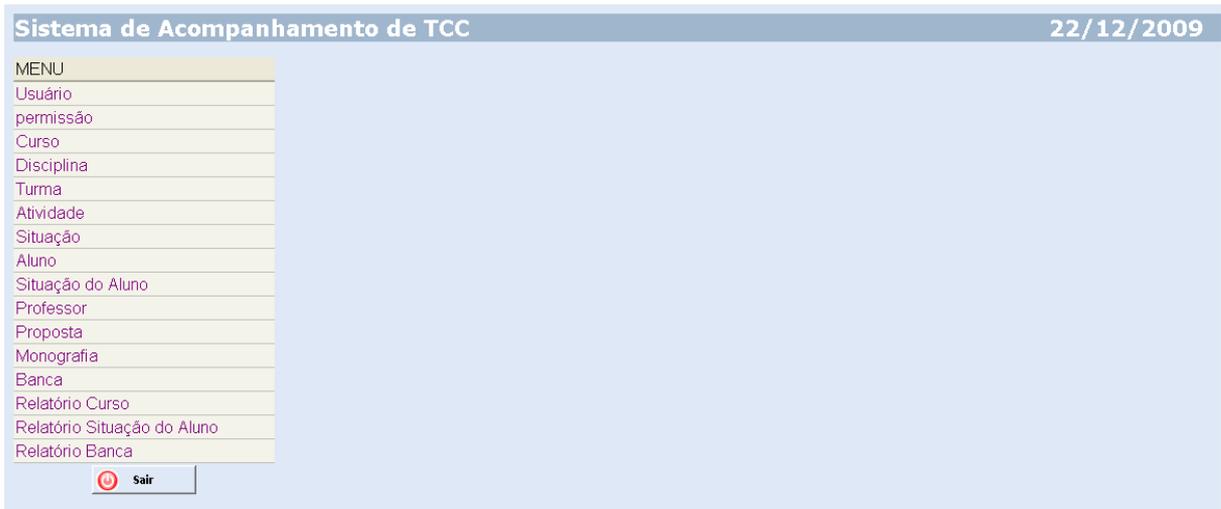


Figura 3 – Tela de menu

Para cadastrar um aluno o usuário escolhe no menu a opção “Aluno” que o levará para a página contendo o respectivo formulário de inclusão de aluno, onde deverão ser preenchidos os dados de acordo com as informações solicitadas. Conforme a Figura 4.

Figura 4 – Tela de formulário de inclusão de aluno

Após preencher os campos obrigatórios do formulário, o usuário pode confirmar sua inclusão, clicando no botão “confirmar”, quando é mostrada uma janela de alerta, “aluno incluído”. Na Figura 5 podemos observar a manutenção do aluno, na opção lista estão todos os dados dos alunos, o usuário poderá alterar ou excluir o aluno escolhido.

Id	Turma	Nome	Fone	Celular	EmailFurb	Opções
1	1	Elisangela Cristina Lombardi Klitzke	33742100	96248450	e@	<a href="#">Alterar</a> <a href="#">Excluir</a>
2	1	Marcelo Kroin	33742100	96248450	e@	<a href="#">Alterar</a> <a href="#">Excluir</a>
3	7	Brenda	33742100	96248450	e@	<a href="#">Alterar</a> <a href="#">Excluir</a>

Total de 3 Alunos

[Incluir um novo Aluno](#) [Voltar](#)

[Sair](#)

[administracao]

Figura 5 – Tela de manutenção de aluno

Na Figura 6 temos o cadastro da proposta onde deve ser informado o curso, a disciplina, a turma e o aluno, os nomes dos professores que farão parte da avaliação, o título e a versão da proposta.

**Inclusão de proposta**

Curso\*:

Disciplina\*:

Turma\*:

Aluno\*:

Avaliador1\*:

Avaliador2\*:

Avaliador3\*:

Proposta\*:

Versão\*:

[Confirmar](#) [Listar](#) [Voltar](#)

[Sair](#)

[administracao]

Figura 6 – Tela de cadastro da proposta

Na Figura 7 temos o cadastro da banca onde deve ser informado o título da monografia a sala a hora e a data da apresentação, os nomes dos professores que farão parte da avaliação e suas respectivas notas. O sistema calcula no final do cadastro de todas as notas as médias parciais e a média final do aluno.

