

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

GUSTAVO SEGANFREDO FARIAS

**Reengenharia do Sistema de  
Administração do PPGC**

Trabalho de Graduação

Prof. Dr. Marcelo Soares Pimenta  
Orientador

Porto Alegre, dezembro de 2013

# SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS.....</b>	<b>4</b>
<b>LISTA DE FIGURAS.....</b>	<b>5</b>
<b>ABSTRACT.....</b>	<b>7</b>
<b>1 INTRODUÇÃO.....</b>	<b>8</b>
1.1 Motivação.....	8
1.2 Objetivos.....	8
1.3 Estrutura do Trabalho.....	9
<b>2 PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO.....</b>	<b>10</b>
2.1 A informatização do PPGC.....	10
2.2 Operação antes da informatização.....	10
2.3 Operação após a informatização.....	11
2.4 Novos requisitos / Necessidades.....	12
<b>3 O SISTEMA DE INSCRIÇÕES DO PPGC.....</b>	<b>13</b>
3.1 Os requisitos iniciais.....	13
3.2 O funcionamento do sistema.....	13
3.2.1 A Interface.....	14
3.2.2 O Servidor.....	16
3.2.3 Atualizações.....	18
3.3 A evolução do sistema de inscrição.....	18
3.3.1 Uma primeira grande refatoração.....	18
3.3.1.1 Projeto de entry points.....	19
3.3.1.2 Internacionalização do sistema.....	19
3.4 Os problemas apresentados.....	19
3.4.1 Dependência do JavaScript.....	19
3.4.2 Interface defasada.....	20
3.4.3 Programação procedural.....	20
3.4.4 Falta de testes automatizados.....	20
<b>4 FUNDAMENTOS E CONCEITOS BÁSICOS.....</b>	<b>21</b>
4.1 Modelagem de Processos de Negócios.....	21
4.2 Estórias de usuário.....	21
4.3 Testes de aceitação e Behaviour-Driven Development (BDD).....	22
4.4 Arquitetura Model-View-Controller (MVC).....	22
4.5 Sistema de versionamento de código distribuído.....	23
4.6 Versionamento do banco de dados.....	24
4.7 Responsive Web Design & Mobile First.....	25
4.8 Ferramentas de suporte.....	25
4.8.1 Framework FuelPHP.....	25
4.8.2 Behat & Mink para a automatização de testes de aceitação e BDD.....	25

4.8.3 Git, Bitbucket e GitFlow.....	26
<b>5 O SISTEMA DE ADMINISTRAÇÃO DO PPGC.....</b>	<b>28</b>
5.1 Os requisitos do sistema.....	28
5.2 O processo de desenvolvimento.....	29
5.2.1 Projeto e elicitação de requisitos.....	29
5.2.2 Modelagem do processo.....	29
5.2.3 Extração das estórias de usuário.....	29
5.2.4 Criação de cenários para os testes automatizados.....	29
5.2.5 Desenvolvimento da funcionalidade.....	31
5.2.6 Validação.....	32
5.3 Novas funcionalidades.....	32
5.3.1 Seleção de Alunos de Doutorado.....	32
5.3.1.1 Descrição passo-a-passo do processo.....	32
5.3.1.2 Modelagem do processo.....	33
5.3.1.3 Extração dos casos de uso e estórias de usuário.....	34
5.3.1.4 Demonstração passo-a-passo.....	35
5.3.2 Submissão do Plano de Ensino e Pesquisa (PEP).....	46
5.3.2.1 Descrição passo-a-passo do processo.....	47
5.3.2.2 Modelagem do processo.....	47
5.3.2.3 Extração de casos de uso e estórias de usuário.....	49
5.3.2.4 Demonstração passo-a-passo.....	49
<b>6 CONCLUSÃO.....</b>	<b>54</b>
<b>7 REFERÊNCIAS.....</b>	<b>55</b>

## LISTA DE ABREVIATURAS E SIGLAS

<b>AJAX</b>	Asynchronous JavaScript and XML
<b>BDD</b>	Behaviour-Driven Development
<b>CSS</b>	Cascading Style Sheets
<b>DSL</b>	Domain Specific Language
<b>DVCS</b>	Distributed Version Control System
<b>FTP</b>	File Transfer Protocol
<b>HTML</b>	HyperText Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>JSON</b>	JavaScript Object Notation
<b>MySQL</b>	My Structured Query Language
<b>MVC</b>	Model-View-Controller
<b>ORM</b>	Object-Relational Mapping
<b>PDO</b>	PHP Data Objects
<b>PEP</b>	Plano de Ensino e Pesquisa
<b>PHP</b>	PHP: Hypertext Preprocessor
<b>PIN</b>	Personal Identification Number
<b>POSCOMP</b>	Exame Nacional para Ingresso na Pós-Graduação em Computação
<b>POSGRAD</b>	Sistema de gerenciamento de cursos de pós-graduação da UFRGS
<b>PPGC</b>	Programa de Pós-Graduação em Computação
<b>XML</b>	Extensible Markup Language

## LISTA DE FIGURAS

Figura 3.1: Modelo de aplicação web utilizando AJAX.....	14
Figura 3.2: Função simplificada utilizada para efetuar requisições assíncronas no sistema.....	15
Figura 3.3: Página inicial o do sistema de inscrições.....	16
Figura 3.4: Exemplo de organização do código do servidor no sistema de inscrições.....	17
Figura 4.1: Estória de usuário representada na forma de um cartão.....	22
Figura 4.2: Diagrama de controle de versionamento distribuído.....	24
Figura 4.3: Representação gráfica do GitFlow.....	27
Figura 5.1: Exemplo de teste de aceitação automatizado.....	30
Figura 5.2: Exemplo de implementação de um passo.....	31
Figura 5.3: Modelagem do processo de seleção de candidatos ao doutorado.....	34
Figura 5.4: Interface para a listagem de pagamentos pendentes.....	36
Figura 5.5: Confirmação de pagamento.....	36
Figura 5.6: Listagem de candidatos ao mestrado para aprovação.....	37
Figura 5.7: Visualização do cadastro do candidato.....	38
Figura 5.8: Recusa de candidato, com justificativa.....	39
Figura 5.9: Candidato confirmado pelo orientador.....	39
Figura 5.10: Panorama do processo seletivo do doutorado.....	40
Figura 5.11: Cadastro de bancas.....	41
Figura 5.12: Confirmação de agendamento da banca.....	42
Figura 5.13: Listagem de pendências para reserva de salas.....	43
Figura 5.14: Marcação de sala para entrevista de doutorado.....	43
Figura 5.15: Ficha de avaliação de candidato de doutorado.....	44
Figura 5.16: Ficha de avaliação submetida.....	45
Figura 5.17: Visualização da ficha da entrevista de candidato ao doutorado.....	46
Figura 5.18: Modelagem do processo de submissão do plano de ensino e pesquisa.....	48
Figura 5.19: Interface para a submissão do plano de ensino e pesquisa.....	50
Figura 5.20: Formulário para a submissão do PEP.....	50
Figura 5.21: Visualização pelo professor dos PEPs enviados pelos seus orientandos.....	51
Figura 5.22: Formulário para a reprovação de PEP.....	51
Figura 5.23: PEP aprovado.....	52
Figura 5.24: Visualização dos PEPs submetidos e pendentes pela secretaria.....	53

## **RESUMO**

Este trabalho descreve a concepção e desenvolvimento de um sistema de administração para o Programa de Pós-Graduação em Computação através da reengenharia do atual sistema, denominado “Sistema de Inscrições do PPGC”.

Apresentaremos a situação atual do PPGC, incluindo parte do desenvolvimento e funcionamento do Sistema de Inscrições, suas peculiaridades e limitações. A partir do conhecimento operacional, utilizaremos um conjunto de técnicas e ferramentas para o desenvolvimento do sistema de administração, visando o futuro da informatização de processos executados pelo Programa, incluindo a interação entre as atividades desenvolvidas por membros da coordenação, secretaria, docentes e discentes.

## **ABSTRACT**

This paper describes the design and development of an administration system for the Graduate Program in Computer Science through the reengineering of the current system, known as “Sistema de Inscrições do PPGC”.

We will present the current situation of PPGC, including part of the development and operation of the application system, its peculiarities and limitations. From the operational knowledge, we will use a set of techniques and tools for the development of the administration system, in view to the future of computerization of processes executed by the Program, including the interaction between the activities undertaken by members of the coordination, secretariat, faculty and students.

# 1 INTRODUÇÃO

A administração do Programa de Pós-Graduação em Computação é uma tarefa complexa, operada por um conjunto de professores (em cargos de coordenação) e técnicos administrativos, interagindo com uma comunidade de cerca de 400 membros, entre discentes e docentes do Programa. A cada ano, cerca de 80 novos alunos de mestrado e 30 de doutorado passam a integrar esta contagem.

Hoje, poucos processos executados internamente no PPGC são automatizados, ou possuem uma ferramenta para auxiliar a comunicação e a organização de informações tramitantes entre os atores que os compõe. Se faz necessário portanto o desenvolvimento de uma ferramenta capaz de integrar - iterativamente - estes processos, centralizando e facilitando o acesso a estes dados.

Na sequência, serão apresentadas a motivação, os objetivos e a estrutura deste trabalho.

## 1.1 Motivação

Este trabalho tem como principal motivação a necessidade do desenvolvimento de um sistema robusto para acomodar as atuais e futuras informatizações de processos do PPGC, como parte da iniciativa do Instituto de Informática de aplicar - dentro de sua estrutura administrativa - os conceitos e práticas desenvolvidos por seus docentes em aula ou projetos de pesquisa.

## 1.2 Objetivos

O objetivo deste trabalho é descrever a concepção e o desenvolvimento de um sistema de administração para o Programa de Pós-Graduação em Computação visando o futuro da informatização deste setor. O sistema representa uma reengenharia do sistema de inscrições atual, adaptando-se aos novos requisitos que surgiram após a sua implantação, separando o conceito de “inscrição de candidatos” e “administração de recursos do Programa”.



### **1.3 Estrutura do Trabalho**

Este trabalho está estruturado da seguinte forma: o capítulo 2 apresenta uma breve introdução do Programa de Pós-Graduação em Computação e apresenta a situação atual da sua informatização. O capítulo 3 comenta sobre o sistema de inscrições, apresentando os seus requisitos iniciais, suas funcionalidades básicas e estruturação. Aqui são apresentados alguns dos problemas que visam ser resolvidos neste trabalho. O capítulo 4 apresenta os conceitos fundamentais aplicados no desenvolvimento deste trabalho. Ele também apresenta as ferramentas utilizadas para o desenvolvimento do sistema de administração do PPGC. O capítulo 5 apresenta a proposta e o desenvolvimento do sistema, com alguns exemplos de funcionalidades operacionais.

## **2 PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO**

O Programa de Pós-Graduação em Computação é um dos mais antigos programas brasileiros de pós-graduação na área. Com atuação em diversas sub-áreas da Computação, consolidou-se, ao longo de mais de três décadas, como centro de excelência em ensino e pesquisa.

O objetivo do programa é a formação de pesquisadores e profissionais qualificados para desenvolver atividades em empresas de alta tecnologia através dos programas de Mestrado em Ciência da Computação (fundado em 1973) e Doutorado em Ciência da Computação (fundado em 1989).

Os programas são fortemente integrados com as atividades de pesquisa, agrupadas em cinco áreas de concentração: Engenharia da Computação, Sistemas de Computação, Sistemas de Informação, Inteligência Artificial e Informática Teórica. Os resultados destas pesquisas se traduzem na publicação de artigos científicos em veículos de grande impacto e no desenvolvimento de produtos e processos de empresas geradoras de tecnologia.

Atualmente, o Programa recebe cerca de 80 novos alunos de mestrado e 30 novos alunos de doutorado por ano.

### **2.1 A informatização do PPGC**

O PPGC vem passando por um processo de informatização das suas atividades, no que diz respeito ao relacionamento entre o Programa e os seus alunos e professores associados. A cerca de 3 anos foi desenvolvido um sistema para informatizar o processo de inscrição e seleção de alunos de mestrado, sendo este posteriormente estendido a seleção de alunos de doutorado e alunos especiais. O mesmo sistema hoje é utilizado para o acompanhamento da produção dos professores associados, entre outras atividades a serem relatadas no decorrer deste trabalho.

### **2.2 Operação antes da informatização**

O processo de inscrição para o mestrado e doutorado era anteriormente feito exclusivamente através de formulários de papel, sendo necessário preencher um requerimento e formulário de inscrição. Adicionalmente, os alunos de mestrado ainda deveriam preencher um segundo formulário com as suas opções de linhas de pesquisa e orientadores. Uma vez preenchidos os formulários, o aluno ainda deveria anexar uma série de cópias de documentos conforme previsto em edital, e entregar na secretaria do Programa para a análise dos orientadores e das linhas de pesquisa, junto com o comprovante de pagamento da taxa de inscrição.

Adicionalmente, os alunos deveriam apresentar cartas de recomendação, a qual o Programa propunha um modelo a ser preenchido pelo recomendador, que posteriormente enviava o documento por correio ou correio eletrônico.

O processo seletivo do mestrado consistia na replicação da documentação de um candidato para todos os orientadores indicados por ele no formulário de inscrição. Estes professores analisariam a documentação, e junto com as suas respectivas linhas de pesquisa efetuariam a seleção dos candidatos que melhor se encaixassem conforme a série de requisitos impostos pelo Programa. Através de uma série de reuniões, se chegaria na listagem oficial dos alunos selecionados, que era repassado a secretaria.

Uma vez selecionados, os novos alunos eram informados por e-mail e os nomes divulgados no site do Programa.

### **2.3 Operação após a informatização**

O processo de inscrição e seleção foi completamente informatizado. Novos candidatos são redirecionados para o Sistema de Inscrições, também conhecido como "applying". O sistema gera um cadastro único para cada candidato - protegido por senha - que pode ser preenchido a qualquer momento.

Uma vez iniciada a inscrição, o candidato preenche digitalmente um formulário e anexa a documentação digitalizada. Ele também pode solicitar cartas de recomendação, que são geradas pelo sistema e preenchidas pelo recomendador através do sistema online. Uma vez que os dados estejam preenchidos, o aluno submete a inscrição. O boleto para pagamento da taxa de inscrição é gerado automaticamente pelo sistema, e assim que o pagamento é recebido, o registro do candidato é efetivado.

Ao final do período de inscrições, os professores tem acesso - através de uma interface web - a toda a documentação de todos os candidatos. Adicionalmente, foi possível incluir artefatos de interesse como ranqueamentos de alunos através da nota do POSCOMP, filtragem de candidatos por linha de pesquisa, por opções de orientação, e afins.

Com o processo de inscrição finalizado, inicia-se a fase de seleção de candidatos. As seleções efetuadas no sistema são visíveis por todos os professores do Programa, o que visa evitar que o mesmo candidato seja selecionado por diferentes pessoas. Em caso de conflito entre a seleção de dois orientadores, o sistema trata automaticamente através de um algoritmo de prioridades.

Terminado o período de seleções, o sistema automaticamente envia a saudação para os novos alunos do Programa, com instruções para a matrícula. A secretaria, de posse da listagem de alunos aceitos efetua o registro destes no sistema da Universidade (POSGRAD).

A seleção de alunos de doutorado é relativamente mais simples. O professor indicado pelo candidato analisa a documentação através da interface web do sistema, e confirma ou rejeita

a inscrição. No caso da inscrição ser aceita, a comissão de seleção do doutorado se encarrega de montar as bancas para a entrevista deste candidato. Através do sistema é registrada a banca, data, hora e local da entrevista.

Durante a entrevista, o relator da banca tem acesso a documentação completa do candidato, bem como a ficha da entrevista, a qual ele deverá preencher com o parecer final da banca (aceitação ou recusa da inscrição do candidato). De posse do parecer, a comissão de seleção repassa uma lista final de candidatos aceitos à secretaria, que os registra no POSGRAD.

