

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE LICENCIATURA EM INFORMÁTICA
CURSO DE ESPECIALIZAÇÃO EM DESENVOLVIMENTO DE SISTEMAS PARA
INTERNET E DISPOSITIVOS MÓVEIS

GUSTAVO DALLASTA DUTRA

Ferramenta para Gestão de Chamadas Públicas

MONOGRAFIA DE ESPECIALIZAÇÃO

FRANCISCO BELTRÃO

2014

GUSTAVO DALLASTA DUTRA

Ferramenta para Gestão de Chamadas Públicas

Monografia apresentada à Coordenação de Licenciatura em Informática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de "Especialista em Desenvolvimento de Sistemas para Internet e Dispositivos Móveis".

Orientador: Prof. Msc. Wellton Costa

FRANCISCO BELTRÃO

2014



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus de Francisco Beltrão
Diretoria de Pesquisa e Pós-Graduação
Curso de Especialização em Desenvolvimento de Sistemas para
Internet e Dispositivos Móveis.

TERMO DE APROVAÇÃO

Dia 16 do mês de outubro de 2014 às 17:00 horas, na sala Q-204 do Câmpus Francisco Beltrão, realizou-se a apresentação pública da monografia pelo estudante Gustavo Dallasta Dutra, intitulada “Ferramenta para Gestão de Chamadas Públicas”. Finalizada a apresentação e arguição, a Banca Examinadora declarou **aprovada** a monografia do estudante, requisito parcial para obtenção do título de Especialista em Desenvolvimento de Sistemas para Internet e Dispositivos Móveis.

Professor Wellton Costa – UTFPR
(ORIENTADOR)

Professor Renato Hallal - UTFPR
(Convidado)

Professor Marcos Mincov Tenório - UTFPR
(Convidado)

Professor Dr. Ademir Roberto Freddo – UTFPR
(COORDENAÇÃO)

Dedico esta monografia à minha família pelo apoio e compreensão para alcançar o objetivo.

AGRADECIMENTOS

À Deus e à minha família pela força e apoio motivacional durante este período. Agradeço aos colegas de trabalho, pela compreensão e apoio durante esta jornada. Aos professores da Universidade Tecnológica Federal do Paraná, pela maneira exemplar com que conduziram o curso. Ao prof. Msc. Wellton Costa, pela atenção disponibilizada na orientação deste trabalho. E também a todas as pessoas que, de alguma forma, contribuíram no êxito desse trabalho.

RESUMO

Atualmente, o conhecimento é a base para a formação não somente de indivíduos capazes, mas também de instituições competitivas e aptas a resolver os problemas da sociedade como um todo. Devido aos desafios impostos, a universidade, através da geração do conhecimento, se transforma num vetor de soluções às comunidades o qual estão inseridas através de seus projetos de pesquisa e extensão. Devido à crescente demanda por projetos de pesquisa e programas de extensão, faz-se necessário o devido gerenciamento do ciclo de vida dos projetos acadêmicos que vão desde a etapa de submissão de trabalhos até a análise estatística da produção acadêmica. O objetivo é reduzir a dependência manual e o retrabalho em gerenciar todo esse ciclo de vida propondo uma solução computacional que possa automatizar o ciclo de vida das chamadas de projetos e editais públicos.

Palavras-chave: Chamadas públicas. Editais. Internet. Sistema de Gestão.

ABSTRACT

Currently , knowledge is the basis for the formation of not only individuals capable , but also competitive and able to solve the problems of society as a whole institutions . Due to the challenges , the university , through the generation of knowledge , becomes a vector of solutions to communities which are inserted through their research projects and extension. Due to the increasing demand for research projects and outreach programs , it is necessary because the management of the lifecycle of academic projects ranging from the stage of submission of papers to the statistical analysis of academic production . The goal is to reduce dependence on manual manage this whole lifecycle proposing a computational solution that can automate the lifecycle of calls for projects and public notices what has been done manually.

Keywords: Public calls. Notices. Internet. Management System.

LISTA DE FIGURAS

FIGURA 1 :	Representação do ciclo de vida de uma chamada pública para editais de projetos	17
FIGURA 2 :	Estrutura do MVC	19
FIGURA 3 :	Representação dos principais componentes gráficos do Primefaces	20
FIGURA 4 :	Representação do model Pessoa - exemplo	21
FIGURA 5 :	Consulta recuperando todos os registros da tabela Pessoa - exemplo	22
FIGURA 6 :	Visão geral da estrutura da aplicação	24
FIGURA 7 :	Diagrama de classes model das entidades que envolvem o cadastro de Projeto..	27
FIGURA 8 :	Diagrama de seqüência referente ao processo de consulta de Dados Pessoais	29
FIGURA 9 :	Diagrama de seqüência referente ao processo de cadastro de Dados Pessoais	30
FIGURA 10 :	Diagrama de seqüência referente ao processo de alteração de Dados Pessoais ..	31
FIGURA 11 :	Diagrama de seqüência referente ao processo de exclusão de Dados Pessoais	32
FIGURA 12 :	Interface de cadastro de dados pessoais	33
FIGURA 13 :	Interface de atribuição de perfil de acesso à aplicação	34
FIGURA 14 :	Interface de cadastro e edição de projetos	34