Sistema de Acompanhamento de TCC 22/12/2009

MENU	<b>Inclusão de Banca</b>	
Usuário	Monografia*: <input type="text"/>	Media Monog.: <input type="text"/>
permissão	Data*: <input type="text"/>	Hora*: <input type="text"/> Sala*: <input type="text"/>
Curso	Avaliador 1: <input type="text"/>	Nota Def.1: <input type="text"/> Nota Implem.1: <input type="text"/>
Disciplina	Avaliador 2: <input type="text"/>	Nota Def.2: <input type="text"/> Nota Implem.2: <input type="text"/>
Turma	Avaliador 3: <input type="text"/>	Nota Def.3: <input type="text"/> Nota Implem.3: <input type="text"/>
Atividade	Nota Coord.: <input type="text"/>	Media Impl.: <input type="text"/> Media Final: <input type="text"/>
Situação	Media Def.: <input type="text"/>	
Aluno	<input type="button" value="Confirmar"/> <input type="button" value="Listar"/> <input type="button" value="Voltar"/> <input type="button" value="Calcular"/>	
Situação do Aluno		
Professor		
Proposta		
Monografia		
Banca		
Relatório Curso		
Relatório Situação do Aluno		
Relatório Banca		
<input type="button" value="Sair"/>		

[administracao]

Figura 7 – Tela de inclusão de banca

Na Figura 8 temos o relatório de situação do aluno, deve ser informada a atividade e a situação que se quer listar.

Sistema de Acompanhamento de TCC 25/01/2010

MENU	<b>Relatório de situação do aluno</b>	
Usuário	Atividade: <input type="text"/>	
permissão	Situacao: <input type="text"/>	
Curso	<input type="button" value="Confirmar"/> <input type="button" value="Listar"/> <input type="button" value="Voltar"/>	
Disciplina		
Turma		
Atividade		
Situação		
Aluno		
Situação do Aluno		
Professor		
Proposta		
Monografia		
Banca		
Relatório Curso		
Relatório Situação do Aluno		
Relatório Banca		
<input type="button" value="Sair"/>		

[administracao]

Figura 8 – Tela de relatório de situação do aluno

### 3.5 TESTES

Para cada ciclo foi elaborado, definido e identificado qual procedimento e quais tipos de testes serão realizados. Esse plano poderá ser alterado de acordo com a melhor definição dos requisitos do sistema. Ele também poderá ser utilizado durante todo o projeto, sendo

modificado a cada iteração.

No primeiro ciclo, foi marcado como o início da construção de um artefato chamado plano de teste. Essa construção foi direcionada pelos requisitos do sistema obtidos até o momento (usuário, permissão). Segundo mostra a figura 9.

<b>Contador:</b>	<b>001</b>
<b>Criticidade:</b>	Alta
<b>Localização:</b>	<a href="http://localhost/windows/">http://localhost/windows/</a>
<b>Objeto de Teste:</b>	Verificação de Usuário cadastrado
<b>Caso de Teste:</b>	Tentar colocar um login e senha não cadastrada
<b>Pré - Condição:</b>	1. Acesso ao sistema devidamente conectado.
<b>Procedimento:</b>	<ol style="list-style-type: none"> <li>1. Entrar na tela de login: <a href="http://localhost/windows/">http://localhost/windows/</a></li> <li>2. Digitar, no campo login, um login não cadastrado;</li> <li>3. Digitar, no campo senha, uma senha não cadastrada;</li> </ol>
<b>Resultado Esperado:</b>	1. O sistema deve informar um aviso Usuário e/ou Senha Invalida.

Figura 9 – Caso de teste, primeiro ciclo – usuário

No segundo ciclo, apresentou-se alterações no plano de teste devido ao número de requisitos já extraídos. Em conjunto com a fase de implementação, foi realizados testes de módulos, com objetivo de verificar o que está sendo feito. É importante lembrar que estes testes não irão validar um requisito, mas apenas verificar se ele foi implementado corretamente. Na figura 10 temos o caso de teste para manutenção dos dados da situação do aluno.