Adicionalmente, o sistema ainda comporta a inscrição de alunos especiais, que se candidatam a cursar disciplinas individuais. Após efetuado o cadastro, os professores que ministram disciplinas do PPGC recebem a listagem de alunos especiais aguardando por vagas nas respectivas disciplinas, inscrição esta que o professor pode aceitar ou rejeitar, conforme critérios próprios e a existência de vagas.

## **2.4 Novos requisitos / Necessidades**

Com o passar do tempo, o Programa começou a vislumbrar a informatização de outros processos, como por exemplo a análise da produção acadêmica dos professores do Programa. Com isso, o sistema que antes fora projetado especificamente para a inscrição/seleção de candidatos passou a receber funcionalidades relativas a outros processos. Estas funcionalidades, bem como a sua implementação será tratada na próxima seção deste trabalho.

### **3 O SISTEMA DE INSCRIÇÕES DO PPGC**

O Programa de Pós-Graduação em Computação já conta hoje com um sistema para a automatização do processo de inscrição, seleção e avaliação de docentes. Este capítulo apresentará os requisitos iniciais deste sistema e parte do seu processo de desenvolvimento. Ao final do capítulo, apresentaremos as dificuldades encontradas para a manutenção deste sistema em vistas ao futuro da informatização do Programa.

#### **3.1 Os requisitos iniciais**

O sistema atual foi projetado para ser um formulário de inscrição específico para o mestrado, tendo os seguintes requisitos iniciais:

- O candidato deve poder editar o seu cadastro a qualquer momento, antes de efetivamente submeter a sua inscrição;
- O formulário de inscrição deve conter todos os campos do formulário em papel, além de informações que a secretaria considerar pertinente;
- A recomendação de candidatos deve ser feita de maneira digital pelo recomendador, e disponibilizada no sistema;
- Os professores devem poder acessar o cadastro dos candidatos para a análise da documentação; e
- O sistema deve rodar sobre a infraestrutura já disponível no Instituto de Informática.

#### **3.2 O funcionamento do sistema**

O sistema utiliza um mecanismo de autenticação baseado na dupla PIN-senha, onde o candidato indica o seu email e recebe suas credenciais de acesso através de uma mensagem do sistema. O formulário para a inscrição em papel foi replicado digitalmente, incluindo campos solicitados pela secretaria do PPGC para facilitar o registro posterior dos candidatos aceitos junto ao sistema da Universidade (POSGRAD).

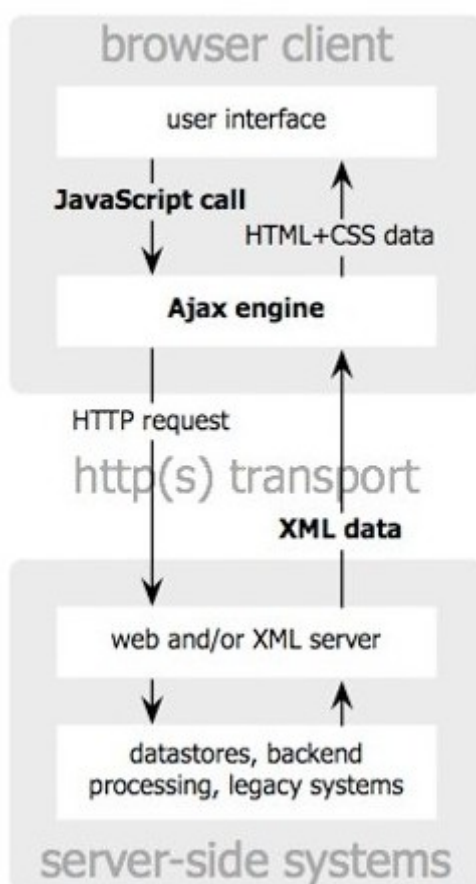
A carta de recomendação foi informatizada. O candidato indica o endereço de e-mail e o nome do recomendador e este recebe a solicitação através de e-mail disparado pelo sistema. Ao receber a mensagem, o recomendador acessa o endereço da formulário de recomendação utilizando um código único e preenche uma série de informações consideradas de interesse do Programa, com um campo para informações adicionais que o recomendador considere pertinente. Após o preenchimento e a submissão da recomendação, o recomendado recebe a

informação de que a carta foi submetida. O conteúdo da recomendação é visível apenas pelos professores do PPGC com acesso ao sistema.

O sistema é acessível através de uma interface web, desenvolvida utilizando a linguagem de programação PHP (versão 5.2) para o lado do servidor, e banco de dados MySQL (versão 5+) como persistência de dados.

### 3.2.1 A Interface

A interface do site é fortemente voltada a tecnologias do lado do cliente, com ênfase no uso de AJAX para o carregamento de "seções" do site. A ideia foi evitar processamento desnecessário por parte do servidor, carregando apenas o que é necessário para a exibição da recurso solicitado.



*Figura 3.1: Modelo de aplicação web utilizando AJAX*

Neste sistema, o AJAX foi implantado interceptando o evento "clique do mouse" sobre os itens do menu e sobrescrevendo a operação padrão do navegador (efetuar uma chamada HTTP

para o link, recarregando a página) utilizando uma função predeterminada, escrita em JavaScript. Esta função efetua a requisição assincronamente, e quando a resposta é recebida, preenche a área destinada ao conteúdo com os dados processados. Esta operação pode ser visualizada na figura 3.1.

Assim, o servidor "serve" a interface apenas uma vez. Todas as requisições posteriores são feitas apenas para conteúdo ou atualização, e todas são feitas através de funções pré-programadas em JavaScript.

```

1  var requestContent = function(content_url) {
2      var loader = $('loader'),
3          content_container = $('content');
4
5      loader.show();
6      content_container.setStyle('opacity', .6);
7      new Request.HTML({
8          url: content_url,
9          update: content_container,
10         onComplete: function(){
11             loader.hide();
12             content_container.setStyle('opacity', 1);
13         }
14     }).post();
15 }

```

*Figura 3.2: Função simplificada utilizada para efetuar requisições assíncronas no sistema*

As linhas 5 e 6 da figura 3.2. preparam o código para simular o efeito "carregando" no site, incluindo uma animação de carregamento (loader) e esmaecendo a cor do conteúdo atual da página.

A requisição é preparada entre as linhas 7 e 14, informando ao objeto *Request.HTML* o recurso a ser buscado (campo *url*), o local onde o conteúdo recuperado deve ser colocado (campo *update*) e uma função a ser executada depois que a requisição for completada (campo *onComplete*). Esta função remove os artefatos incluídos para simular o efeito "carregando".

O design da interface (figura 3.3) segue os padrões de identificação visual do Instituto de Informática, utilizando as cores vermelho, preto e branco. O logotipo da Universidade também foi utilizado para dar um caráter institucional ao sistema.

**INSCRIÇÃO** ✕

## Inscrições

Bem-vindo ao sistema de inscrição ao Programa de Pós-Graduação em Computação.

Caso esta seja a sua primeira vez neste sistema, entre com o seu e-mail no formulário abaixo. O campo "Curso" reflete o programa que você optará por se inscrever. Lembrando que as inscrições para mestrado e alunos especiais devem obedecer as datas definidas no calendário do Programa de Pós-Graduação, e a opção fica acessível somente durante o período de inscrição. O sistema gera um PIN (Personal Identification Number) e uma senha, que são enviados à respectiva caixa de correio. Com estes dados em mãos, você pode acessar o sistema e preencher a sua inscrição pelo formulário abaixo (a direita).

**Ainda não estou registrado - Registrar**

E-Mail

Curso

Código de Verificação

**Já estou registrado - Login**

PIN

Senha

[Esqueci minha senha](#)

Universidade Federal do Rio Grande do Sul - Instituto de Informática - Programa de Pós-Graduação em Computação  
 PROPG · PROPESQ · CNPq · CAPES · FAPERGS · UFRGS

Figura 3.3: Página inicial o do sistema de inscrições

### 3.2.2 O Servidor

A implementação do servidor foi feita utilizando uma arquitetura monolítica. Ao receber uma requisição de recurso por parte do cliente, o servidor executa o arquivo php responsável por tratar esta requisição.



```

1  <?php
2
3  include BASE_PATH.'/session.php';
4  include BASE_PATH.'/db.php';
5
6  $sql = "SELECT * FROM ...";
7  $data = $dbh->query($sql);
8
9  echo "<table>";
10 foreach ($data as $line) {
11     echo "<tr>";
12     echo "<td>".$data['nome']."</td>";
13     echo "<td>".$data['email']."</td>";
14     echo "</tr>";
15 }
16 echo "</table>";
17

```

Figura 3.4: Exemplo de organização do código do servidor no sistema de inscrições

Observa-se na figura 3.4 que para cada recurso o cabeçalho representa a inclusão de um arquivo com configurações comuns (*session.php*, linha 3) e eventuais requisições do arquivo que criava a conexão com o banco de dados (*db.php*, linha 4). Ambos arquivos efetuam uma série de processamentos, incluindo constantes de configuração e variáveis no escopo global da aplicação. O recurso pode portanto utilizar estas configurações e variáveis durante a sua execução.

A exemplo, o arquivo *db.php* contém as configurações e o processo de conexão com o banco de dados, gerando uma instância da classe PDO na variável "*\$dbh*", que é registrada no escopo global da aplicação. O recurso pode se valer desta variável para efetuar transações com o banco de dados.

O conteúdo do arquivo do recurso consiste no processamento e na geração do retorno da requisição. Um exemplo de como é feito o processamento pode ser visto entre as linhas 6 e 16. Nesta arquitetura, o processamento e o retorno encontram-se emaranhados, não ocorrendo separação entre os dados e a visualização.

Toda a implementação do sistema é feita através de recursos considerados "*front*" - os quais produzem respostas em HTML - e "*back*" - os quais produzem respostas em JSON.

Recursos "*front*" normalmente consistem na apresentação de conteúdo, como por exemplo, exibir a listagem de candidatos ao mestrado inscritos. A resposta destes recursos é em HTML puro, apresentado diretamente ao cliente. Recursos "*back*" são utilizados para atualizar, remover ou incluir registros no servidor, como por exemplo efetuar a seleção de um candidato ao

mestrado. A resposta destes recursos é em JSON, um objeto a ser processado em client-side, gerando uma atualização pontual na interface (alterando o status do candidato de "Aguarda Seleção" para "Selecionado", por exemplo).

### **3.2.3 Atualizações**

Atualizações de funcionalidades eram implantadas no servidor (produção) através de FTP. Os arquivos criados/alterados eram enviados manualmente, um-a-um para o seu local de destino. O registro dos arquivos alterados deveria ser feito manualmente em um documento para o seu posterior envio.

## **3.3 A evolução do sistema de inscrição**

Ao longo do tempo, novas funcionalidades foram integradas ao sistema. Neste momento, o sistema consistia nos atores "professor", "secretaria", "diretoria" e "candidato ao mestrado", todos ocupados com a tarefa primária do sistema, que era reproduzir o processo de inscrição e seleção ao mestrado.

A primeira funcionalidade adicionada em produção foi a inclusão do processo de inscrição ao doutorado, que possuía grandes similaridades ao de mestrado. Posteriormente, o processo de inscrição e seleção de alunos especiais também foi incluído no sistema, reaproveitando os formulários já projetados para mestrado e doutorado.

Mais adiante, foram adicionadas funcionalidades que já fugiam do escopo original do sistema de inscrições. A primeira delas foi relativa a avaliação da produção de professores, visando o processo de credenciamento destes junto ao PPGC. O processo consiste na avaliação da produção acadêmica de cada docente, com base em dados previamente preenchidos junto a Plataforma Lattes, extraídos em um arquivo XML e enviados ao sistema de inscrições. O sistema então efetua um processamento do arquivo e extrai a produção acadêmica em periódicos, conferências e livros, registrando-as no banco de dados. Com estes dados em mãos, é possível conferir se o professor possui os requisitos mínimos para a renovação do vínculo, descritos na resolução 01/2010. Novas funcionalidades surgiram aproveitando a presença destes dados, como o ranqueamento de orientadores, listagem da produção acadêmica para disposição no site do Programa, entre outras.

O sistema evoluiu para agregar funcionalidades que incluíam um novo ator: os alunos regulares do Programa. Alunos de doutorado deveriam poder incluir o currículo lattes para análise do sistema, gerando o ranqueamento entre eles visando distribuição de bolsas. Alunos de mestrado deveriam poder incluir solicitação de aproveitamento de disciplinas, além de submeter o "plano de curso" no começo do seu primeiro semestre.

O sistema inscrições também foi traduzido para a língua inglesa. A tradução foi feita duplicando o sistema original e traduzindo o texto em português.

### **3.3.1 Uma primeira grande refatoração**

Com o aumento das responsabilidades do sistema, o tempo despendido para a manutenção e inclusão de novas funcionalidades passou a ser um problema constante. Foi verificado que o sistema necessitaria passar por uma grande refatoração, alterando a estrutura original do código, limpando o escopo global e incluindo elementos internos de controle (captura de erros e exceções para a geração de logs).

### **3.3.1.1 Projeto de entry points**

O primeiro grande ato foi a remoção de múltiplos pontos de entrada do sistema, isto é, evitar que requisições sejam feitas diretamente a um recurso. O novo design força que requisições sejam feitas a um arquivo de entrada que prepara a execução do sistema, incluindo a configuração inicial, a configuração do sistema de captura e gerenciamento de erros e o "autoloader" - função responsável pelo carregamento de classes ou funções utilizadas pelo recurso mas não encontradas pelo PHP no escopo dinâmico durante a execução.

Tendo um entry point, acaba a necessidade da inclusão de um cabeçalho recursos (explicado na seção 3.2.2), bem como se torna possível centralizar o controle de execução e permissão de acesso aos recursos. O entry point vira o responsável por orquestrar o sistema, funcionando como um roteador. Ele avalia a requisição e inclui o arquivo responsável pela disponibilização do recurso, diminuindo o assim acoplamento do sistema. O arquivo contendo o código do recurso agora só precisa se preocupar em processar os parâmetros da requisição e retornar a solicitação.

### **3.3.1.2 Internacionalização do sistema**

Foi incluído um sistema simples de tradução, onde os textos em português e em inglês são armazenados em um arquivo a parte, e referenciados em código através de um índice para estes arquivos no momento da exibição do texto. Desta maneira é possível eliminar a replicação, facilitando a manutenção do código e a tradução do texto.

## **3.4 Os problemas apresentados**

Da maneira em que foi projetado - levando em conta apenas três atores e funcionalidades para inscrição e seleção de alunos - o sistema se mostrou inadequado para agregar as demais atividades do PPGC. Problemas com a manutenibilidade, interface, e a própria estrutura do banco de dados começaram a dificultar a informatização destes processos, o que começou a apontar para a necessidade de uma reengenharia do sistema, visando o futuro da sua utilização. Algumas das dificuldades encontradas estão descritas nas seções abaixo.

### **3.4.1 Dependência do JavaScript**

O sistema é fundamentalmente dependente do JavaScript. Todas as requisições (exceto a primeira) são efetuadas através de código executado ao lado do cliente, assim como o retorno desta requisição. Apesar dos esforços da comunidade para a evolução e padronização das engines responsáveis pela execução do JavaScript no navegador, ainda existem muitas peculiaridades que

devem ser tratadas individualmente em código. O não-tratamento destas peculiaridades podem resultar na inoperabilidade total do sistema para algum navegador.

Outro impacto negativo desta dependência é sobre a testabilidade do sistema. Apesar de existirem mecanismos para automatização de testes de front-end, a sua execução é feita através de simulações em um navegador real, o que demanda tempo e não garante que problemas de compatibilidade descritos anteriormente não afetem o sistema.

### **3.4.2 Interface defasada**

Outro problema crucial é a defasagem da interface com relação aos padrões de usabilidade atuais. Existe uma demanda cada vez maior pela disponibilização de sistemas compatíveis com dispositivos móveis, o que nos leva a um novo paradigma que deve balizar o desenvolvimento da interface, desde o seu projeto. No próximo capítulo será tratado o conceito de “Responsive Design” e “Mobile First”, técnicas que visam responder a estas demandas.