LISTA DE SIGLAS

AJAX	<i>Asynchronous Javascript And Xml</i>
API	<i>Application Programming Interface</i>
DAO	<i>Data Access Object</i>
EE	<i>Enterprise Edition</i>
HTML	<i>Hypertext Markup Language</i>
IDE	<i>Integrated Development Environment</i>
IFES	<i>Instituições Federais de Ensino Superior</i>
J2EE	<i>Java Enterprise Edition</i>
JVM	<i>Java Virtual Machine</i>
JDBC	<i>Java Database Connectivity</i>
JSP	<i>Java Server Pages</i>
JSF	<i>Java Server Faces</i>
MVC	<i>Model View Controller</i>
MVCC	<i>Multi Version Control Concurrency</i>
ODBC	<i>Open Database Connectivity</i>
ORM	<i>Object Relational Mapping</i>
SQL	<i>Structured Query Language</i>
SGBD	<i>Sistema de Gestão de Banco de Dados</i>
UI	<i>User Interface</i>
XHTML	<i>EXtensible HyperText Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO	12
1.1 PROBLEMA DA PESQUISA	13
1.2 HIPÓTESE	14
1.3 OBJETIVOS	14
1.3.1 OBJETIVO GERAL	14
1.3.2 OBJETIVOS ESPECÍFICOS	14
1.4 JUSTIFICATIVA	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 CHAMADA PÚBLICA	16
2.2 JAVA.....	17
2.3 MVC.....	18
2.4 JSF	19
2.5 PRIMEFACES	19
2.6 JBOSS	20
2.7 HIBERNATE	21
2.8 POSTGRESQL	22
2.9 ECLIPSE	23
3 METODOLOGIA	24
3.1 VISÃO GERAL	25
4 RESULTADOS	28

5 CONCLUSÃO E TRABALHOS FUTUROS	36
REFERÊNCIAS BIBLIOGRÁFICAS	37

1 INTRODUÇÃO

O mundo globalizado de hoje, necessita que as pessoas estejam cada vez mais capacitadas, requerendo acesso aos meios acadêmicos para fins de instrução e competir melhor no mercado de trabalho. Diante dessa necessidade, referente a atender essa demanda crescente por mão de obra qualificada, houve um incremento por parte do governo federal em ofertar um maior número de instituições públicas de ensino superior e, conseqüentemente, de um maior número de vagas acadêmicas nas IFES.

Por conta de crescente produção acadêmica, existem instituições que controlam de forma manual, todo o ciclo de vida de publicação, gerenciamento e concorrência dos editais dos programas de pesquisa e extensão internos e externos à universidade. Diante disso, faz-se necessária a utilização de uma solução automatizada que gerencie e centralize o ciclo de vida da publicação de editais e chamadas públicas dos programas de ensino, pesquisa e extensão.

Desta forma, propõe-se um ambiente para promoção da interação entre pesquisadores das instituições acadêmicas como forma de agente promotor de inovação e desenvolvimento tecnológico. Esse ambiente é o projeto do qual propõe esse trabalho, de reunir numa solução informatizada e de fácil acesso, toda a gestão do ciclo de vida de editais de chamadas públicas, desde a sua disponibilização passando pela etapa de submissão de propostas, avaliação, até chegar na etapa final de deferimento da proposta e indicadores de produção.

1.1 PROBLEMA DE PESQUISA

Dentro do meio acadêmico, é natural que o corpo docente ofereça, através dos programas de pesquisa e extensão, editais de chamadas públicas de projetos, estes abertos a determinados segmentos da comunidade acadêmica ou do público em geral. Existem também chamadas públicas feitas inclusive em parcerias com outros órgãos governamentais e empresas que envolvem, inclusive, recursos financeiros de forma a patrocinar o andamento de certos projetos. Portanto, faz-se necessária a elaboração de uma ferramenta que faça não só a publicação e gerenciamento dos editais de pesquisa, ensino e extensão, mas também do controle financeiro que certas chamadas fazem uso. Devido ao envolvimento de docentes na avaliação dos questionários de pontuação de cada edital, é necessária a gestão de formulários de avaliação e o envio do mesmo aos docentes avaliadores que podem ser, inclusive, de outras instituições.

Existem atualmente, ferramentas informatizadas que fazem a gestão de projetos de pesquisa de forma genérica, ou seja, fazem a publicação e avaliação dos editais de pesquisa, permitindo o cadastro de avaliadores para pontuar os editais cadastrados, com uma gestão simples.

1.2 HIPÓTESE

Através de demanda levantada por uma IFES da região, a solução computacional a ser aplicada deve gerenciar todo o processo de avaliação e submissão de projetos assim como gerenciar os projetos aprovados mantendo uma base de dados para possibilitar a busca e rápida recuperação dos mesmos assim como prover uma análise estatística de dados gerenciais através de gráficos e tabelas com índices de produção.

1.3 OBJETIVOS

1.3.1 OBJETIVO GERAL

O referido trabalho tem por objetivo geral propor o desenvolvimento de um sistema computacional que gerencie o ciclo de vida de publicação e controle da oferta de editais de pesquisa e extensão dos programas de graduação e pós-graduação de instituições públicas ou privadas de ensino superior.