<b>Contador:</b>	<b>002</b>
<b>Criticidade:</b>	Alta
<b>Localização:</b>	<a href="http://localhost/windows/">http://localhost/windows/</a>
<b>Objeto de Teste:</b>	Operações referentes à manutenção dos dados
<b>Caso de Teste:</b>	Testar o funcionamento do botão “Confirmar”
<b>Pré - Condição:</b>	1. Acesso do usuário devidamente autenticado.
<b>Procedimento:</b>	<ol style="list-style-type: none"> <li>1. Entrar na tela: <a href="http://localhost/windows/man_alunoSituacao.php">http://localhost/windows/man_alunoSituacao.php</a></li> <li>2. Selecionar uma das opções do campo “Curso”;</li> <li>3. Selecionar uma das opções do campo “Disciplina”;</li> <li>4. Selecionar uma das opções do campo “Turma”;</li> <li>5. Selecionar uma das opções do campo “Aluno”;</li> <li>6. Selecionar uma das opções do campo “Atividade”;</li> <li>7. Selecionar uma das opções do campo “Situação”;</li> <li>8. Clicar no botão “Confirmar”.</li> </ol>
<b>Resultado Esperado:</b>	<ol style="list-style-type: none"> <li>1. O sistema atualizará no Banco de Dados os dados referentes à situação do aluno;</li> <li>2. O sistema exibirá a mesma tela em que estava;</li> <li>3. O sistema exibirá a mensagem padrão de operação realizada com sucesso.</li> </ol>

Figura 10 – Caso de teste, segundo ciclo – situação do aluno

No terceiro ciclo, consegue-se entender os pontos críticos de implementação e elaborar o plano de teste quase que totalmente. Os testes “caixa preta” são muito importantes nestas iterações, pois eles irão verificar a conformidade do sistema com as exigências do cliente. Testes de módulos continuam sendo feitos, conforme demonstra a figura 11 e neste instante

serão feitos também os testes de integração, onde os módulos são combinados e testados em grupo e resulta num sistema integrando e preparado para o teste de sistema.

<b>Contador:</b>	<b>003</b>
<b>Criticidade:</b>	Média
<b>Localização:</b>	Manutenção > Servidor > Pesquisa
<b>Objeto de Teste:</b>	Verificação de Máscaras
<b>Caso de Teste:</b>	Tentar colocar uma string no campo Data
<b>Pré - Condição:</b>	1. Acesso do usuário devidamente autenticado.
<b>Procedimento:</b>	1. Entrar na tela de cadastro da banca: <a href="http://localhost/windows/">http://localhost/windows/</a> 2. Selecionar uma das opções do campo "Monografia"; 3. Digitar, no campo data, uma string qualquer caracter diferente de números.
<b>Resultado Esperado:</b>	1. O sistema exibirá a mensagem padrão de data inválida.

Figura 11 – Caso de teste, terceiro ciclo – cadastro de banca

### 3.6 RESULTADOS E DISCUSSÃO

O desenvolvimento do sistema permitiu a utilização e aprimoramento de conhecimentos adquiridos na vida acadêmica, principalmente no que diz respeito ao desenvolvimento de sistemas orientado a objetos, que também contribuiu para a adoção do método UP, onde seguindo as quatro fases apresentadas por ele obteve-se o resultado desejado.

Na fase de concepção foram obtidas as primeiras informações com o coordenador do curso de TCC, que possui grande conhecimento de todo o processo, na qual foi elaborado o levantamento de requisitos, a organização dos requisitos e o planejamento do desenvolvimento.

Na fase de elaboração detalhou-se os casos de uso que na fase de concepção foram somente listados em alto nível. Criou-se um diagrama de seqüência para representar a seqüência dos eventos dos Casos de Uso. Analisando a visão que o usuário tinha das situações gerenciadas pelo sistema, foi construído um diagrama de classes. Também nesta fase foram definidos fatores como linguagem de programação, sistema operacional e banco de dados.

Na fase de construção foram feitas as implementações e testes, ou seja, foram automatizadas as classes correspondentes através das operações e consultas do sistema. A fase de transição não foi concluída, gerando assim uma extensão para trabalhos futuros.