### **3.4.3 Programação procedural**

PHP é uma linguagem que apenas recentemente evoluiu para o conceito de orientação a objetos. Como comentado anteriormente, o sistema foi construído utilizando uma arquitetura monolítica, procedural. A falta de estruturação do código é um dos principais inimigos da manutenibilidade, pois induz ao alto acoplamento, baixa legibilidade e introduz dificuldades para a automatização de testes.

### **3.4.4 Falta de testes automatizados**

Nenhum dos componentes do sistema está coberto por testes automatizados. Normalmente é efetuado apenas o “debugging” e o teste de aceitação, ambos manuais. Qualquer alteração no sistema demanda reavaliar manualmente potenciais riscos aos demais recursos. Com o crescimento das responsabilidades do sistema, o que se observou foi o aumento da ocorrência de bugs reportados por alterações que geraram efeitos colaterais em outros componentes.

## 4 FUNDAMENTOS E CONCEITOS BÁSICOS

Este capítulo irá introduzir uma série de metodologias e práticas utilizadas para compor o processo de desenvolvimento do sistema final, passando pelas fases de planejamento, desenvolvimento e distribuição.

### 4.1 Modelagem de Processos de Negócios

Processos de negócios consistem em um conjunto de atividades (de sistema, interações de usuário ou manuais), cuja execução coordenada realiza algum objetivo de negócio (WESKE, 2012). A modelagem de processos de negócio é a atividade de representação destes processos, a fim de que o processo atual possa ser analisado, aperfeiçoado e, eventualmente automatizado. O modelo de processo também facilita o compartilhamento de conhecimento sobre o processo entre os envolvidos em sua execução.

A notação para a modelagem de processos de negócio (BPMN) foi desenvolvida sob a coordenação do Object Management Group (OMG). O principal objetivo da BPMN é prover uma notação fácil de ser entendida por todos os usuários de negócio, incluindo analistas que capturam as primeiras definições do processo, desenvolvedores técnicos responsáveis pela implementação do sistema que irá executar o processo e finalmente, aqueles que irão monitorar o processo.

A BPMN é uma notação baseada em diagrama de fluxo para a definição de processo que inclui um mecanismo para a geração de um processo executável (BPEL). Ela apresenta um conjunto mínimo de elementos de modelagem que inclui quatro categorias principais: objetos de fluxo, objetos de conexão, raias e artefatos (THOM, 2011).

### 4.2 Estórias de usuário

Uma estória de usuário é uma maneira de quebrar requisitos em pequenas ações que focam nos objetivos do usuário e como o sistema alcançará estes objetivos (FOWLER, 2013). Estas ações são descritas em linguagem natural, especificando um ator (aquele interessado na funcionalidade), uma ação (o que o ator pretende fazer) e um valor (retorno produtivo da ação). Concentrando-se no *comportamento* da aplicação ao invés da *implementação* da aplicação se torna fácil reduzir mal-entendidos entre os *stakeholders* do projeto.

A figura 4.1 apresenta como uma estória pode ser escrita utilizando um cartão.

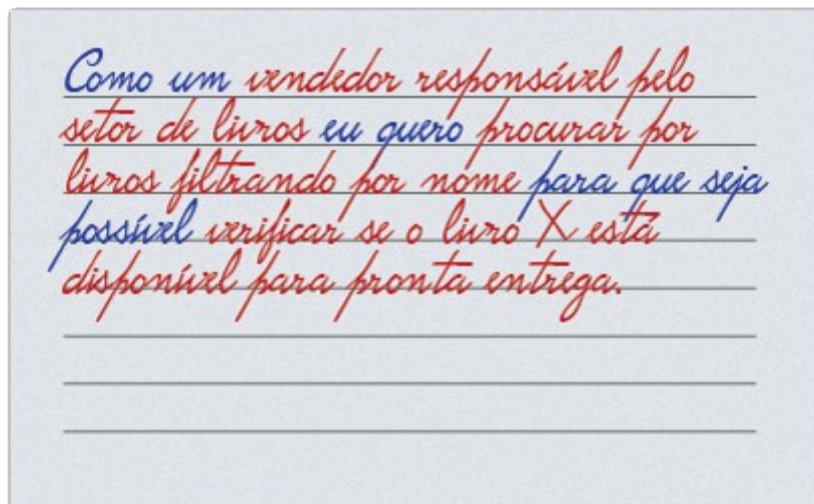


Figura 4.1: Estória de usuário representada na forma de um cartão

A linguagem padronizada para a escrita de cartões com histórias de usuário é considerada uma Linguagem Específica de Domínio (DSL), onde o domínio aplicado em questão seria o de negócios. Esta padronização facilita o reuso do cartão como forma de automatizar testes de aceitação, o que será visto em detalhes na seção de desenvolvimento deste trabalho.

### 4.3 Testes de aceitação e Behaviour-Driven Development (BDD)

Testes de aceitação fornecem critérios básicos que podem ser utilizados para determinar se uma história foi totalmente implementada, conforme os requisitos de especificação ou contrato. Ter critérios que nos dizem quando algo é feito é a melhor maneira de evitar colocar muito - ou muito pouco - tempo e esforço para isso.

Estes testes são escritos aproveitando os cartões de histórias de usuário definido - em uma DSL de negócios, em linguagem natural - diferentes cenários que são interpretados e executados automaticamente por uma suite de testes.

O conceito de BDD parte da premissa de que testes de aceitação provêm uma grande quantidade de informações que o desenvolvedor pode aproveitar durante a codificação da história. Se o teste é escrito antes de que o programador comece o desenvolvimento do código, isto irá lembrar-lhe de tratar casos previstos durante o desenvolvimento do teste. Uma vez que a funcionalidade esteja efetivamente implementada, roda-se a suite de testes para confirmar (ou não) se todas os cenários descritos são cobertos pelo código escrito.

### 4.4 Arquitetura Model-View-Controller (MVC)

O MVC é uma abordagem para separar o código dependendo de qual papel ele interpreta dentro de uma aplicação (GAMMA, 2006). Os componentes básicos desta abordagem

são o modelo (*model*), que trata da manipulação de dados e lógica de negócios; a visão (*view*), que trata da apresentação dos dados junto à interface; e o controlador (*controller*), que em aplicações modernas é o responsável por mediar a comunicação entre o modelo e a visão, bem como aplicar validação aos dados de entrada. Por serem componentes separados, o desenvolvimento destes pode ser feito em paralelo, o que permite que uma equipe se preocupe com a apresentação dos dados, ao mesmo tempo que outra trata de como estes dados são organizados.

#### **4.5 Sistema de versionamento de código distribuído**

O princípio fundamental de design por trás de um sistema de versionamento de código distribuído é que cada desenvolvedor mantenha um repositório auto-contido na sua própria máquina (HUMBLE, 2010). Alterações feitas neste repositório podem ser distribuídas aos demais desenvolvedores, ou a apenas um grupo destes. Desta maneira, em caso de problemas com a máquina contendo o repositório central, qualquer um dos clientes pode ser utilizado para restaurá-lo (CHACON, 2009). Cada checkout é uma cópia completa de todo o conteúdo do repositório (figura 4.2).

Outra grande diferença entre sistemas de versionamento distribuídos e sistemas centralizados é a de que quando você efetua o commit das suas alterações, você as inclui ao seu repositório local. Para que suas alterações sejam compartilhadas com outros usuários, uma série adicional de passos é necessária. Desta maneira, sistemas distribuídos incentivam commits constantes para cada alteração de software, e isto permite uma melhor documentação no registro de alterações do código e uma eventual recuperação de estágios anteriores em caso de problemas.

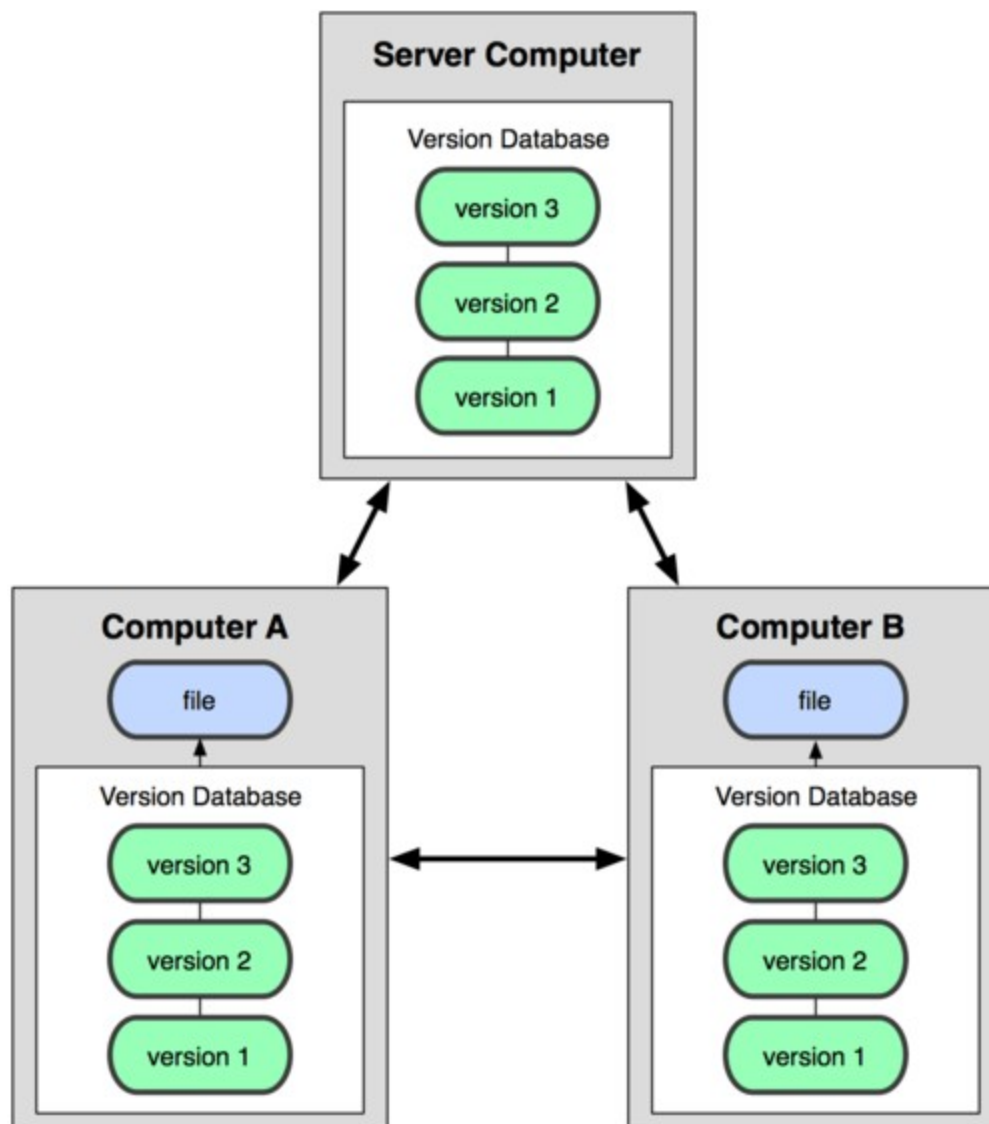


Figura 4.2: Diagrama de controle de versionamento distribuído

#### 4.6 Versionamento do banco de dados

Assim como qualquer outra alteração no sistema, mudanças em bancos de dados devem poder ser gerenciadas através de processos automatizados. Isto significa que a inicialização de um banco de dados e todas as suas migrações devem ser registradas como *scripts*, que de maneira incremental possam aplicar estas alterações ao banco de dados final (HUMBLE, 2010).

Esta abordagem é especialmente necessária quando do desenvolvimento de sistemas utilizando DVCSs, pois cada desenvolvedor trabalha utilizando uma versão diferente do código e do banco de dados que dá suporte a aplicação. Ao integrar código de terceiros ao seu repositório



local, o desenvolvedor pode rodar os *scripts* de migração para efetuar as alterações previamente projetadas ao banco de dados.

## 4.7 Responsive Web Design & Mobile First

Um problema recorrente da camada de apresentação do sistema é a necessidade de lidar com uma quantidade grande - e cada vez maior - de diferentes dispositivos e diferentes resoluções de tela mantendo-se flexível o suficiente para evitar problemas de usabilidade quando acessados (especialmente) por dispositivos de telas menores (NIELSEN, 2013).

Para lidar com tal situação, existem uma série de técnicas que compõe a abordagem *Responsive Web Design* (MARCOTTE, 2011), especialmente a utilização de *media queries* (W3C, 2013) e o conceito de *leiaute* baseado em uma grade fluída.

O conceito *mobile first* (WROBLEWSKI, 2011), sugere que a interface de sistemas web deve ser projetada tendo como prioridade a sua utilização em dispositivos móveis, usando como argumento o grande crescimento do market-share destes dispositivos no acesso de sistemas web; as restrições impostas por dispositivos deste gênero que forçam o desenvolvedor a focar no conteúdo que é realmente necessário, facilitando a visualização das informações e a organização dos dados; e a possibilidade de explorar novas capacidades integradas a navegadores móveis modernos, como por exemplo geolocalização, bússola digital e gestos multi-touch.

## 4.8 Ferramentas de suporte

Nesta seção introduziremos as ferramentas utilizadas para a implementação do sistema e aplicação prática das metodologias.

### 4.8.1 Framework FuelPHP

FuelPHP (FUELPHP, 2013) é um framework de código livre para o desenvolvimento de aplicações web, que implementa o padrão MVC. Escrito em PHP 5.3 e inspirado no Kohana (popular framework em PHP), ele segue os seguintes preceitos arquiteturais:

- Flexibilidade, quase todos os componentes do kernel do framework podem ser estendidos ou substituídos;
- Modularidade, aplicações podem ser divididas em módulos; e
- Extensibilidade, novas funcionalidades podem ser adicionadas ao framework através do uso de pacotes (packages).

Adicionalmente, o framework ainda provém uma implementação de ORM (inspirada no padrão *active records*), abstraindo o acesso ao banco de dados e facilitando a rápida prototipação de novas funcionalidades, bem como o versionamento do banco de dados através de migrações.

### 4.8.2 Behat & Mink para a automatização de testes de aceitação e BDD

Behat é apresentado como "um framework php para testar as suas expectativas de negócio" (BEHAT, 2013). Inspirado no projeto Cucumber - framework para BDD desenvolvido em Ruby - ele utiliza a linguagem Gherkin (GHERKIN, 2013) para a geração de testes de aceitação a partir de uma funcionalidade e de seus cenários.

Mink é um framework de código livre para testes de aceitação (MINK, 2013), e funciona juntamente com o Behat servindo como uma camada de abstração entre o Behat e o browser, que tanto pode ser emulado (através de uma implementação da operação do browser em PHP) ou real (através de controladores externos como o Selenium (SELENIUM, 2013)).

### 4.8.3 Git, Bitbucket e GitFlow

O Git é um sistema distribuído de versionamento de código desenvolvido para lidar com tudo, de pequenos a grandes projetos, com velocidade e eficiência (GIT, 2013). O sistema foi desenvolvido inicialmente por Linus Torvalds para utilização no desenvolvimento do kernel do Linux. Características que destacam o git de outros sistemas existentes são o foco no desenvolvimento não-linear - facilitando o *branching* e *merging* - e o desenvolvimento distribuído através da replicação do repositório e de todo o histórico de alterações entre as máquinas dos desenvolvedores. Um complexo sistema de integração de código identifica e trata conflitos entre códigos em diferentes estágios.

O Bitbucket é um servidor para gerenciamento de código voltado à colaboração e revisão de projetos de código livre ou fechado (BITBUCKET, 2013). No escopo deste trabalho, ele é utilizado como o repositório "origem", e integra o código de todos os desenvolvedores.