1.3.2 OBJETIVOS ESPECÍFICOS

Como objetivos específicos, pretende-se contemplar o desenvolvimento inicial da ferramenta, contemplando o item 1 - elaboração - do ciclo de vida das chamadas públicas, com as seguintes funcionalidades:

- manter o cadastro das tabelas básicas e de apoio à aplicação;
- controlar o cadastro de projetos e seus integrantes;
- manter um cadastro de docentes e/ou pesquisadores para atuarem como avaliadores de cada submissão de edital;
- efetuar o cadastro de usuários, perfis de acesso e logs do sistema;
- utilizar-se de ferramentas livres durante o processo de desenvolvimento abrangendo a plataforma *web*;

1.4 JUSTIFICATIVA

Nos dias atuais, vive-se uma constante dentro do mundo globalizado onde a peça chave para o sucesso passa pelo conhecimento. A geração de conhecimento é crucial desde para manter as instituições competitivas quanto para resolver os problemas da sociedade. Para que o conhecimento ocorra, deve-se haver um estímulo constante das instituições acadêmicas através da pesquisa, das atividades de extensão para aplicar a pesquisa e através do ensino para formar a mão de obra de futuros pesquisadores.

As instituições acadêmicas, através do seu corpo docente, estimulam seu corpo discente e sua comunidade a inserir-se nos diversos programas de pesquisa e extensão a fim de promover a troca de conhecimento em favor de resolver os mais variados problemas que ocorrem nas suas comunidades. Isto se dá através das chamadas de editais e bolsas de pesquisa, onde as partes interessadas se juntam aos projetos a fim de gerar conhecimento necessário para promover a solução de problemas.

Como a razão de ser de uma universidade dá-se também através da pesquisa, é natural que com o passar do tempo ocorra um aumento no volume de programas e projetos, fazendo necessário um gerenciamento eficaz de tudo o que é ofertado a favor da promoção do conhecimento.

Tendo em vista a necessidade de controlar os requisitos de negócio levantados na instituição de ensino pesquisada, este projeto visa a implementação de uma ferramenta informatizada que contemple todo o controle do ciclo de vida de chamadas públicas.

O propósito da ferramenta visa principalmente substituir todo o controle manual ou feito através de planilhas de cálculo e arquivos de documentos de texto para centralizar os dados em uma única plataforma que seja acessível por diferentes níveis de usuário ou de diferentes setores e departamentos. Para tanto, faz-se necessário o uso de tecnologias atuais e robustas no seu processo de desenvolvimento e execução da ferramenta depois de concluída.

Assim, o resultado proposto é um produto desenvolvido com qualidade necessária para coroar os conhecimentos aprendidos durante o curso, uma vez que ele abrange conceitos abordados nas disciplinas, como linguagem Java, desenvolvimento orientado a objeto baseado em camadas, framework Web Java Server Faces e banco de dados Postgres.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 CHAMADA PÚBLICA

O conceito de chamada pública (CAMARA, 2006) tem por finalidade assegurar o princípio da ampla publicidade dos atos da administração e deve sempre visar ao interesse público. Também define que a entidade executora é responsável pela chamada pública, devendo conter informações suficientes para que os pesquisadores interessados formulem corretamente os projetos de pesquisa e extensão, como o detalhamento da proposta de adesão, equipe a participar, cronograma de desenvolvimento, locais onde se localizará a pesquisa e seus custos envolvidos.

A entidade executora também deve publicar as demandas no desenvolvimento dos projetos de ensino, pesquisa e extensão em um jornal de circulação local, página na internet ou mural em local público de ampla circulação.

Uma chamada pública padrão obedece ao seguinte ciclo de vida:

- elaboração e publicação dos editais de chamadas públicas
- submissão das propostas de cada edital publicado
- definição dos avaliadores internos e/ou externos
- avaliação interna e/ou externa
- formalização/homologação dos projetos
- acompanhamento da execução dos projetos
- recebimento e análise da prestação de contas

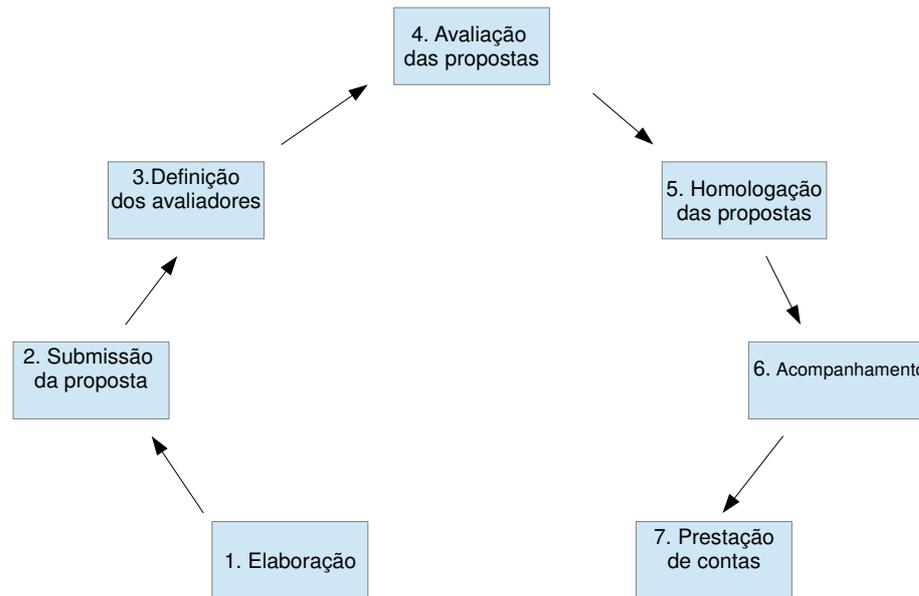


Figura 1 : Representação do ciclo de vida de uma chamada pública para editais de projetos.