Referente a segurança, conforme descrito na sessão 2.3, o PHP foi configurado para que

as variáveis superglobais fiquem desativadas e executar somente no modo de segurança. Para arquivos de include .inc, foram renomeados com a extensão .php. As senhas foram criptografadas usando a função md5(). O protocolo SSL não foi obtido nesta implementação.

Durante o período de implementação do sistema identificou-se, ainda, a incompatibilidade do mesmo com navegadores diferentes, como por exemplo, o *Internet Explorer* (versão 7.0.5730.13) e o *Mozilla Firefox* (versão 3.0.3). As incompatibilidades encontradas estão relacionadas à interface (alinhamento e posicionamento de campos e informações em tela). Além disso, foram encontradas diferenças também em relação à conexão do banco de dados nos sistemas operacionais Windows e Linux.

Por fim, pode-se observar que o sistema atende suas funcionalidades básicas. O que o torna interessante é justamente a principal característica das aplicações web, que vêm sendo utilizadas em larga escala nos sistemas mais recentes do mercado. O qual se deve o fato de que os sistemas podem ser acessados a qualquer hora, de qualquer computador e em qualquer lugar do mundo onde se tenha uma conexão com a internet, possibilitando que haja uma maior integração entre os usuários em geral.

## 4 CONCLUSÃO

A construção de *sites* dinâmicos tornou-se muito comum com o decorrer dos tempos, trazendo inúmeras vantagens para as empresas, porque expressões como iniciativa, agilidade, informação e segurança estão diretamente relacionadas ao sucesso das organizações. Porém, isso pode se converter em um problema caso o *site* ou a aplicação não seja desenvolvido com as devidas preocupações relacionadas à segurança da informação. Todas as linguagens de programação têm suas particularidades. Para contornar este problema, neste trabalho utilizam-se alguns comandos de segurança do PHP, que tem uma grande base de *sites* existentes atualmente, procurando combiná-los da melhor maneira possível.

Desde o princípio manteve-se a preocupação de seguir o método UP. Através dele pode-se validar a viabilidade do sistema, examinando os principais riscos no desenvolvimento, para tentar abordá-los o mais cedo possível. Assim, os casos de uso com maior riscos foram resolvidos primeiro, que foi o caso, por exemplo, do cadastro da banca.

Para permitir a apresentação do sistema no curto espaço disponível, várias simplificações foram consideradas, por exemplo, implementação em um ambiente real, para eventuais ajustes, validações como cadastro da banca e calendário de disponibilidade do professor, troca de mensagens entre os usuários (coordenador, professor e aluno).

Porém, as principais funcionalidades foram atendidas. O sistema disponibiliza informações para alunos e professores sobre o andamento dos TCCs, através de relatórios. Possibilita ao coordenador supervisionar todo o processo através das consultas e agiliza o tramite de informações às pessoas envolvidas.

O protótipo foi desenvolvido com tecnologias que acompanham a tendência do mercado, como por exemplo, o PHP5, que é uma ferramenta completa, estável, simples e agora com maior aderência ao modelo de orientação a objetos.

### 4.1 EXTENSÕES

Durante o desenvolvimento do trabalho foram verificadas diversas possíveis melhorias, muitas das quais não puderam ser contempladas neste trabalho. Portanto, as sugestões para extensão do trabalho são mostradas através de uma lista de tarefas. A lista de

tarefas é apresentada no Quadro 24, onde há a descrição de cada tarefa.

<b>TAREFA</b>
Implantar o sistema no usuário final, finalizar os testes e fazer treinamento.
Adaptações da estrutura do sistema para outros cursos.
Implementar uma consulta por palavras chaves nos arquivos da monografia.
Desenvolver relatórios melhorados, rotinas ou ferramentas de gerenciamento que permitam análises em relação ao cadastramento da banca a fim de aprimorar o processo e não deixar marcar uma banca, por exemplo, se o professor avaliador estiver indisponível nesta data.
Implementar a exibição dos dados estatísticos das bancas cadastradas.
Desenvolver recurso para possibilitar a criação do perfil professor e aluno, para que ambos pudessem interagir com o sistema.
Desenvolver recurso para a troca de E-mail entre os usuários.
Aperfeiçoar a utilização da aplicação pelos usuários, a partir da análise de usabilidade.
Obter protocolo SSL
Utilizar XML como interface e funções de criptografia no banco de dados, para aumentar a segurança.
Desenvolver recurso para gerar backup automático do banco de dados.
Desenvolver maior número de testes, que além de simular operações de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado, também incluir testes de configuração, de recuperação de falhas, de segurança e de desempenho.