GitFlow é um modelo para *branching* utilizando Git desenvolvido por Vincent Driessen (DRIESSEN, 2013). Ele define uma série de regras para o *branching* e *merging* de código. Seu modelo pode ser visualizado graficamente na figura 4.3.

A ideia por trás do GitFlow é organizar a criação, evolução e integração de *branches* dentro do processo de versionamento de código. Basicamente, temos um *branch* "master", onde fica o código estável que roda em produção e um *branch* "develop", com código instável em desenvolvimento. Para cada nova funcionalidade a ser incluída no sistema, um novo *branch* deve ser criado a partir do develop. uma vez que a funcionalidade esteja concluída, o código é reintegrado a *branch* develop através de uma operação de *merge*. O histórico de desenvolvimento é mantido e pode ser visualizado graficamente com a ajuda de ferramentas como o gitk ou o próprio Bitbucket.

Após o desenvolvimento de funcionalidades suficientes para o release, se cria uma *branch* "release", onde o código é preparado para ser enviado a produção (caso haja algum preparativo a ser feito). Alterações efetuadas nesta *branch* devem ser reintegradas a *branch* develop. Uma vez que o código esteja pronto, integramos a *branch* release com o master e efetuamos a implantação deste código no servidor de produção.

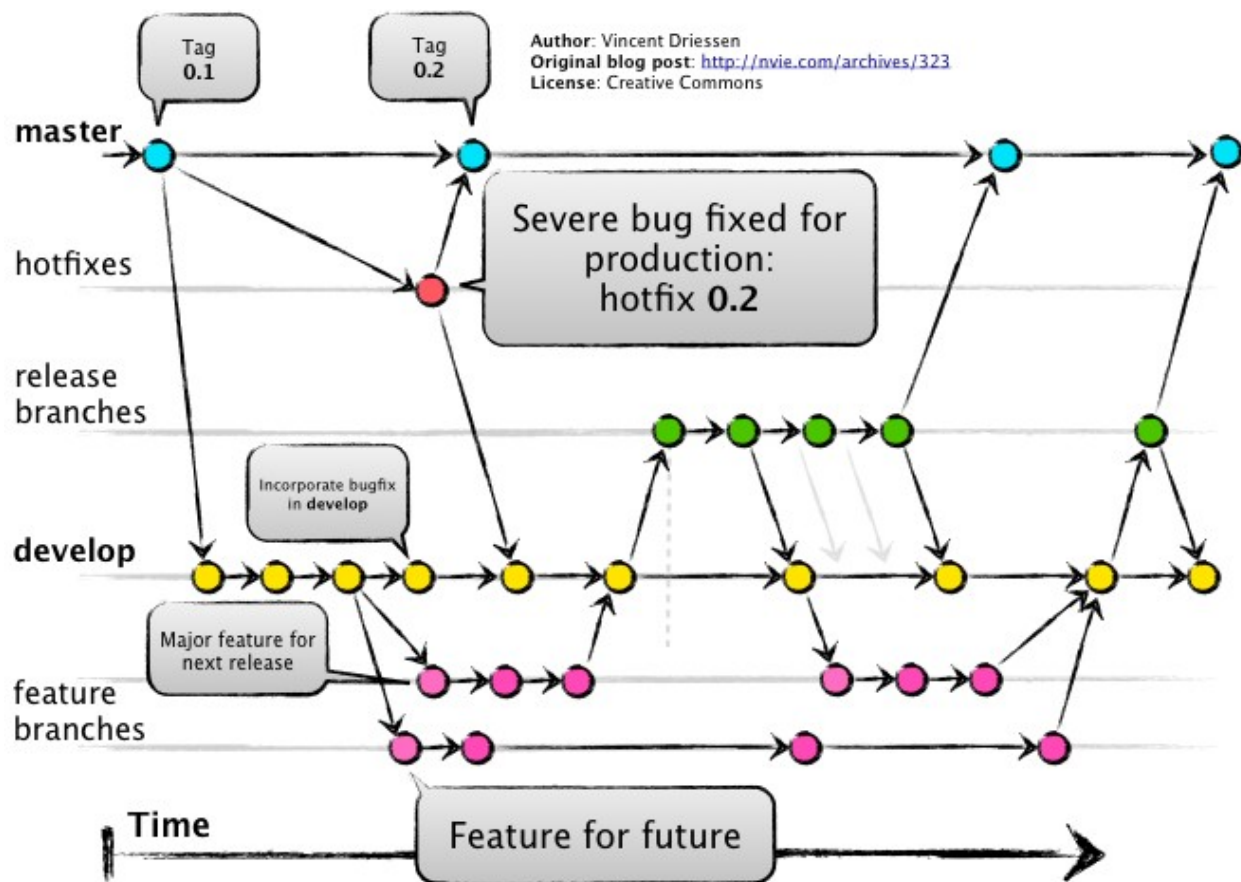


Figura 4.3: Representação gráfica do GitFlow

Hotfixes (bugs com alta severidade) geram *branches* à parte (criados a partir do *branch* master). Uma vez que o bug esteja corrigido, sua correção é integrada novamente ao master e ao develop.

## **5 O SISTEMA DE ADMINISTRAÇÃO DO PPGC**

Neste capítulo desenvolveremos a proposta deste trabalho. Será apresentado o processo de desenvolvimento e as técnicas aplicadas para a reengenharia do Sistema de Inscrições do PPGC para assumir o papel de Sistema de Administração do PPGC, permitindo a inclusão das novas funcionalidades previstas no futuro do programa.

### **5.1 Os requisitos do sistema**

Com a evolução da informatização do PPGC novas funcionalidades foram sendo necessárias, fugindo do escopo original do sistema de inscrições e impondo dificuldades para a operação sobre a estrutura atual do sistema. A proposta de reengenharia do sistema antigo, e desenvolvimento de um sistema de administração do PPGC, se opondo ao sistema de inscrições seguiram os seguintes requisitos:

- O sistema de administração deve reimplementar todas as atividades relacionadas ao corpo docente, secretaria e diretoria do Programa presentes no sistema de inscrição. Isto inclui a seleção de mestrado, doutorado e alunos especiais;
- O sistema deve permitir a informatização de processos de alunos regulares de mestrado e doutorado, com por exemplo a submissão do Plano de Ensino e Pesquisa (PEP), proposta de Trabalho Individual (TI) ou outros documentos;
- O sistema deve absorver as atividades relacionadas ao acompanhamento da produção acadêmica do corpo docente do instituto, com vistas nos processos de credenciamento e recredenciamento de professores. O mesmo sistema também deve permitir o acompanhamento de produção de alunos regulares;
- A interface deve ser modernizada, facilitando o acesso ao sistema através de dispositivos móveis ou de menor resolução;
- Deve haver uma melhor documentação dos processos implementados e do código fonte do sistema.

Para o desenvolvimento deste sistema, iremos implementar os conceitos apresentados no capítulo 4, através das ferramentas apresentadas na sessão 4.8. Por simplicidade, acompanharemos o processo de desenvolvimento de duas funcionalidades: a seleção de alunos de doutorado (já presente no sistema de inscrições) e a submissão de plano de ensino e pesquisa (nova funcionalidade envolvendo alunos de mestrado).

## **5.2 O processo de desenvolvimento**

O processo de desenvolvimento do sistema envolve as fases de entendimento do problema, modelagem do processo, extração de estórias de usuário, criação de testes automatizados com cenários pertinentes para cada estória de usuário, codificação da funcionalidade, aplicação dos testes desenvolvidos anteriormente, e apresentação ao usuário. Cada atividade deste processo será detalhada nas seções abaixo.

### **5.2.1 Projeto e elicitação de requisitos**

Na fase de elicitação, procura-se entender o domínio de aplicação para facilitar a identificação dos requisitos para a modelagem do processo. Quando possível, é feita a priori a análise de documentação pública relacionada ao problema, seja do PPGC, seja de outra instituição. Com as informações coletadas, o desenvolvedor rascunha um modelo para o processo, conforme o seu entendimento. Se identifica então um ou mais indivíduos-chave (normalmente, o solicitante e um dos operadores do processo), e através de entrevista(s), se confirma e aperfeiçoa o modelo projetado inicialmente.

### **5.2.2 Modelagem do processo**

A modelagem final do processo é feita utilizando BPMN e uma ferramenta para modelagem de processos. É sempre interessante que o(s) indivíduo(s)-chave validem a modelagem final antes que comece o processo de desenvolvimento do sistema.

### **5.2.3 Extração das estórias de usuário**

A extração de estórias de usuário deve ser feita preferencialmente durante a validação do modelo do processo pelo(s) indivíduo(s)-chave. As vantagens desta abordagem são que favorecem a criação de estórias de usuário que são escritas na linguagem do negócio, não utilizando jargões técnicos. Isto facilita que os indivíduos consigam priorizar estórias a serem desenvolvidas. Ainda, como principais visionários do processo, os indivíduos-chave estão em melhor posição para descrever o comportamento do sistema (COHN, 2004)

Uma boa estória de usuário deve ser independente, negociável, estimável, pequena, valorosa para o usuário e testável (WAKE, 2013).

### **5.2.4 Criação de cenários para os testes automatizados**

Nesta fase se quebra cada estória de usuário em cenários, que mostram as diferentes maneiras em que a estória pode ser executada. Cada cenário deve em média ser composto por uma sequência de 3 a 8 passos. Preferencialmente, os cenários devem ser escritos de maneira declarativa, que são normalmente menores, mais fáceis de entender e de manter (FOX, 2013).

```
1 # language: pt
2
3 Funcionalidade: Submissão do PEP pelo aluno
4   Para que eu possa efetuar a submissão do meu PEP
5   Como aluno de mestrado
6   Eu devo poder efetuar o upload do meu documento no sistema
7
8   Cenário: Upload do documento
9     Dado que estou na página de submissão do PEP
10    Quando eu carrego um arquivo pdf no campo "Documento"
11    E preencho o campo "Observação" com "Envio do meu plano de ensino e pesquisa"
12    E pressiono o botão "Enviar"
13    Então eu devo ver uma mensagem de sucesso e o PEP na tabela
14
15   Cenário: Preenchimento inválido do formulário
16     Dado que estou na página de submissão do PEP
17     Quando eu carrego um arquivo jpg no campo "Documento"
18     E pressiono o botão "Enviar"
19     Então eu devo ver uma mensagem de erro
20
21   Cenário: Não envio do documento deve retornar erro
22     Dado que estou na página de submissão do PEP
23     Quando eu pressiono o botão "Enviar"
24     Então eu devo ver uma mensagem de erro
25
26   Cenário: Submissão com campo observação vazio não deve ser problema
27     Dado que estou na página de submissão do PEP
28     Quando eu carrego um arquivo pdf no campo "Documento"
29     E pressiono o botão "Enviar"
30     Então eu devo ver uma mensagem de sucesso e o PEP na tabela
31
```

*Figura 5.1: Exemplo de teste de aceitação automatizado*

Um exemplo de teste automatizado escrito utilizando a DSL Gherkin para a submissão de PEP pode ser visto na figura 5.1.

Uma vez que o teste, em linguagem natural esteja escrito, deve-se executá-lo para identificar passos que ainda não foram implementados na suite de testes. Cada linha dentro de um cenário corresponde a um passo, que quando interpretado pela suite executa um código predefinido que emula a execução do passo dentro de uma instância de um navegador, conforme pode ser visto no exemplo da figura 5.2.

```

1 <?php
2
3 class FeatureContext extends MinkContext
4 {
5     /**
6      * @Given /^que estou na pagina (.*)$/
7      */
8     public function queEstouNaPagina($arg1)
9     {
10         switch ($arg1)
11         {
12             case "de entrada":
13             case "homepage":
14                 $page = \Uri::create('/');
15                 break;
16             case "de submissão do pep":
17                 $page = \Uri::create('mestrado/pep');
18                 break;
19             case "Dado que estou na página da listagem dos PEPs":
20                 $page = \Uri::create('mestrado/pep/list');
21                 break;
22             default:
23                 return false;
24         }
25         parent::visit($arg1);
26     }
27
28     /**
29      * @When /^(?:|Eu )carrego um arquivo (.*?) no campo "(?P<field>(?:[^\]|\\|")*)"$/
30      */
31     public function carregoUmArquivoNoCampo($type, $field)
32     {
33         $field = $this->fixStepArgument($field);
34         switch ($type)
35         {
36             case "pdf": $file = test_dir(). 'sample.pdf'; break;
37             case "jpg": $file = test_dir(). 'sample.jpg'; break;
38             default: return false;
39         }
40
41         $this->getSession()->getPage()->attachFileToField($field, $path);
42     }
43
44     ...|
45
46 }
47

```

Figura 5.2: Exemplo de implementação de um passo

Vale lembrar que o desenvolvimento de testes de automação já faz parte do processo de codificação do software e portanto, o arquivo deve ser adicionado ao sistema de versionamento de código. Utilizando o *branching model* citado no capítulo 4, para cada nova funcionalidade descrita deve ser criado um novo *branch*.

### 5.2.5 Desenvolvimento da funcionalidade

Uma vez que o teste esteja escrito, escreve-se apenas o código necessário para que o teste seja executado com sucesso. Segue-se codificando e executando a suite de testes até que todos os cenários executem com sucesso.

### **5.2.6 Validação**

Por fim, o desenvolvedor roda a suite completa de testes para garantir que a integração da funcionalidade ao sistema não afete outras funcionalidades já implementadas. Somente após a execução da suite completa, sem erros é que o código é integrado ao código em produção.

## **5.3 Novas funcionalidades**

Nesta seção iremos apresentar - a nível de exemplo, duas funcionalidades desenvolvidas e já integradas em produção no sistema de administração do PPGC. As funcionalidades escolhidas foram a de seleção de alunos de doutorado - extraída do sistema de inscrições - e a submissão de plano de ensino e pesquisa, nova funcionalidade desenvolvida aproveitando o suporte aos novos atores do sistema (alunos de mestrado, no caso).

### **5.3.1 Seleção de Alunos de Doutorado**

O processo de seleção de alunos de doutorado consiste em uma série de atividades que vão desde o recebimento da inscrição do candidato até a sua seleção (e posterior matrícula no sistema da Universidade) ou não-seleção.

#### **5.3.1.1 Descrição passo-a-passo do processo**

1. O administrador do sistema recebe uma notificação de que um novo candidato ao doutorado efetuou a submissão do cadastro;
2. O administrador acompanha o pagamento da taxa de inscrições, confirmando o recebimento do valor no sistema após a efetivação do depósito na conta da Universidade;
3. O professor indicado recebe uma notificação de que há um candidato ao doutorado aguardando sua avaliação no sistema;
4. O professor avalia a documentação submetida, e indica se aceita ou não a candidatura
  - a. Em caso negativo, o orientador deve preencher uma justificativa, que é registrada no sistema e enviada para o candidato recusado, junto com a mensagem de candidatura indeferida;
5. A comissão de seleção de doutorado recebe a candidatura e avalia a área de atuação e plano de curso do candidato;
6. A comissão de seleção de doutorado convida três membros do corpo docente do PPGC, que atuam na área do trabalho para efetuar a avaliação do candidato;
7. Uma vez que os convidados entrem em acordo sobre uma data e hora para a entrevista, a mesma é registrada através do sistema de administração;
8. A secretaria recebe uma notificação, e deve reservar uma sala na data e hora informadas, e cadastra estes dados no sistema;
9. O candidato recebe um convite para comparecer à entrevista na data e hora marcadas;



10. Na data, hora e local marcados a banca realiza a entrevista com o candidato;
11. O relator da banca preenche a ficha de avaliação do candidato dando o parecer final da banca, que pode ser: aceitar o candidato, recusar o candidato ou orientá-lo a resubmeter a sua proposta em outro processo seletivo;
12. A comissão de seleção de doutorado avalia o resultado das entrevistas, e dá o parecer final quanto à seleção dos alunos;
13. O administrador do sistema gera a lista de alunos aceitos, que é divulgada no site do Programa;
14. O administrador do sistema envia a carta de aceitação ou recusa, com demais instruções caso necessário.