Fonte: arquivo do autor

2.2 JAVA

A plataforma Java é descrita como uma linguagem de programação orientada a objetos, simplificada, segura, com uma extensa API e um ambiente de execução multi-plataforma (SILVEIRA, 2012). Criada pela Sun, teve seu lançamento público em 1995, porém vinha sendo mantida desde 1991 cujo mercado inicial era o desenvolvimento de sistemas embutidos, como *set-top boxes* para televisores. Atualmente mantida pela Oracle, o Java foi concebido como uma solução para o desenvolvimento de aplicativos portáteis.

A linguagem Java é uma evolução das linguagens C e C++ que até então eram as mais utilizadas na época de sua concepção. De acordo com Deitel (2005, p 6) havia uma necessidade na época de adicionar conteúdo dinâmico às páginas web, como interatividade e animações. Por acaso, a *World Wide Web* popularizou-se em 1993 e a equipe da Sun na época viu o imediato potencial de adicionar esse tipo de conteúdo às páginas na *Web*, originando assim o conceito de *applets* e dando nova vida ao projeto.

Atualmente a plataforma Java consiste de um pacote de desenvolvimento denominado JDK - *Java Development Kit*. Uma suíte de desenvolvimento que contempla as funcionalidades básicas da linguagem como strings, listas, classes, dentre inúmeras funcionalidades. O pacote JDK hoje, pode ser instalado em três diferentes versões: Java SE ou *Standard Edition*, que

contempla as funcionalidades básicas, Java EE ou *Enterprise Edition*, que adiciona as funcionalidades da versão SE mais funcionalidades voltadas a web e a versão Java ME ou *Micro Edition*, que seria uma versão resumida da Java SE voltada dispositivos móveis e softwares embutidos.

2.3 MVC

O modelo MVC - *Model View Controller* (SILVEIRA, 2012, p.151) é um padrão arquitetural que visa separar a camada visual da aplicação do domínio e do negócio. Uma característica desse modelo é garantir que uma camada não acesse outra que não seja seu vizinho direto. Este modelo surgiu de uma época em que código o HTML era misturado com a linguagem de consulta SQL e a lógica de negócios, tornando difícil a manutenção dos códigos.

O projeto proposto neste trabalho está baseado nessa arquitetura, dividida nas 3 camadas propostas pelo modelo:

- Objetos do modelo de dados (*Model*): mantém, encapsulam e abstraem os dados, não contendo métodos com lógica para a aplicação funcionar. Mantém as implementações de métodos para serializar os dados definidos nos tipos propostos para cada coluna no banco de dados. Aqui não entram eventos de interface, como ações de clique de botão, por exemplo, que devem ser implementados no *controller*.
- Objetos de visão (*View*): mostra as informações do modelo para o usuário. É nesta camada que são capturadas as interações de interface, como cliques e pressionamento de teclas, e os converte em “ações” que são passados para o *controller* processar.
- Objetos de controle (*Controller*): implementa as funções e métodos da lógica da aplicação.