Quadro 24 – Lista de tarefa

## REFERÊNCIAS BIBLIOGRÁFICAS

FALBO, Ricardo de Almeida. **A experiência na definição de um processo padrão baseado no processo unificado**. Vitória, 2000. Disponível em: <<http://www.inf.ufes.br/~falbo/download/pub/simpros2000.pdf>>. Acesso em: 01 set. 2009.

MARTINS, José Carlos Cordeiro. **Gerenciando projetos de desenvolvimento de software com PHP,RUP e UML**. Rio de Janeiro: Brasport, 2005.

MARTINS, Vidal. **O processo unificado de desenvolvimento de software**. Curitiba, 1999. Disponível em: <<http://www.batebyte.pr.gov.br/modules/conteudo/conteudo.php?conteudo=1227>>. Acesso em: 18 nov. 2009.

ROSA, Fabiano. **Protótipo de um diário de classe em dispositivos móveis utilizando J2ME**. 2005. 99 f. Trabalho de Conclusão de Curso, Universidade Regional de Blumenau, Blumenau.

SEABRA JÚNIOR, Rodolfo M. **Análise e projeto orientado a objetos usando UML e o processo unificado**. 2001. 113 f. Trabalho de Conclusão de Curso, Universidade Federal do Pará, Belém.

SEVERO, Rosane. **Uma Análise sobre a revolução da Informação**. [S.1.], [2006?]. Disponível em: <[http://www.consultores.com.br/artigos.asp?cod\\_artigo=72](http://www.consultores.com.br/artigos.asp?cod_artigo=72)>. Acesso em: 14 out. 2009.

SICA, C.; REAL Petter V. **Programação Segura Utilizando PHP**. Rio de Janeiro: Ciência Moderna, 2007.

SILVA, Elaine Martins da. **Cuidado com a engenharia social**. [S.1.], 2008. Disponível em: <<http://www.baixaki.com.br/info/1078-cuidado-com-a-engenharia-social.htm>> Acesso em: 01 Setembro 2009.

SOUZA, Alexandre Perin de; MOMBACH. Daniel D`Ávila. **Aplicação do processo unificado no desenvolvimento do sistema de informação para acompanhamento das informações profissionais dos egressos da UNIPLAC**. Lages, 2005. Disponível em: <<http://dcc.unesc.rct/sulcomp/Art0865SulComp2005.pdf>>. Acesso em: 26 abr. 2006.

UNIVERSIDADE REGIONAL DE BLUMENAU. Conselho de Ensino, Pesquisa e Extensão. **Resolução 13/2004**. Aprova o regulamento do trabalho de conclusão de Curso – TCC do curso de ciências da computação – bacharelado, na forma dos anexos I, II, III e IV. Blumenau: FURB, 2004. Disponível em: <[http://www.inf.furb.br/~roque/tcc/regulamento2004\\_tcc.html](http://www.inf.furb.br/~roque/tcc/regulamento2004_tcc.html)>. Acesso em: 01 set. 2009.

WAZLAWICK, Raul Sidnei. **Análise e projeto de sistemas de informação orientado a objetos**. São Paulo: Campus, 2004.