#### **5.3.1.2 Modelagem do processo**

O processo modelado pode ser visualizado na figura 5.3.

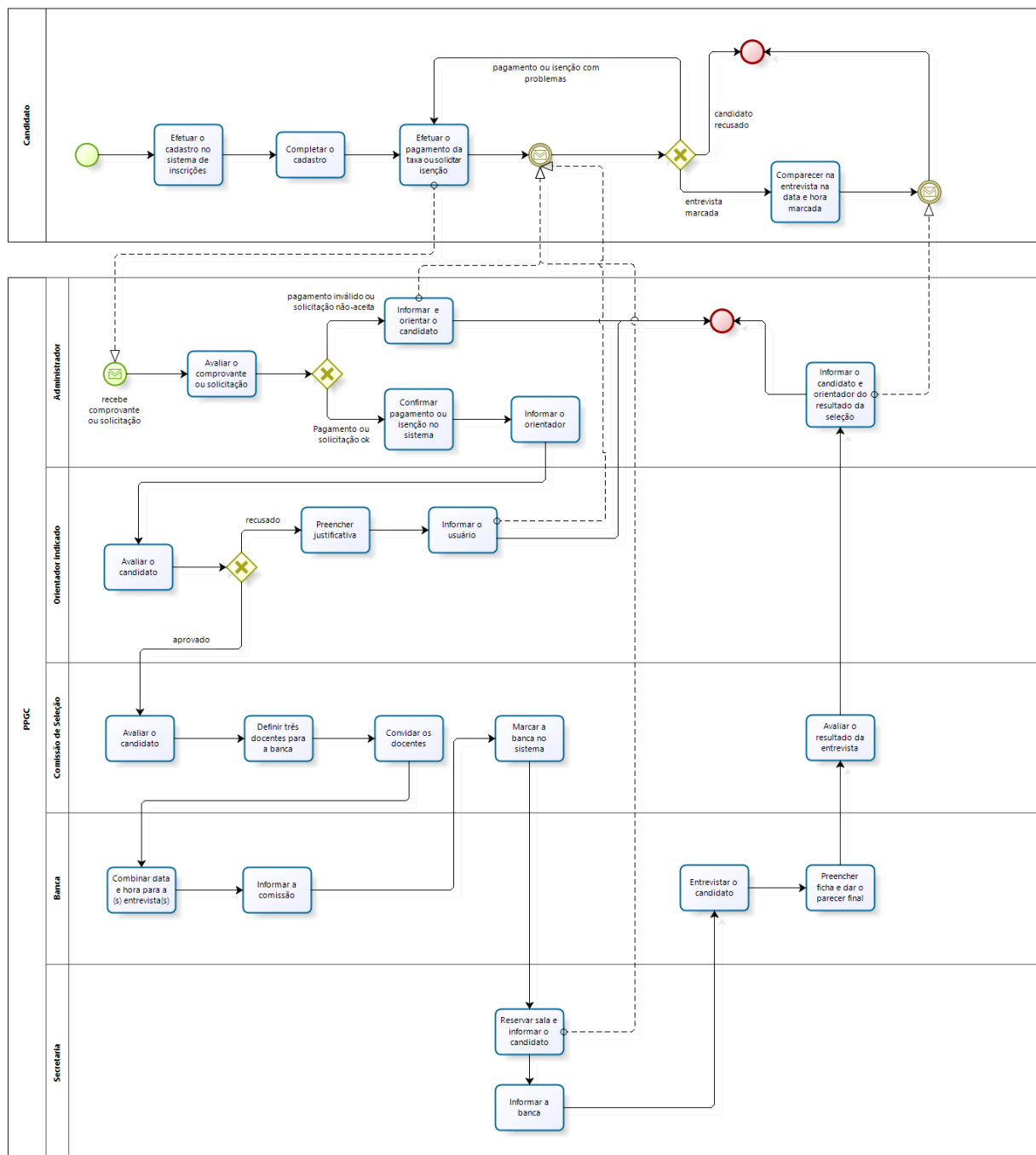


Figura 5.3: Modelagem do processo de seleção de candidatos ao doutorado

### 5.3.1.3 Extração dos casos de uso e estórias de usuário

- **Acompanhamento de inscrições**  
Para que eu possa acompanhar o cadastro de candidatos  
Como um administrador do sistema

- Eu devo poder ver a listagem de pagamentos pendentes
- **Liberação de inscrição**  
 Para que eu possa liberar a inscrição de um candidato  
 Como um administrador do sistema  
 Eu devo poder confirmar o pagamento ou isenção deste candidato
  - **Avaliação de candidato a doutorado**  
 Para que eu possa avaliar um candidato  
 Como um docente do PPGC com um candidato ao doutorado, um membro da comissão de seleção ou da banca de entrevista  
 Eu devo poder ver todos os dados cadastrais do candidato
  - **Indicação de interesse**  
 Para que eu possa indicar interesse na orientação de um candidato  
 Como um docente do PPGC com um candidato de doutorado  
 Eu devo poder confirmar o interesse neste candidato
  - **Recusa de candidato**  
 Para que eu possa indicar que não tenho interesse na orientação de um candidato  
 Como um docente do PPGC com um candidato de doutorado  
 Eu devo poder reprovar a indicação de um candidato
  - **Marcão de bancas**  
 Para liberar as fichas de avaliação de doutorado  
 Como membro da comissão de seleção  
 Eu devo poder indicar uma banca, data, local e entrevistados para o doutorado
  - **Marcação de sala**  
 Para informar a todos o local de uma entrevista  
 Como membro da secretaria do PPGC  
 Eu devo poder registrar a sala e prédio de uma entrevista
  - **Preenchimento da ficha de avaliação**  
 Para que eu possa recomendar o candidato  
 Como membro da banca de entrevista de um candidato ao doutorado  
 Eu devo poder preencher a ficha de avaliação deste candidato
  - **Avaliação final de candidato**  
 Para que eu possa validar as recomendações das bancas  
 Como membro da comissão de seleção do doutorado  
 Eu devo poder visualizar as fichas de avaliação de candidatos

#### 5.3.1.4 Demonstração passo-a-passo

O processo inicia com um candidato inscrito enviando o comprovante de pagamento da taxa de inscrição, ou a solicitação de isenção. Na figura 5.4 está demonstrado a interface para a listagem de pagamentos pendentes no sistema.

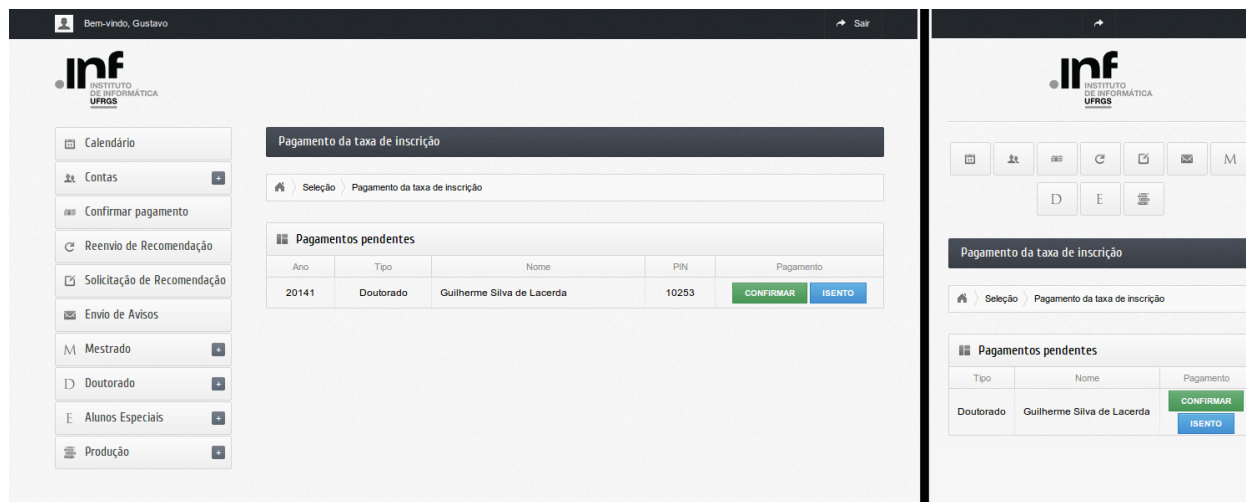


Figura 5.4: Interface para a listagem de pagamentos pendentes

O administrador do sistema confirma o pagamento clicando sobre o botão “Confirmar”. Em caso de solicitação de isenção de pagamento, caso confirmada o administrador pode clicar em “Isento”.

O sistema atualiza o status do candidato e exibe uma mensagem de sucesso (conforme figura 5.5). O orientador recebe uma mensagem por e-mail que ele possui um novo candidato a sua orientação.

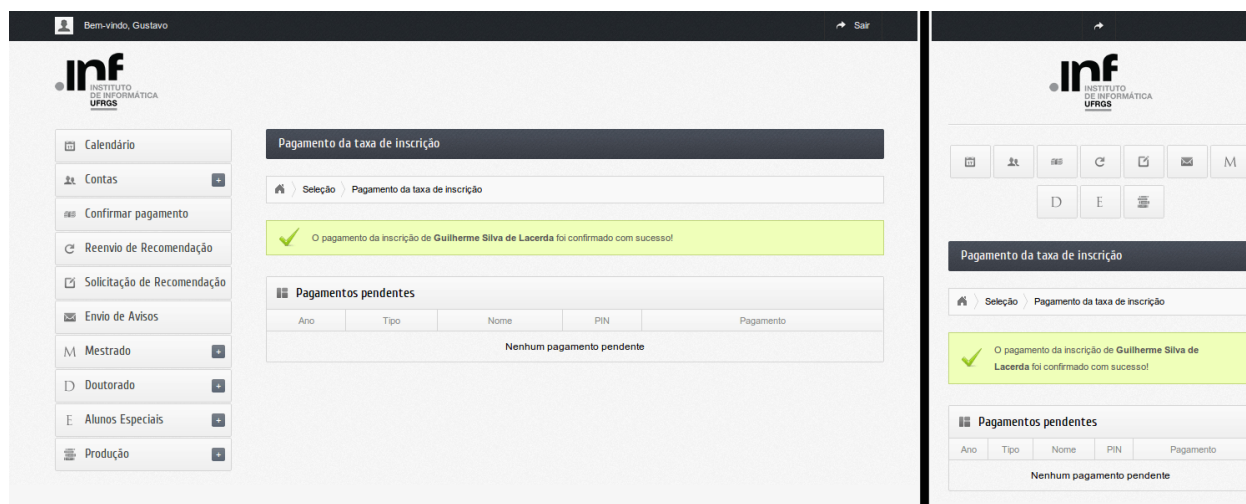



Figura 5.5: Confirmação de pagamento

O orientador, ao acessar o sistema tem a visualização da pendência conforme a figura 5.6. Ele normalmente confere o cadastro do usuário (figura 5.7), clicando no link sobre o seu nome. Para efetuar a recusa da inscrição do candidato, o orientador clica sobre o botão  na interface. É exibido um formulário para a justificativa (figura 5.8), que fica registrada no sistema

e é enviada para o candidato. Para confirmar a inscrição, o orientador clica sobre o ícone ✓. Uma mensagem de sucesso é emitida, e o status do candidato é alterado na tabela (figura 5.9).

The figure displays two side-by-side screenshots of a web application interface for candidate management. Both screenshots show a sidebar on the left with navigation options: 'Calendário', 'Meus Orientandos', 'Mestrado', 'Doutorado', 'Alunos Especiais', and 'Produção'. The main content area is titled 'Doutorado - Seleção' and contains a table of candidates.

**Left Screenshot (Initial State):**

Meus Candidatos			Ativos	Finalizados
Data	Nome	Status	Entrevista	Interesse
24/10/2013	Guilherme Silva de Lacerda	Aguarda confirmação do orientador	-	✓

**Right Screenshot (After Confirmation):**

Meus Candidatos			Ativos	Finalizados
Data	Nome	Status	Entrevista	Interesse
24/10/2013	Guilherme Silva de Lacerda	Aguarda confirmação do orientador	-	✓

Figura 5.6: Listagem de candidatos ao mestrado para aprovação

The image displays two views of a candidate's profile on the INF (Instituto de Informática UFRRS) website. The left view shows the full profile, and the right view is a zoomed-in version of the same profile.

**Left View Profile Details:**

- Header:** Bem-vindo, Marcelo Soares Pimenta
- Navigation:** Calendário, Meus Orientandos, Mestrado, Doutorado, Alunos Especiais, Produção.
- Selecção - [Nome do Curso]**
  - Foto:** [Imagem do candidato]
  - Plano de Curso:** Documento, Orientador: Marcelo Soares Pimenta
  - Dados Básicos:**
    - Data de Nascimento: [Data]
    - Naturalidade: [Localidade]
    - Sexo: Masculino
    - Estado Civil: [Estado]
    - Email: [Email]
    - Telefones: [Número]
  - Dados Adicionais:**
  - Experiências:**
    - Curriculo Lattes: [Link]
    - Curriculo: [Link]
    - Acadêmica: [Descrição]
    - Profissional: [Descrição]
    - Idiomas: [Idiomas]
    - Emprego: [Emprego]
    - Empresa: [Empresa]
  - Bolsa:**
    - Escolha: Não sou candidato a bolsa e farei o curso com dedicação parcial
    - Justificativa: É uma grande chance de continuar e aprofundar meus estudos em uma área que trabalho há alguns anos. Hoje, me sinto mais maduro para isso. Além disso, como docente, sinto a necessidade de me envolver mais profundamente com pesquisa.
  - Produção Acadêmica:**
    - Periódicos
    - Conferências

Ano	Artigo	Periódico	Quais
2010	Atuação em [Título]	[Revista]	-
2011	[Título]	[Revista]	-
2012	[Título]	[Revista]	-
2013	[Título]	[Revista]	-
- Históricos Escolares:** Graduação, Pós-Graduação
- Recomendações:** [Listas de recomendações]
- Cursos de Formação:** [Listas de cursos]

**Right View Profile Details:**

- Header:** Bem-vindo, Marcelo Soares Pimenta
- Navigation:** Calendário, Meus Orientandos, Mestrado, Doutorado, Alunos Especiais, Produção.
- Selecção - [Nome do Curso]**
  - Foto:** [Imagem do candidato]
  - Plano de Curso:** Documento, Orientador: Marcelo Soares Pimenta
  - Dados Básicos:**
    - Data de Nascimento: [Data]
    - Naturalidade: [Localidade]
    - Sexo: Masculino
    - Estado Civil: [Estado]
    - Email: [Email]
    - Telefones: [Número]
  - Dados Adicionais:**
  - Experiências:**
    - Curriculo Lattes: [Link]
    - Curriculo: [Link]
    - Acadêmica: [Descrição]
    - Profissional: [Descrição]
    - Idiomas: [Idiomas]
    - Emprego: [Emprego]
    - Empresa: [Empresa]
  - Bolsa:**
    - Escolha: Não sou candidato a bolsa e farei o curso com dedicação parcial
    - Justificativa: É uma grande chance de continuar e aprofundar meus estudos em uma área que trabalho há alguns anos. Hoje, me sinto mais maduro para isso. Além disso, como docente, sinto a necessidade de me envolver mais profundamente com pesquisa.
  - Produção Acadêmica:**
    - Periódicos
    - Conferências

Ano	Artigo	Periódico	Quais
2010	Atuação em [Título]	[Revista]	-
2011	[Título]	[Revista]	-
2012	[Título]	[Revista]	-
2013	[Título]	[Revista]	-
- Históricos Escolares:** Graduação, Pós-Graduação
- Recomendações:** [Listas de recomendações]
- Cursos de Formação:** [Listas de cursos]

**Footer:** Clique aqui para visualizar a versão para impressão desta ficha

Figura 5.7: Visualização do cadastro do candidato

Figura 5.8: Recusa de candidato, com justificativa

Assim que a seleção do candidato é confirmada, os membros da comissão de seleção do doutorado já podem visualizar os seus dados, e preparar a marcação da banca pra entrevista de doutorado. A interface (figura 5.10) apresenta três tabelas: lista de bancas já marcadas, calendário com a agenda de entrevistas e listagem dos candidatos para um determinado processo seletivo.