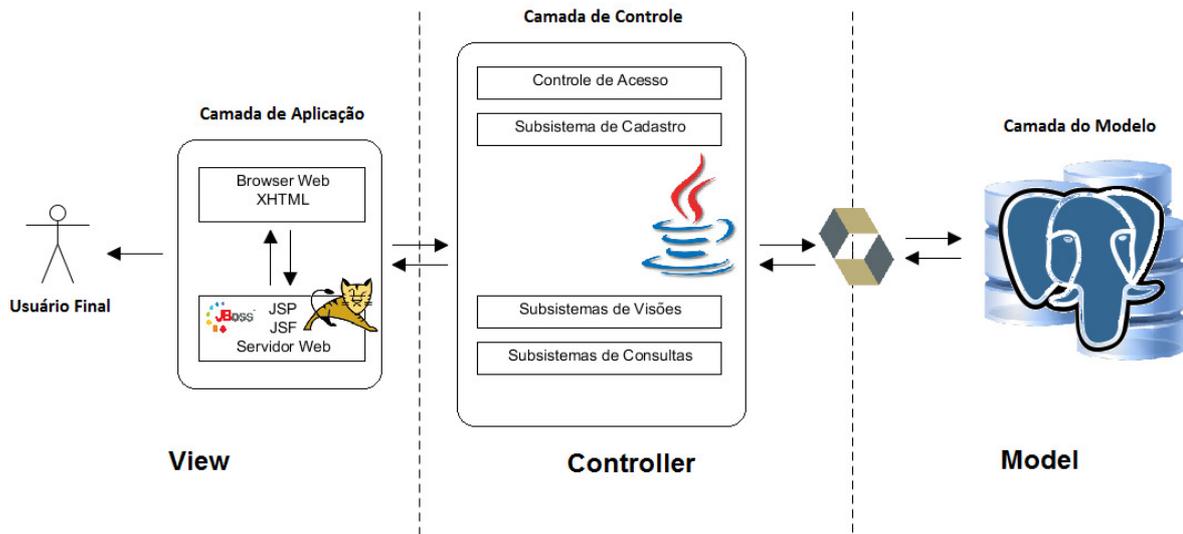


Figura2 : Estrutura do MVC

Fonte: arquivo do autor

2.4 JSF

É uma especificação Java que teve uma forte aceitação no mercado devido à sua facilidade de implementação e farta documentação. Este framework (SILVEIRA, 2012, p 154), trabalha com uma hierarquia de componentes manipulando uma representação de cada elemento como um objeto organizando todos eles num formato de árvore de componentes. Cada elemento da árvore guarda o estado atual dos componentes da tela e, a cada evento disparado, é feita uma requisição para o JSF, que se encarrega de, logo no início da requisição, atualizar as informações nesta árvore. Isso torna possível no desenvolvimento de sistemas para *Web*, que o valor de um campo seja automaticamente guardado em um atributo de um *bean* já convertido para o formato correto, por exemplo. Ou o clique de um botão na tela dispare um método escolhido pelo desenvolvedor diretamente ao *bean*.

2.5 PRIMEFACES

Primefaces é uma suíte de componentes gráficos baseado no JSF, com código aberto e sob licença Apache v2. É criado e mantido por uma organização denominada Prime Teknoloji, baseada na Turquia e presta consultoria a empresas nas áreas de Java EE e Agile (ÇIVICI 2014, p12).

Sua utilização consiste no download e configuração do arquivo da biblioteca num projeto Java Web, mais especificamente na pasta */lib* do projeto Eclipse.

As principais características dessa suíte de componentes são (Figura 3):

- Rico conjunto de componentes (HTML Editor , de diálogo , de conclusão automática de ações, gráficos e muito mais) .
- Ajax embutido com base no padrão de JSF 2.0.
- Leve, composto de biblioteca única, sem dependências.
- Suporte nativo a Ajax Push através de *websockets* .
- Possui um kit UI Mobile para criar aplicações web móveis para dispositivos portáteis.
- A vasta documentação e suporte .
- Grande comunidade de desenvolvedores desde pequenos usuários a clientes corporativos.



Figura 3 : Representação dos principais componentes gráficos do Primefaces.

Fonte: PRIMEFACES, 2014.

2.6 JBOSS

JBoss é um servidor de aplicações Java, baseado na plataforma J2EE. Possui ambiente multiplataforma onde o único pré-requisito necessário é o sistema operacional oferecer suporte a JVM - *Java Virtual Machine* na versão 6 ou superior. O projeto é custeado por uma rede mundial de colaboradores atualmente mantida pela empresa RedHat, o qual fornece um pacote de bibliotecas básicas para implantar e executar aplicações web baseadas nas especificações da Java EE, possuindo um servidor de aplicações Tomcat embutido, do qual foi baseado.

Uma das principais características do servidor JBoss é sua robustez em executar aplicações de médio e grande porte, a grande possibilidade de configurações de execução dos projetos web implantados e configurações de segurança das aplicações instaladas. Este servidor de aplicação possui versões gratuitas, mantidas pela comunidade de desenvolvedores e pela equipe da RedHat e versões licenciadas mantidas pelo time de desenvolvimento da RedHat.

2.7 HIBERNATE

O Hibernate é um framework de Mapeamento Objeto Relacional das tabelas do banco de dados, a fim de reduzir a complexidade envolvida no desenvolvimento de aplicações que desejam trabalhar com um banco de dados relacional. As tabelas do banco de dados (SILVEIRA 2012, p.159) são representadas por classes Java e as operações de persistência e recuperação de dados são efetuados pelo próprio framework.

Um exemplo prático seria a representação de uma entidade Pessoa em uma aplicação Java mapeada com base na tabela pessoa de um banco de dados. A representação do *model* Pessoa é representada na figura 4 abaixo.

```
public class Pessoa {  
  
    private int id;  
  
    private String nome;  
  
    private long cpf;  
  
    private int idade;  
  
    //Construtor  
    public Pessoa() {}  
  
    //Métodos getters e setters
```

Figura 4 : Representação do model Pessoa - exemplo

Fonte: arquivo do autor

As consultas utilizando o Hibernate podem ser feitas em duas formas principais: utilizando uma query para consulta através da linguagem HQL - Hibernate Query Language, ou via Criteria *query* API que seria uma forma de consulta baseada e tempo de compilação. Como a aplicação proposta neste trabalho utiliza a primeira abordagem para consultas, através de *queries* em HQL, é mostrado um exemplo prático na figura 5, efetuando a recuperação de

todos os registros da tabela pessoa, mapeada pela classe Pessoa e cujos registros recuperados estão envolvidos numa lista.

```

public class ConexaoHiberante {

    public static void main(String[] args){

        SessionFactory sessionFactory =
            new Configuration().configure().buildSessionFactory();

        Session session = sessionFactory.openSession();

        String hql = "from Pessoa";

        Query query = session.createQuery( hql );

        List peessoasList = query.list();

    }

}