## ANEXO A – Avaliação da proposta

## 1.1 AVALIAÇÃO DO(A) ORIENTADOR(A)

Acadêmico(a): \_\_\_\_\_

Orientador(a): \_\_\_\_\_

ASPECTOS AVALIADOS		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO			
	1.1. O tema de pesquisa está devidamente contextualizado/delimitado?			
	1.2. O problema está claramente formulado?			
	2. OBJETIVOS			
	2.1. O objetivo geral está claramente definido e é passível de ser alcançado?			
	2.2. São apresentados objetivos específicos (opcionais) coerentes com o objetivo geral? Caso não sejam apresentados objetivos específicos, deixe esse item em branco.			
	3. RELEVÂNCIA			
	3.1. A proposta apresenta um grau de relevância em computação que justifique o desenvolvimento do TCC?			
	4. METODOLOGIA			
	4.1. Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
4.2. Os métodos e recursos estão devidamente descritos e são compatíveis com a metodologia proposta?				
4.3. A proposta apresenta um cronograma físico (período de realização das etapas) de maneira a permitir a execução do TCC no prazo disponível?				
5. REVISÃO BIBLIOGRÁFICA				
5.1. As informações apresentadas são suficientes e têm relação com o tema do TCC?				
5.2. São apresentados trabalhos correlatos, bem como comentadas as principais características dos mesmos?				
6. REQUISITOS DO SISTEMA A SER DESENVOLVIDO				
6.1. Os requisitos funcionais e não funcionais do sistema a ser desenvolvido foram claramente descritos?				
7. CONSIDERAÇÕES FINAIS				
7.1. As considerações finais relacionam os assuntos apresentados na revisão bibliográfica com a realização do TCC?				
ASPECTOS METODOLÓGICOS	8. REFERÊNCIAS BIBLIOGRÁFICAS			
	8.1. As referências bibliográficas obedecem às normas da ABNT?			
	8.2. As referências bibliográficas contemplam adequadamente os assuntos abordados na proposta (são usadas obras atualizadas e/ou as mais importantes da área)?			
	9. CITAÇÕES			
	9.1. As citações obedecem às normas da ABNT?			
9.2. As informações retiradas de outros autores estão devidamente citadas?				
10. AVALIAÇÃO GERAL (organização e apresentação gráfica, linguagem usada)				
10.1. O texto obedece ao formato estabelecido?				
10.2. A exposição do assunto é ordenada (as idéias estão bem encadeadas e a linguagem utilizada é clara)?				

A proposta de TCC deverá ser revisada, isto é, necessita de complementação, se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos **4 (quatro)** itens dos **ASPECTOS TÉCNICOS** tiverem resposta ATENDE PARCIALMENTE; ou
- pelo menos **4 (quatro)** itens dos **ASPECTOS METODOLÓGICOS** tiverem resposta ATENDE PARCIALMENTE.

**PARECER:** ( ) APROVADA ( ) NECESSITA DE COMPLEMENTAÇÃO

**ANEXO B – FICHA DE AVALIAÇÃO DA MONOGRAFIA (equivalente a 40% da nota final)**

Avaliador:

Acadêmico:

<p><b>ESCOLHA DO TEMA:</b> * relevante para o desenvolvimento da ciência e/ou tecnologia em Computação.</p> <p><b>DESENVOLVIMENTO LÓGICO:</b></p> <ul style="list-style-type: none"> <li>– <b>INTRODUÇÃO:</b> <ul style="list-style-type: none"> <li>a) contextualização do tema;</li> <li>b) formulação precisa do problema a ser tratado;</li> <li>c) elaboração de objetivos claros e coerentes com a proposta (ou com suas alterações ao longo do trabalho);</li> <li>d) justificativa para as proposições fundamentais.</li> </ul> </li> <li>– <b>FUNDAMENTAÇÃO TEÓRICA:</b> * objetiva e suficiente para o entendimento do trabalho.</li> <li>– <b>ESPECIFICAÇÃO/IMPLEMENTAÇÃO:</b> <ul style="list-style-type: none"> <li>a) requisitos do sistema/software claros e bem descritos;</li> <li>b) uso de uma metodologia, método ou técnica coerente com o problema proposto, tal que os modelos apresentados sejam consistentes, legíveis e essenciais para a implementação;</li> <li>c) descrição da implementação seguindo a especificação realizada.</li> </ul> </li> <li>– <b>BIBLIOGRAFIA:</b> * uso de bibliografia atualizada e consistente com o tema apresentado.</li> </ul> <p><b>REDAÇÃO:</b></p> <ul style="list-style-type: none"> <li>a) clara, precisa e objetiva;</li> <li>b) seqüência lógica e coerente;</li> <li>c) uso correto da língua portuguesa e de terminologia adequada.</li> </ul> <p><b>APRESENTAÇÃO:</b> * uso das normas técnicas adotadas pela FURB.</p>	
	<b>Nota (0,0 a 10,0)</b>
	<b>40%</b>

**Observação:** Este formulário deve ser entregue até 48 horas antes da apresentação pública para o coordenador de TCC. É considerado apto para apresentação pública o trabalho que obtiver nota igual ou superior a 6,0 (seis) de cada um dos membros da banca.