Doutorado - Seleção						
Selecção > Doutorado > Meus Candidatos						
<p>✓ Você selecionou o candidato Guilherme Silva de Lacerda com sucesso</p>						
Meus Candidatos					Ativos	Finalizados
Data	Nome	Status	Entrevista	Interesse		
24/10/2013	Guilherme Silva de Lacerda	Aguarda marcação de entrevista	-	Confirmado		

Figura 5.9: Candidato confirmado pelo orientador

The image displays two views of a web application for the PhD selection process. The left view is a detailed dashboard, and the right view is a simplified version.

**Left View (Detailed Dashboard):**

- Header:** "Bem-vindo, Manuel Menezes De Oliveira Neto" and "Sair".
- Logo:** "inf INSTITUTO DE INFORMÁTICA UFRGS".
- Sidebar:**
  - Calendário
  - Meus Orientandos
  - Mestrado
  - Doutorado
  - Alunos Especiais
  - Produção
- Dashboard:**
  - 1 Candidatos inscritos
  - 1 Aguardando entrevista
  - 0 Aprovados
  - 0 Recusados
- Bancas Marcadas:**

Banca	Candidato	Data	Ficha
Nenhuma banca marcada ainda			
- Agenda:**

November 2013						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7
- Candidatos:**

Data	Nome	Orientador	Status
24/10/2013	Guilherme Silva de Lacerda	Marcelo Soares Pimenta	Aguarda marcação de entrevista

**Right View (Simplified Dashboard):**

- Header:** "Bem-vindo, Manuel Menezes De Oliveira Neto" and "Sair".
- Logo:** "inf INSTITUTO DE INFORMÁTICA UFRGS".
- Navigation:** Home, Selection, PhD.
- Dashboard:**
  - 1 Candidatos inscritos
  - 1 Aguardando entrevista
  - 0 Aprovados
  - 0 Recusados
- Bancas Marcadas:**

Banca	Candidato	Data	Ficha
Nenhuma banca marcada ainda			
- Agenda:**

November 2013						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7
- Candidatos:**

Data	Nome	Orientador	Status
24/10/2013	Guilherme Silva de Lacerda	Marcelo Soares Pimenta	Aguarda marcação de entrevista

Figura 5.10: Panorama do processo seletivo do doutorado

Para marcar uma banca, o membro da comissão responsável avalia a área de atuação do candidato (pelo cadastro, figura 5.7) e convida 3 membros para compor a banca. Estes organizam entre si uma data e horário para a realização da entrevista. Após a confirmação, o responsável cadastra a banca no sistema (figura 5.11), indicando seus membros, o(s) entrevistado(s) e o(s) horário(s) da(s) entrevista(s).



The image displays two side-by-side screenshots of a web application interface for scheduling interviews. The left screenshot shows the 'Cadastro de bancas' (Bank Registration) form, and the right screenshot shows the 'Agendamento de entrevistas' (Interview Scheduling) form.

**Left Screenshot (Cadastro de bancas):**

- Header: Bem-vindo, Manuel Menezes De Oliveira Neto | Sair
- Logo: inf INSTITUTO DE INFORMÁTICA UFRGS
- Navigation: Calendário, Meus Orientandos, M Mestrado, D Doutorado, E Alunos Especiais, Produção
- Section: Agendamento de entrevistas
- Breadcrumbs: Seleção > Doutorado > Entrevistas > Agendamento
- Form:
  - Membros da Banca: Nicolas Bruno Maillard x, Valter Roesler x, Antonio Carlos Da Rocha Costa x
  - Candidatos: Guilherme Silva de Lacerda x
  - Guilherme Silva de Lacerda: 09/12/2013, 13:20
  - Button: MARCAR ENTREVISTAS
- Message: Uma vez marcada a banca, a secretária receberá uma notificação para efetuar a reserva da sala na data/hora marcada. Somente após a reserva da sala que o candidato receberá a notificação com informações para a entrevista

**Right Screenshot (Agendamento de entrevistas):**

- Header: inf INSTITUTO DE INFORMÁTICA UFRGS
- Navigation: M, D, E
- Section: Agendamento de entrevistas
- Breadcrumbs: Seleção > Doutorado > Entrevistas > Agendamento
- Form:
  - Membros da Banca: Nicolas Bruno Maillard x, Valter Roesler x, Antonio Carlos Da Rocha Costa x
  - Candidatos: Guilherme Silva de Lacerda x
  - Guilherme Silva de Lacerda: 09/12/2013, 16:20
  - Button: MARCAR ENTREVISTAS
- Message: Uma vez marcada a banca, a secretária receberá uma notificação para efetuar a reserva da sala na data/hora marcada. Somente após a reserva da sala que o candidato receberá a notificação com informações para a entrevista

*Figura 5.11: Cadastro de bancas*

Após a confirmação da banca, o responsável recebe uma mensagem de sucesso, o candidato é adicionado a listagem de bancas marcadas (figura 5.12) e a secretaria recebe uma notificação para marcar o local da entrevista.

Bem-vindo, Manuel Menezes De Oliveira Neto

**inf** INSTITUTO DE INFORMÁTICA UFRGS

Calendário  
Meus Orientandos  
Mestrado  
Doutorado  
Alunos Especiais  
Produção

**Doutorado - Panorama**

Seleção > Doutorado

Banca marcada para o dia 09/12/2013 as 13:20  
Membros: Nicolas Bruno Maillard, Valter Roesler, Antonio Carlos Da Rocha Costa  
Entrevistados: Guilherme Silva de Lacerda

1 Candidatos inscritos    1 Aguardando entrevista    0 Aprovados    0 Recusados

**Bancas Marcadas** Marcar Banca

Banca	Candidato	Data	Ficha
Nicolas Bruno Maillard Valter Roesler Antonio Carlos Da Rocha Costa	Guilherme Silva de Lacerda	09/12/2013 13:20	-

**Agenda** month week day

November 2013

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

**Candidatos**

Data	Nome	Orientador	Status
24/10/2013	Guilherme Silva de Lacerda	Marcelo Soares Pimenta	Aguarda entrevista

**inf** INSTITUTO DE INFORMÁTICA UFRGS

Seleção > Doutorado

Banca marcada para o dia 09/12/2013 as 13:20  
Membros: Nicolas Bruno Maillard, Valter Roesler, Antonio Carlos Da Rocha Costa  
Entrevistados: Guilherme Silva de Lacerda

1 Candidatos inscritos    1 Aguardando entrevista    0 Aprovados    0 Recusados

**Bancas Marcadas** Marcar Banca

Banca	Candidato	Data	Ficha
Nicolas Bruno Maillard Valter Roesler Antonio Carlos Da Rocha Costa	Guilherme Silva de Lacerda	09/12/2013 13:20	-

**Agenda** month week day

November 2013

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

**Candidatos**

Data	Nome	Orientador	Status
24/10/2013	Guilherme Silva de Lacerda	Marcelo Soares Pimenta	Aguarda entrevista

Figura 5.12: Confirmação de agendamento da banca

A secretaria acessa o sistema e pode visualizar a lista de pendências para alocação de sala. No caso da figura 5.13, há apenas uma pendência para a entrevista. Para cada entrevista, a secretaria irá registrar a sala e prédio onde ela ocorrerá (figura 5.14). O candidato é informado por email da data, hora e sala da entrevista. Os membros da banca, após a marcação de todas as entrevistas de uma banca são avisados do local reservado para cada entrevista.

The left screenshot shows a user interface for room reservation. At the top, it says "Bem-vindo, Elisane" and "Saír". The main content area is titled "Lista de entrevistas" and includes a breadcrumb trail: "Seleção > Doutorado > Entrevista > Locais". Below this is a section for "Agendamento de salas para entrevistas" with a table:

Banca	Entrevistados	Data/Hora	Local
Nicolas Bruno Maillard Valter Roesler Antonio Carlos Da Rocha Costa	Guilherme Silva de Lacerda	09/12/2013 13:20	Reservar sala

The right screenshot shows a similar view but with a calendar navigation bar at the top containing icons for "Calendário", "M" (Mestrado), and "D" (Doutorado). The table content is identical to the left screenshot.

Figura 5.13: Listagem de pendências para reserva de salas

The left screenshot shows a user interface for marking a room for an interview. At the top, it says "Bem-vindo, Elisane" and "Saír". The main content area is titled "Local de entrevista de Guilherme Silva de Lacerda" and includes a breadcrumb trail: "Seleção > Doutorado > Guilherme Silva de Lacerda > Entrevista > Local". Below this is a section for "Agendamento de entrevista" with a form:

Candidato: Guilherme Silva de Lacerda

Orientador: Marcelo Soares Pimenta

Banca: Nicolas Bruno Maillard  
Valter Roesler  
Antonio Carlos Da Rocha Costa

Data: 09/12/2013 às 13:20

Sala:

Prédio:

Entrevista será por skype:

The right screenshot shows a similar view but with a calendar navigation bar at the top containing icons for "Calendário", "M" (Mestrado), and "D" (Doutorado). The form content is identical to the left screenshot.

Figura 5.14: Marcação de sala para entrevista de doutorado

Durante a banca, os entrevistadores podem acessar o sistema para verificar os dados do candidato. Ao final da entrevista, eles devem preencher a ficha da entrevista (figura 5.15), onde uma série de campos relativos a análise do candidato devem ser preenchidos. O membro da banca pode salvar o formulário para posterior conferência clicando em “Salvar Avaliação”. Para submeter a avaliação, ele clica em “Submeter Avaliação”. Uma vez submetida, a avaliação não pode ser alterada.

The image displays two versions of a web application interface for evaluating a PhD candidate. The left version is the full form, and the right version is a simplified or mobile-optimized view.

**Left Screenshot (Full Form):**

- Header:** "inf INSTITUTO DE INFORMÁTICA UFRGS". User: "Bem-vindo, Valter Roesler".
- Navigation:** Calendário, Meus Orientandos, Mestrado, Doutorado, Alunos Especiais, Produção.
- Breadcrumb:** Seleção > Doutorado > Guilherme Silva de Lacerda > Entrevista
- Entrevista:** Entrevistado: Guilherme Silva de Lacerda; Banca: Nicolas Bruno Maillard, Valter Roesler, Antonio Carlos Da Rocha Costa; Data: 09/12/2013 às 13:20; Local: sala 215, prédio 43421.
- Avaliação da Proposta:** Objectivos claros, específicos, mensuráveis, realistas; contribuição relevante. Includes a "Nível" dropdown and a "Comentários" text area.
- Considerações Finais:** Probabilidade de concluir o doutorado, Recomendação, and a large "Comentários para o orientador e proponente" text area.
- Submeter avaliação:** "Esta avaliação ainda não foi submetida. Uma vez submetida, a avaliação não poderá ser alterada. Você pode salvar este formulário para continuar preenchendo posteriormente." Buttons: SUBMITER AVALIAÇÃO (green), SALVAR AVALIAÇÃO (blue).
- Avaliação Pessoal (Detailed):**
  - Histórico Escolar:** "Nível da insituição de graduação o ou mestrado; notas obldidas na graduação ou no mestrado; tempo e área da graduação ou do mestrado; prêmios e distinções". Includes "Nível" dropdown and "Comentários" text area.
  - Conhecimento e Experiência:** "Publicações com contribuição relevante do aluno na área da proposta; disciplinas cursadas com sucesso na área da proposta; participação em congressos, seminários ou similar na área da proposta." Includes "Nível" dropdown and "Comentários" text area.
  - Maturidade:** "Conhecimento de revistas, congressos, grupos de pesquisa nacionais e internacionais; conhecimento do ambiente acadêmico; participação em intercâmbios; experiência com aluno de iniciação científica, monitor; atividades complementares (e.g. maratonas de programação) ou extra-curriculares." Includes "Nível" dropdown and "Comentários" text area.
  - Motivação:** "Contribuição na elaboração da proposta; dedicação para o doutorado; objetivos de medio e longo prazo; interesse pessoal na solução do problema." Includes "Nível" dropdown and "Comentários" text area.
  - Potencial:** "Independência e iniciativa; capacidade de entender e resolver problemas; criatividade." Includes "Nível" dropdown and "Comentários" text area.

**Right Screenshot (Simplified Form):**

- Header:** "inf INSTITUTO DE INFORMÁTICA UFRGS".
- Navigation:** Home, M, D, E, Print icons.
- Breadcrumb:** Seleção > Doutorado > Guilherme Silva de Lacerda > Entrevista
- Entrevista:** Same data as the full form.
- Avaliação da Proposta:** Same as the full form.
- Considerações Finais:** Same as the full form.
- Submeter avaliação:** Same as the full form.

Figura 5.15: Ficha de avaliação de candidato de doutorado

Após a submissão, além da mensagem de sucesso a entrevista fica marcada como realizada e o parecer final da banca fica visível no sistema (figura 5.16)

The figure displays two screenshots of the INF (Instituto de Informática UFPA) system interface, showing the 'Entrevistas' (Interviews) section. The left screenshot shows the 'Ficha de avaliação de Guilherme Silva de Lacerda enviada com sucesso!' (Evaluation form for Guilherme Silva de Lacerda sent successfully) message. Below this, there is a section for 'Bancas ativas' (Active Committees) with a candidate profile for Guilherme Silva de Lacerda (Marcelo Soares Pimenta) and a 'Parecer: Aceito' (Opinion: Accepted) status. The right screenshot shows the same interface after the evaluation has been submitted, with the 'Parecer' field filled with 'Aceito'.

**Left Screenshot (Before Submission):**

- Header:** Bem-vindo, Valtter Roesler
- Navigation:** Seleção > Doutorado > Entrevistas
- Message:** Ficha de avaliação de Guilherme Silva de Lacerda enviada com sucesso!
- Info:** Abaixo estão listadas as bancas de seleção de doutorado ao qual você foi designado(a), agrupadas por processo seletivo. Para visualizar os dados do candidato, clique sobre o seu nome. Para acessar e preencher a ficha de avaliação, clique sobre o link em "Preencher ficha", ao lado do campo "Parecer"
- Bancas ativas:**
  - Candidato: Guilherme Silva de Lacerda (Marcelo Soares Pimenta)
  - Banca: Nicolas Bruno Maillard, Valtter Roesler, Antonio Carlos Da Rocha Costa
  - Data: 09/12/2013 13:20
  - Local: sala 215, prédio 43421
  - Parecer: Aceito (Visualizar ficha)
- Agenda:** Nov 24 – 30 2013
- Bancas Passadas:** Nenhuma banca foi encontrada para outros processos seletivos

**Right Screenshot (After Submission):**

- Header:** Bem-vindo, Valtter Roesler
- Navigation:** Seleção > Doutorado > Entrevistas
- Message:** Ficha de avaliação de Guilherme Silva de Lacerda enviada com sucesso!
- Info:** Abaixo estão listadas as bancas de seleção de doutorado ao qual você foi designado(a), agrupadas por processo seletivo. Para visualizar os dados do candidato, clique sobre o seu nome. Para acessar e preencher a ficha de avaliação, clique sobre o link em "Preencher ficha", ao lado do campo "Parecer"
- Bancas ativas:**
  - Candidato: Guilherme Silva de Lacerda (Marcelo Soares Pimenta)
  - Banca: Nicolas Bruno Maillard, Valtter Roesler, Antonio Carlos Da Rocha Costa
  - Data: 09/12/2013 13:20
  - Local: sala 215, prédio 43421
  - Parecer: Aceito (Visualizar ficha)
- Agenda:** Nov 24 – 30 2013
- Bancas Passadas:** Nenhuma banca foi encontrada para outros processos seletivos

Figura 5.16: Ficha de avaliação submetida

Os membros da comissão de seleção do doutorado agora possuem acesso a ficha da entrevista, de onde podem confirmar a seleção do candidato ou não (figura 5.17). A lista de candidatos aceitos é divulgada no site da Instituição.