```

Figura 5 : Consulta recuperando todos os registros da tabela Pessoa - exemplo

Fonte: arquivo do autor

2.8 POSTGRESQL

O PostgreSQL é um sistema de gerenciamento de banco de dados objeto relacional de código fonte aberto desenvolvido a partir do banco de dados Ingres pela Universidade da Califórnia, em Berkeley, EUA. Por ser rico em recursos, é um SGBD cada vez mais utilizado tanto no meio acadêmico como empresarial (SILBERCHATZ, 2006, p.653) e dentre seus principais recursos, podemos citar:

- utilização do padrão SQL92 e SQL1999 oferecendo recursos como consultas complexas, chaves estrangeiras, *triggers*, *views*, integridade transacional e controle de concorrência de múltiplas versões;
- suporte a multiusuário e multiplataforma, possibilidade de *tunning* de configurações para otimizar desempenho, suporte a variáveis similares a *shells* de comando Unix, interfaces nativas para ODBC e JDBC com vínculos para várias linguagens de programação, como C++, PHP, Perl, Python;

- suporte a integridade referencial garantindo a solidez dos dados no banco;
- Controle de Concorrência Multi-Versão (MVCC) e bloqueio de transações em duas fases, onde é feito o gerenciamento das permissões de leitura e escrita do banco evitando *deadlocks* ocasionados por outros usuários que estão inserindo dados no mesmo instante.
- arquitetura baseada em ambiente cliente/servidor, onde o Postgres segue o modelo de processo por transação controlado por um processo coordenador central, chamado *postmaster*, responsável por inicializar e encerrar o servidor e também por tratar de solicitações de conexão de novos clientes.

2.9 ECLIPSE

O Eclipse é uma IDE, ou seja um ambiente integrado de desenvolvimento na linguagem de programação Java suportando também outras linguagens, como C++, PHP e Python. Esta ferramenta (IBM 2014), possui código fonte aberto cuja comunidade de desenvolvedores concentra-se em manter esta plataforma de forma livre e aberta. A ferramenta foi desenvolvida originalmente pela IBM em novembro de 2001 sendo repassada a uma organização se fins lucrativos e independente chamada Eclipse Foundation.

Possui uma grande variedade de plug-ins suportados e hospedados pela Eclipse Foundation encontrados no Web site do Eclipse ou na ferramenta Eclipse Marketplace disponibilizada pela própria IDE.

3 METODOLOGIA

A figura 6 mostra a estrutura da aplicação proposta, resultado da análise dos objetivos citados no primeiro capítulo abrangendo os requisitos propostos.

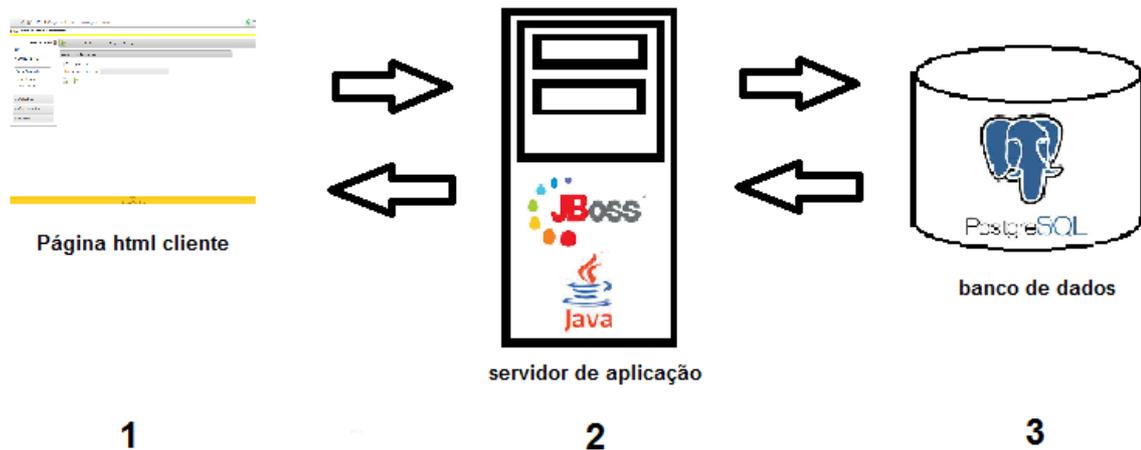


Figura 6: Visão geral da estrutura da aplicação

Fonte: arquivo do autor

A estrutura de funcionamento da aplicação, como mostra a figura 6, compreende o seguinte:

- página html do cliente (1) que faz as requisições do tipo *HTTP Request* ao servidor de aplicação JBoss (2);
- o servidor de aplicação JBoss (2), processa a requisição do cliente web (1). Como a maioria das requisições dependem de conexão ativa ao banco de dados, chamadas via Connector JDBC ao SGBD Postgres (3) são realizadas a todo instante através de um pool de conexões previamente configurado no servidor de aplicação;
- o SGBD Postgres (3) responde as chamadas da aplicação retornando ao servidor de aplicação JBoss (2) a requisição da aplicação;
- o servidor de aplicação (2) retorna ao cliente web o processamento final da requisição via *HTTP Response*.