Data da avaliação: \_\_\_/\_\_\_/\_\_\_\_\_

Assinatura

do

avaliador:

Data do recebimento pelo coordenador do TCC: \_\_\_/\_\_\_/\_\_\_\_\_

Assinatura

do

coordenador:

## ANEXO C – FICHA DE AVALIAÇÃO DA APRESENTAÇÃO PÚBLICA

Avaliador: \_\_\_\_\_

Acadêmico: \_\_\_\_\_

<b>DEFESA (equivalente a 25% da nota final)</b>	
<ul style="list-style-type: none"> <li>a) domínio do tema;</li> <li>b) uso de linguagem técnico-científica clara e adequada;</li> <li>c) exposição do assunto seguindo uma seqüência lógica;</li> <li>d) bom aproveitamento do tempo;</li> <li>e) ênfase nos aspectos relevantes do trabalho;</li> <li>f) habilidade de comunicação;</li> <li>g) habilidade no uso dos recursos;</li> <li>h) clareza nas respostas às perguntas formuladas pela banca examinadora e platéia.</li> </ul>	
<b>Nota (0,0 a 10,0)</b>	
<b>25%</b>	
<b>IMPLEMENTAÇÃO (equivalente a 30% da nota final)</b>	
<ul style="list-style-type: none"> <li>a) domínio do ambiente utilizado para demonstração;</li> <li>b) apresentação de casos que testam as principais funcionalidades do sistema/software;</li> <li>c) o sistema/software apresentado possui poucos erros;</li> <li>d) o sistema/software apresentado corresponde ao especificado/descrito na monografia.</li> </ul>	
<b>Nota (0,0 a 10,0)</b>	
<b>30%</b>	

**Observação:** Este formulário deve ser entregue ao presidente da banca examinadora logo após o término dos questionamentos para cálculo da média parcial do TCC.

Data da avaliação: \_\_\_/\_\_\_/\_\_\_\_\_

Assinatura do avaliador: \_\_\_\_\_

## ANEXO D – FICHA DE AVALIAÇÃO GERAL DO TCC

Acadêmico: \_\_\_\_\_

<b>AVALIADOR</b>	<b>Avaliador 1 (orientador)</b>	<b>Avaliador 2</b>	<b>Avaliador 3</b>	<b>Coordenação</b>	<b>Média Aritmética</b>
<b>TIPO DE AVALIAÇÃO</b>					
<b>Monografia (40%)</b> (anterior à apresentação pública)					
<b>Defesa (25%)</b> (após à apresentação pública)					
<b>Implementação (30%)</b> (após à apresentação pública)					
<b>Média parcial</b> (soma das <b>médias aritméticas</b> da monografia, defesa e implementação)					
<b>Coordenação (5%)</b> (após entrega e aceitação do material final pedido pela coordenação)					
<b>Nota final</b> (soma da <b>média parcial</b> e da <b>coordenação</b> )					

**Observações:**

- este formulário deve ser preenchido pelo presidente e conferido pelos demais membros da banca examinadora até a **média parcial**, a qual deve ser transportada para a ata. A média parcial **NÃO** deve ser informada ao acadêmico ao final dos trabalhos da apresentação pública;
- **não** devem ser informadas as notas individuais dos membros avaliadores ao acadêmico;
- a monografia, o CD e a ficha de autorização para publicação eletrônica devem ser entregues ao coordenador e ter aceitação deste, para que a nota final possa ser calculada e validada. Caso isto não aconteça, a nota final é 0,0 (zero);
- o acadêmico é considerado aprovado se a **nota final** for igual ou superior a 6,0 (seis).

Data da avaliação da apresentação pública: \_\_\_/\_\_\_/\_\_\_\_\_

Assinatura do avaliador 1 (presidente da banca): \_\_\_\_\_

Assinatura do avaliador 2: \_\_\_\_\_

Assinatura do avaliador 3: \_\_\_\_\_

Data da avaliação do coordenador: \_\_\_/\_\_\_/\_\_\_\_\_

Assinatura do coordenador: \_\_\_\_\_