The image displays two side-by-side screenshots of a web application interface for a candidate interview report. The left screenshot shows a detailed view with a sidebar menu and a breadcrumb trail. The right screenshot shows a simplified version of the same report. Both reports are titled "Entrevista com Guilherme Silva de Lacerda" and contain sections for "Entrevista", "Avaliação da Proposta", "Considerações Finais", and "Avaliação Pessoal".

**Left Screenshot (Detailed View):**

- Sidebar:**
  - Calendário
  - Meus Orientandos
  - Mestrado
  - Doutorado
  - Alunos Especiais
  - Produção
- Breadcrumb:** Seleção > Doutorado > Guilherme Silva de Lacerda > Entrevista
- Entrevista:**
  - Entrevistado: Guilherme Silva de Lacerda
  - Banca: Nicolas Bruno Maillard, Valter Roesler, Antonio Carlos Da Rocha Costa
  - Data: 09/12/2013 às 13:20
  - Local: sala 215, prédio 43421
- Avaliação da Proposta:**
  - Objetivos claros, específicos, mensuráveis, realistas; contribuição relevante
  - Nível: Alto
  - Comentários: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi aliquet tellus eu enim cursus venenatis. Nulla sed turpis non nibh pulvinar pulvinar. Sed ac ultrices enim. Duis lacinia urna id aliquam dapibus.
- Considerações Finais:**
  - Probabilidade de concluir o doutorado: Alto
  - Recomendação: Aceitar
  - Comentários: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi aliquet tellus eu enim cursus venenatis. Nulla sed turpis non nibh pulvinar pulvinar. Sed ac ultrices enim. Duis lacinia urna id aliquam dapibus.
  - Comentários para o orientador e proponente: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi aliquet tellus eu enim cursus venenatis. Nulla sed turpis non nibh pulvinar pulvinar. Sed ac ultrices enim. Duis lacinia urna id aliquam dapibus.
- Avaliação Pessoal:**
  - Histórico Escolar:
    - Nível da instituição de graduação o ou mestrado; notas obtidas na graduação ou no mestrado; tempo e área da graduação ou do mestrado; prêmios e distinções
  - Nível: Alto
  - Comentários: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi aliquet tellus eu enim cursus venenatis. Nulla sed turpis non nibh pulvinar pulvinar. Sed ac ultrices enim. Duis lacinia urna id aliquam dapibus.
  - Conhecimento e Experiência:
    - Publicações com contribuição relevante do aluno na área da proposta; disciplinas cursadas com sucesso na área da proposta; participação em congressos, seminários ou similar na área da proposta.
  - Nível: Alto
  - Comentários: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi aliquet tellus eu enim cursus venenatis. Nulla sed turpis non nibh pulvinar pulvinar. Sed ac ultrices enim. Duis lacinia urna id aliquam dapibus.
  - Maturidade:
    - Conhecimento de revistas, congressos, grupos de pesquisa nacionais e internacionais; conhecimento do ambiente acadêmico; participação em intercâmbios; experiência com aluno de iniciação científica, monitor; atividades complementares (e.g. maratonas de programação) ou extracurriculares.
  - Nível: Médio
  - Comentários: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi aliquet tellus eu enim cursus venenatis. Nulla sed turpis non nibh pulvinar pulvinar. Sed ac ultrices enim. Duis lacinia urna id aliquam dapibus.
  - Motivação:
    - Contribuição na elaboração da proposta; dedicação para o doutorado; objetivos de médio e longo prazo; interesse pessoal na solução do problema.
  - Nível: Médio
  - Comentários: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi aliquet tellus eu enim cursus venenatis. Nulla sed turpis non nibh pulvinar pulvinar. Sed ac ultrices enim. Duis lacinia urna id aliquam dapibus.
  - Potencial:
    - Independência e iniciativa; capacidade de entender e resolver problemas; criatividade.

**Right Screenshot (Simplified View):**

- Breadcrumb:** Seleção > Doutorado > Guilherme Silva de Lacerda > Entrevista
- Entrevista:**
  - Entrevistado: Guilherme Silva de Lacerda
  - Banca: Nicolas Bruno Maillard, Valter Roesler, Antonio Carlos Da Rocha Costa
  - Data: 09/12/2013 às 13:20
  - Local: sala 215, prédio 43421
- Avaliação da Proposta:**
  - Objetivos claros, específicos, mensuráveis, realistas; contribuição relevante
  - Nível: Alto
  - Comentários: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi aliquet tellus eu enim cursus venenatis. Nulla sed turpis non nibh pulvinar pulvinar. Sed ac ultrices enim. Duis lacinia urna id aliquam dapibus.
- Considerações Finais:**
  - Probabilidade de concluir o doutorado: Alto
  - Recomendação: Aceitar
  - Comentários: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi aliquet tellus eu enim cursus venenatis. Nulla sed turpis non nibh pulvinar pulvinar. Sed ac ultrices enim. Duis lacinia urna id aliquam dapibus.
  - Comentários para o orientador e proponente: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi aliquet tellus eu enim cursus venenatis. Nulla sed turpis non nibh pulvinar pulvinar. Sed ac ultrices enim. Duis lacinia urna id aliquam dapibus.
- Avaliação Pessoal:**
  - Histórico Escolar:
    - Nível da instituição de graduação o ou mestrado; notas obtidas na graduação ou no mestrado; tempo e área da graduação ou do mestrado; prêmios e distinções
  - Nível: Alto
  - Comentários: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi aliquet tellus eu enim cursus venenatis. Nulla sed turpis non nibh pulvinar pulvinar. Sed ac ultrices enim. Duis lacinia urna id aliquam dapibus.

Figura 5.17: Visualização da ficha da entrevista de candidato ao doutorado

### 5.3.2 Submissão do Plano de Ensino e Pesquisa (PEP)

O plano de ensino e pesquisa (PEP) é um documento criado pelo aluno de mestrado ao final do seu primeiro ano de curso. Ele deve tratar, sucintamente da proposta de pesquisa do aluno, contendo uma introdução ao problema, referências de trabalhos similares, a proposta de solução, a metodologia a ser adotada e um cronograma de trabalho.

Após o desenvolvimento do plano, o aluno deverá submetê-lo através do sistema para a apreciação do orientador (e co-orientador) e posterior arquivamento na secretaria.

### **5.3.2.1 Descrição passo-a-passo do processo**

1. O aluno cria o PEP
2. O aluno submete o documento através do sistema do PPGC
3. O orientador e co-orientador (caso exista) do aluno recebem notificação para avaliar o documento
4. O orientador e co-orientador (caso exista) avaliam o documento
  - a. No caso de reprovação por qualquer um dos orientadores uma justificativa é necessária. Esta justificativa é enviada para o candidato e o PEP é marcado como reprovado
  - b. No caso de aprovação por todos os orientadores, o sistema envia mensagem para o aluno e para a secretaria informando que o PEP foi aceito
5. A secretaria acompanha a entrega de PEPs e reforça a solicitação pelo envio do documento aos alunos que atrasarem a entrega

### **5.3.2.2 Modelagem do processo**

O processo de submissão do plano de ensino e pesquisa, modelado utilizando a BPMN pode ser visto na figura 5.18.

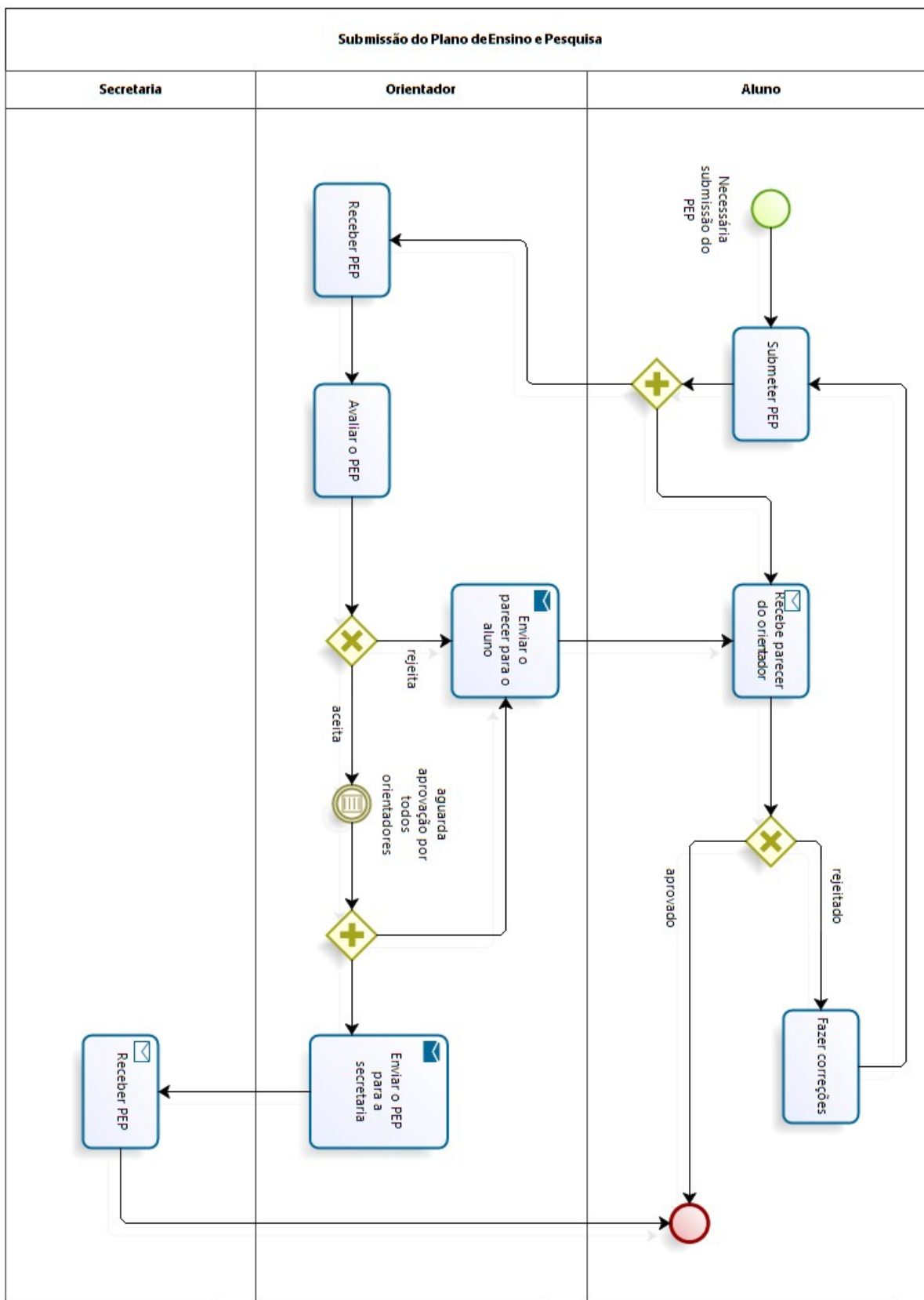


Figura 5.18: Modelagem do processo de submissão do plano de ensino e pesquisa

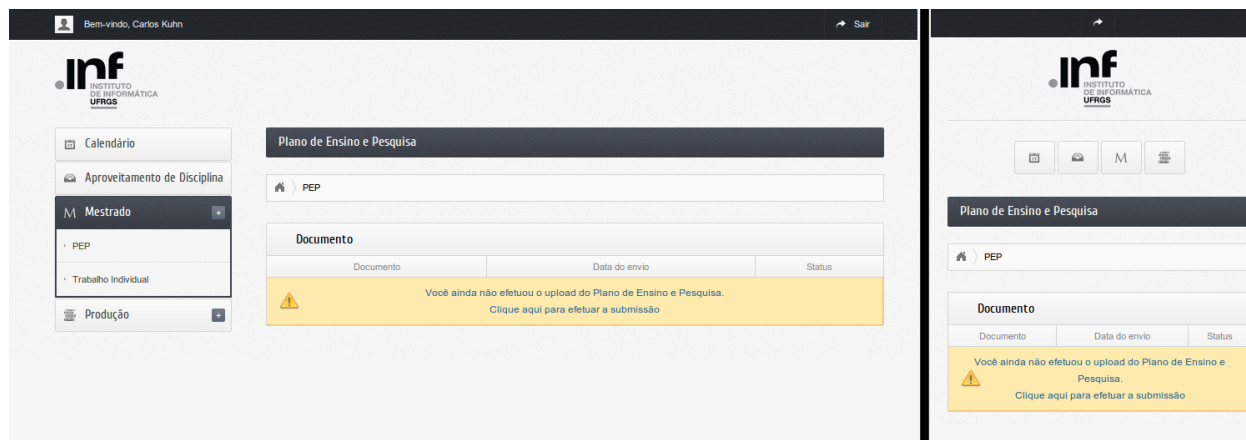


### 5.3.2.3 Extração de casos de uso e estórias de usuário

- **Upload do PEP**  
Para que eu possa efetuar a submissão do meu PEP  
Como aluno de mestrado  
Eu devo poder efetuar o upload do documento no sistema
- **Visualização dos PEPs enviados**  
Para que eu possa avaliar os PEPs dos meus orientandos de mestrado  
Como professor  
Eu devo poder visualizar uma listagem de todos os PEPs submetidos pelos meus alunos
- **Reprovação de PEP**  
Para que eu possa devolver o PEP do meu orientando para correções  
Como professor  
Eu devo poder reprovar o PEP deste aluno, com justificativa
- **Aprovação de PEP**  
Para que o meu orientando possa prosseguir o seu curso  
Como professor  
Eu devo poder aprovar o PEP deste aluno
- **Acompanhamento do recebimento dos PEPs**  
Para que eu possa acompanhar a submissão dos PEPs dos alunos de mestrado  
Como um membro da secretaria  
Eu devo poder visualizar a listagem de todos os alunos que devem enviar o PEP, bem como o status da submissão

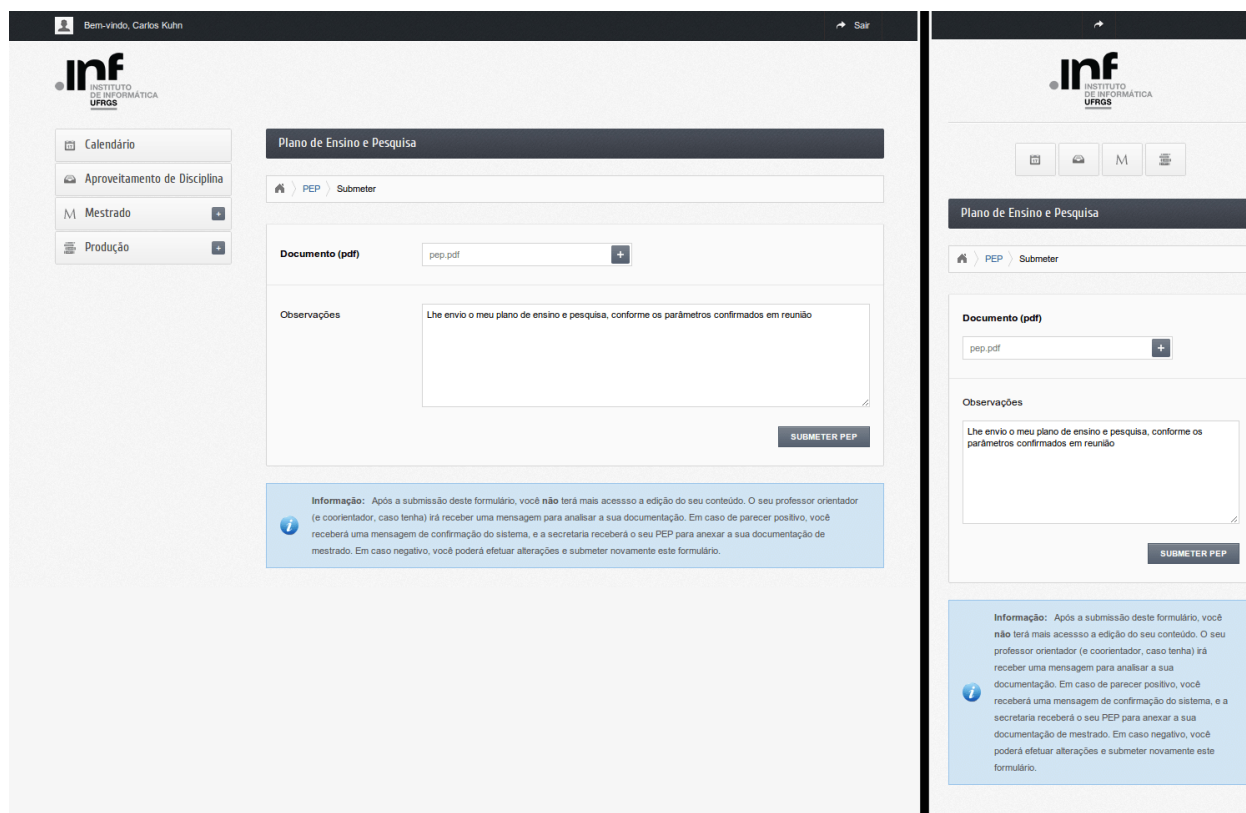
### 5.3.2.4 Demonstração passo-a-passo

Todas as figuras desta demonstração são visualizadas na versão completa (1024 pixels de largura), compatível com desktops e notebooks e compacta (360 pixels de largura), compatível com celulares.