O esquema de funcionamento é o padrão em qualquer aplicação cliente-servidor. Procurou-se manter esse esquema por questões de facilidade na implementação do projeto.

3.1 VISÃO GERAL

Para o desenvolvimento da aplicação foram utilizadas os seguintes recursos:

- Linguagem de programação Java versão 6u31;
- IDE de desenvolvimento Eclipse Kepler versão 3.4.1
- Framework de persistência de dados Hibernate versão 3.4
- Framework web JSF versão 2.0
- Suíte de componentes gráficos Primefaces versão 3.2
- Servidor de aplicação JBoss versão 6 Community
- Banco de dados Postgres versão 9.1

O acesso à aplicação é feito através de controle por usuário, onde cada usuário pode estar inserido em 3 categorias (*roles*) diferentes de privilégios de acesso. Isto é necessário devido ao controle de quem pode publicar os editais para posterior submissão de propostas por parte dos pesquisadores. A aplicação contempla três níveis de privilégios, a saber:

- administrador : perfil de acesso do tipo "super-usuário" o qual tem acesso irrestrito à todas as páginas e funcionalidades do sistema. Apenas usuários deste perfil de acesso podem publicar editais de chamadas públicas, definir os avaliadores de cada edital ou projeto e ratificar os resultados das avaliações encaminhadas pelos avaliadores. Também neste perfil são feitos os gerenciamentos de usuários e perfis de acesso ao sistema;
- pesquisador: perfil de acesso restrito onde é possível submeter propostas aos editais de chamadas públicas cadastrados pelo nível superior (administrador) bem como consultar às informações pertinentes às propostas enviadas. É neste perfil que os pesquisadores poderão inserir outros integrantes da equipe de projeto da proposta ao edital submetido;
- avaliador: perfil de acesso restrito onde são feitas as avaliações das propostas encaminhadas pelos pesquisadores.

A aplicação foi estruturada na forma de pacotes de classes no Eclipse, separando cada pacote de acordo com o nível de acoplamento do sistema, baseado no conceito de ORM - *Object Relational Mapping* ou Mapeamento Objeto Relacional (SILVEIRA, 2012. p. 159). O conceito de ORM tem o objetivo de fazer uma ligação entre o paradigma entidade relacional e o orientado a objetos reduzindo a distância entre essas duas abordagens.

A fim de atender este conceito, a estrutura do projeto ficou representada nos seguintes pacotes de classes:

- *beans*: contem as classes de manipulação e controle das interfaces (telas) do sistema, agrupando os controles de regras de negócio;
- *converters*: contém os conversores dos objetos de algumas classes *beans* do projeto. Por conta da necessidade de certos componentes da interface gráfica disponibilizados pelo Framework Primefaces, faz-se necessário criar conversores de tipo para trabalhar em conjunto com os *validators*.
- *daos*: concentra as classes de manipulação e consulta dos dados representados nos modelos de cada entidade do banco de dados. Entenda-se uma entidade como representante de uma tabela do banco de dados;
- *factories*: contem a classe que distribui e encerra as conexões com o banco de dados;
- *filters*: contem a classe que controla as conexões ativas e as transações de persistência de dados.
- *models*: contem as classes que mapeiam as tabelas do banco de dados;
- *utils*: contem classes auxiliares ao funcionamento das chamadas de interface (*beans*) concentrando componentes comuns a elas.
- *validators*: contém classes que desempenham o papel de validar os dados recebidos pelos componentes gráficos das interfaces do sistema.

O projeto contempla nesta primeira etapa de desenvolvimento, os cadastros básicos para alimentar os dados das tabelas padrão do sistema, como área de concentração, linha de pesquisa, pessoas e projetos. Numa segunda etapa de desenvolvimento, será implementada as funcionalidades de submissão de editais pelos pesquisadores e vínculo dos avaliadores das propostas submetidas. Mais a frente numa terceira etapa de projeto é que serão executadas as interfaces de indicadores de produção científica de cada pesquisador, assim como interfaces para a prestação de contas dos pesquisadores que tiveram seus projetos aprovados pela banca de avaliadores. Apesar da aplicação não oferecer todas as funcionalidades nesta etapa inicial

de desenvolvimento dos cadastros básicos, a aplicação em si teve sua estrutura lógica projetada para contemplar as necessidades futuras de implementação.

A título informativo, na figura 7 é mostrado o diagrama das classes *model* que contemplam a funcionalidade de cadastro de projeto e integrantes do projeto, considerados importantes no âmbito do escopo de projeto.