*Figura 5.19: Interface para a submissão do plano de ensino e pesquisa*

Após preparar o PEP para submissão, o aluno deve acessar o sistema através do menu “Mestrado” e “PEP”. Uma tabela com submissões anteriores do plano será visualizada. Caso o aluno ainda não tenha submetido, uma mensagem com um link aparecerá, conforme apresentado na figura 5.19. Para submeter o plano, basta clicar no link.



*Figura 5.20: Formulário para a submissão do PEP*

Clicando no link, o formulário para a submissão irá aparecer. O aluno deverá anexar o documento no formato PDF, e opcionalmente preencher uma observação que ele considere

pertinente para o observador (figura 5.20). Esta observação será enviada por email junto com o aviso, para o seu orientador e co-orientador (caso exista).

Ao receber email do sistema informando de que um dos seus orientandos submeteu o PEP, o orientador acessa o sistema e visualiza a listagem de PEPs enviados por seus alunos, conforme a figura 5.21. Ele pode visualizar o arquivo submetido pelo aluno clicando sobre o nome do arquivo. Para aprovar o PEP, basta clicar sobre o ícone de “Visto”. Para reprovar o PEP, o orientador deve clicar sobre o “X”.

The figure displays two views of the 'Planos de Ensino e Pesquisa' (PEPs) management interface. The left view shows a sidebar with navigation options like 'Calendário', 'Pesquisa de inscritos', 'Meus Orientandos', 'Mestrado', 'Doutorado', 'Alunos Especiais', and 'Produção'. The main content area shows a table of PEPs for 'Alunos de Mestrado e seus PEPs'. The table has columns for 'Aluno', 'Status', 'Documento', 'Obs', and 'Ações'. One entry for 'Carlos Kuhn' is shown with status 'Aguarda análise' and document 'pep.pdf'. The right view is a zoomed-in version of the table entry, showing the 'Ações' column with a green checkmark and a red 'X' icon.

Aluno	Status	Documento	Obs	Ações
Carlos Kuhn	Aguarda análise	pep.pdf	Lhe envio o meu plano de ensino e pesquisa, conforme os parâmetros confirmados em reunião	✓ ✗

Figura 5.21: Visualização pelo professor dos PEPs enviados pelos seus orientandos

The figure displays two views of the 'Recusa de proposta de PEP' (PEP rejection) form. The left view shows the main form with a sidebar and a large text area for 'Justificativa'. The right view is a zoomed-in version of the 'Justificativa' text area and the 'CONFIRMAR RECUSA' button. Below the form, there is an information box with a blue background and a white 'i' icon, containing text about the email notification and the student's ability to resubmit the PEP.

**Informação:** Após a confirmação da recusa deste PEP, o seu aluno orientado receberá um e-mail do sistema contendo a justificativa informada acima. A partir deste momento, o aluno poderá acessar o sistema novamente para efetuar correções sugeridas no PEP, e efetuar uma nova submissão.

Figura 5.22: Formulário para a reprovação de PEP

Em caso de reprovação do documento, o professor deve incluir uma justificativa que é enviada ao aluno junto com o informativo por email. Na figura 5.22 é possível verificar o formulário para a recusa de PEP.

Uma vez recusado, a listagem é atualizada e o professor aguarda uma nova submissão do documento. Conforme se vê na figura 5.23, as operações de recusa e aprovação não ficam mais disponíveis até a submissão de uma nova versão do documento.

The figure displays two side-by-side screenshots of the INF (Instituto de Informática UFRRS) web interface. Both screenshots show the 'Planos de Ensino e Pesquisa' (PEP) section for a student named Carlos Kuhn. The status is 'Recusado' (Rejected). A green banner at the top of the PEP section in both screenshots reads: 'O PEP do Carlos Kuhn foi recusado com sucesso'. Below this, a table lists the student's PEPs. The table has columns for 'Aluno', 'Status', 'Documento', 'Obs', and 'Ações'. The row for Carlos Kuhn shows 'Recusado' status, 'pep.pdf' document, and the observation: 'Lhe envio o meu plano de ensino e pesquisa, conforme os parâmetros confirmados em reunião'. The left screenshot shows a sidebar with navigation options like 'Calendário', 'Meus Orientandos', 'Mestrado', 'Doutorado', 'Alunos Especiais', and 'Produção'. The right screenshot shows a simplified navigation bar with icons for 'M', 'D', 'E', and 'P'.

Figura 5.23: PEP aprovado

Caso o orientador opte pela aprovação, e caso não possua co-orientador, o sistema confirmará a aprovação do PEP e enviará email informativo para o aluno. Caso haja, ambos os orientadores devem aprovar o plano. Enquanto o plano não for aprovado por ambos, ele permanecerá com o status “parcialmente aprovado”.

A secretaria também pode acompanhar o envio da documentação em tempo real através do sistema. Na figura 5.24, visualizamos a situação geral e a listagem de todos os alunos que devem enviar o documento, bem como a situação atual do seu envio. É possível efetuar o download do PEP clicando sobre o link “Download”, que aparece apenas após o PEP ser aprovado pelo orientador.

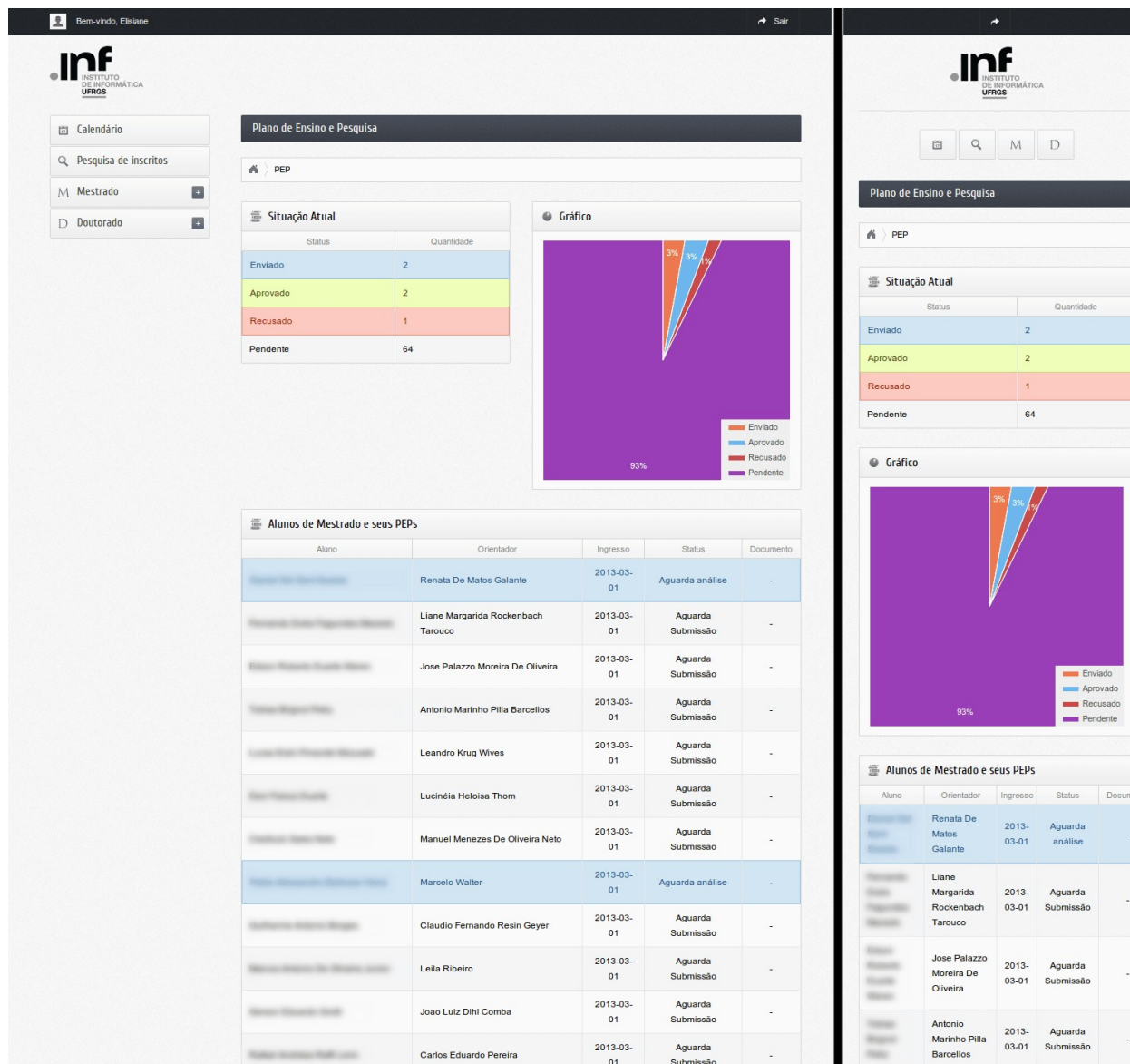


Figura 5.24: Visualização dos PEPs submetidos e pendentes pela secretaria

## 6 CONCLUSÃO

Este trabalho apresentou a concepção e o desenvolvimento de um sistema de administração para o Programa de Pós-Graduação em Computação - através da reengenharia do Sistema de Inscrições - visando o futuro da informatização do setor. Foi apresentado todo o processo de desenvolvimento de novas funcionalidades, desde a modelagem do processo até o teste e implantação da funcionalidade. Para tal, foi separado como exemplo o desenvolvimento de duas funcionalidades: submissão do plano de ensino e pesquisa (PEP) e seleção de candidatos ao doutorado.

Foram utilizados conceitos de modelagem de processos para a geração de um modelo de processo (na notação BPMN) para cada uma das funcionalidades, através do qual o desenvolvedor extrai histórias de usuário para guiar o desenvolvimento do software. Através da abordagem *Mobile First*, a interface foi projetada para suportar o uso do sistema por dispositivos móveis, abordagem permitida através da aplicação de técnicas de *Responsive Web Design*.

As funcionalidades que migraram do sistema de inscrições para o sistema de administração foram reescritas sobre uma arquitetura MVC, provida pelo framework FuelPHP. A utilização de testes automatizados extraídos diretamente das histórias de usuário geradas nos primeiros passos da modelagem do processo, em combinação com técnicas como *Behaviour-Driven Development* forneceram uma segurança adicional à qualidade final do software desenvolvido, que também passou a ter o seu código versionado de maneira distribuída, o que favorece o desenvolvimento compartilhado e garante maior segurança em caso de problemas com o repositório raiz.

Por fim, o sistema já está operacional para os processos de seleção de alunos de mestrado e doutorado, submissão de planos de ensino e pesquisa, submissão de propostas de TI (Trabalho Individual) e cargas do xml do lattes para docentes e alunos regulares do programa.

Algumas melhorias ainda se mostram necessárias, especialmente no que diz respeito ao desenvolvimento do sistema em paralelo entre dois ou mais desenvolvedores. O uso de técnicas ligadas a metodologias ágeis, como a utilização de quadros de kanban (através de ferramentas como Trello ou PivotalTracker), organização das histórias de usuários em iterações, melhor cobertura de código através de testes unitários (e eventualmente o uso de TDD) são assuntos que devem ser tratados futuramente. Também com este propósito se faz necessária a melhoria da documentação do banco de dados, mapeando o relacionamento entre as entidades que compõem o sistema.

## 7 REFERÊNCIAS

WESKE, M.. **Business Process Management: Concepts, Languages, Architectures**. Springer, 2012.

THOM, L. H.. **Gerenciamento de Processos de Negócio e Aplicabilidade na Saúde e na Robótica**. JAI - XXXI Jornadas de Atualização em Informática, 2012.

FOWLER, M.. **Use Cases and Stories**. Disponível em <http://www.martinfowler.com/bliki/UseCasesAndStories.html>. Acesso em dezembro, 2013.

GAMMA, E. et al. **Padrões de Projetos: Soluções Reutilizáveis**. Bookman, 2006.

HUMBLE, J.; FARLEY, D.. **Continuous Delivery**. Addison-Wesley, 2010.

NIELSEN, J.. **Mobile Usability, First Findings**. Disponível em <http://www.nngroup.com/articles/mobile-usability-first-findings/>. Acesso em dezembro, 2013.

MARCOTTE, E.. **Responsive Web Design**. A Book Apart, 2011.

W3C. **Media Queries: W3C Recommendation, 19 June 2012**. Disponível em <http://www.w3.org/TR/2012/REC-css3-mediaqueries-20120619/>. Acesso em dezembro, 2013.

WROBLEWSKI, L.. **Mobile First**. A Book Apart, 2011.

FUELPHP. **FuelPHP: A simple, flexible, community driven PHP 5.3+ framework**. Disponível em <http://fuelphp.com/>. Acesso em dezembro, 2013.

BEHAT. **Behat: A php framework for testing your business expectations**. Disponível em <http://behat.org/>. Acesso em dezembro, 2013.

GHERKIN. **Gherkin: A Business Readable, Domain Specific Language**. Disponível em <https://github.com/cucumber/cucumber/wiki/Gherkin>. Acesso em dezembro, 2013.

MINK. **Mink: Web acceptance testing**. Disponível em <http://mink.behat.org/>. Acesso em dezembro, 2013.

SELENIUM. **SELENIUM: a suite of tools to automate web browsers across many platforms**. Disponível em <http://docs.seleniumhq.org/>. Acesso em dezembro, 2013.

CHACON, S.. **Pro Git**. Apress, 2009. (Também disponível em <http://git-scm.com/book>. Acesso em dezembro, 2013)

GIT. **Git: A Free and Open Source Distributed Version Control System**. Disponível em <http://git-scm.com/>. Acesso em dezembro, 2013.

BITBUCKET. **Bitbucket: Free source code hosting for Git and Mercurial**. Disponível em <https://bitbucket.org/>. Acesso em dezembro, 2013.

DRIESSEN, V.. **GitFlow: A successful Git branching model**. Disponível em <http://nvie.com/posts/a-successful-git-branching-model/>. Acesso em dezembro, 2013.

COHN, M.. **User Stories Applied**. Addison-Wesley, 2004.

WAKE, W. C.. **INVEST in Good Stories, and SMART Tasks**. Disponível em <http://www.xp123.com>. Acesso em dezembro, 2013.

FOX, A., PATTERSON, D.. **Engineering Long-Lasting Software: An Agile Approach Using SaaS and Cloud Computing**. Strawberry Canyon LLC, 2012.