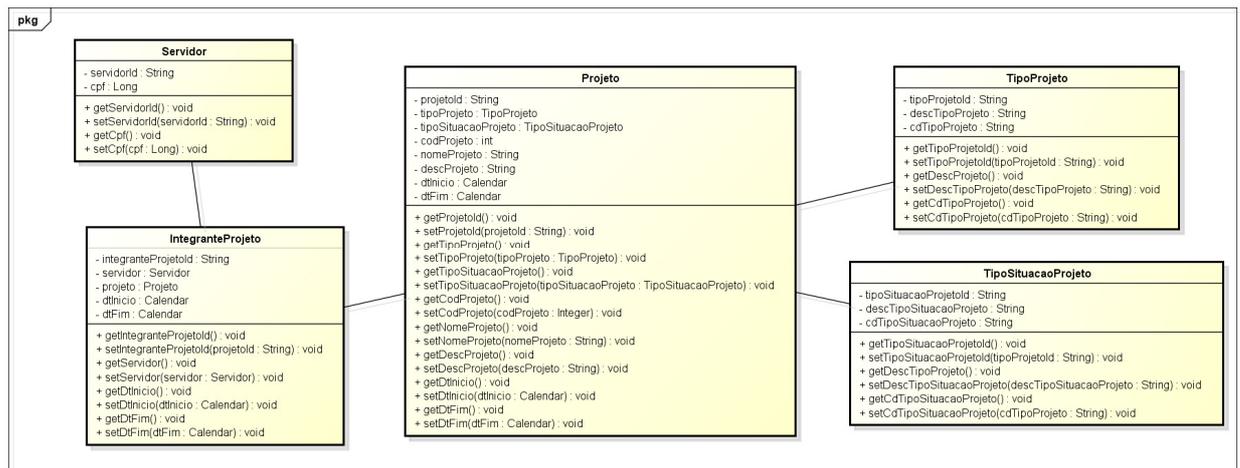


Figura 7 : Diagrama de classes model das entidades que envolvem o cadastro de Projeto

Fonte: arquivo do autor

Importante ressaltar que em todas as classes, os identificadores de chave primária são compostos por Strings e não por inteiros como é a abordagem tradicional. No momento de inserir um novo registro no banco de dados, é chamada uma função que gera uma chave primária do tipo *varchar* composta por 36 caracteres. Tal ação previne a existência de chaves compostas na modelagem da base de dados da aplicação.

4 RESULTADOS

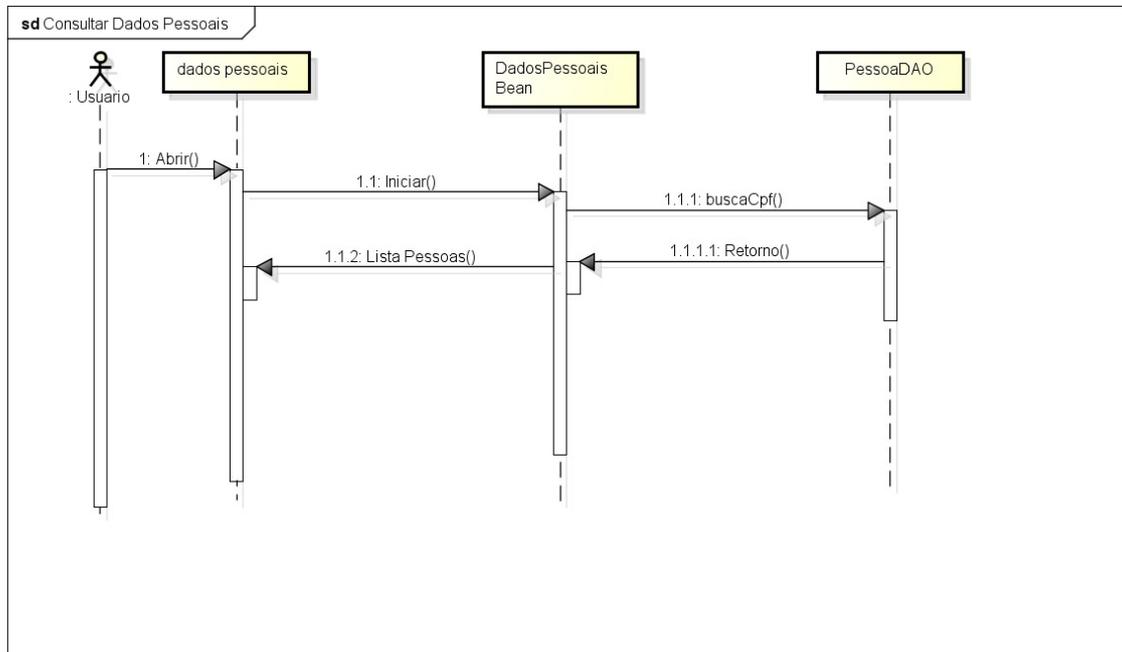
A aplicação implementa, nesta primeira fase, as seguintes funcionalidades:

- cadastro de tipos e vínculos genéricos;
- cadastro de dados pessoais;
- tipo de vínculo das pessoas cadastradas;
- cadastro de áreas de concentração , linhas de pesquisa e o vínculo entre ambas;
- vínculo dos usuários da aplicação: definição das permissões de acesso;
- cadastro de projeto: necessário para posteriormente vincular os editais de chamadas a serem publicados.

O cadastro de dados pessoais e a configuração do vínculo da pessoa com a instituição é realizado através da interface de dados pessoais, onde além do cadastro propriamente dito, é realizado qual vínculo a mesma terá na instituição.

Toda pessoa cadastrada também deverá estar vinculada a um campus de lotação a fim de realizar associações do registro de lotação da pessoa com permissões de acesso a funcionalidades liberadas especificamente a um determinado local. Por exemplo: um docente cadastrado no campus Francisco Beltrão que estiver com o vínculo de pesquisador, não poderá concorrer a editais de pesquisa disponibilizados apenas ao campus Curitiba.

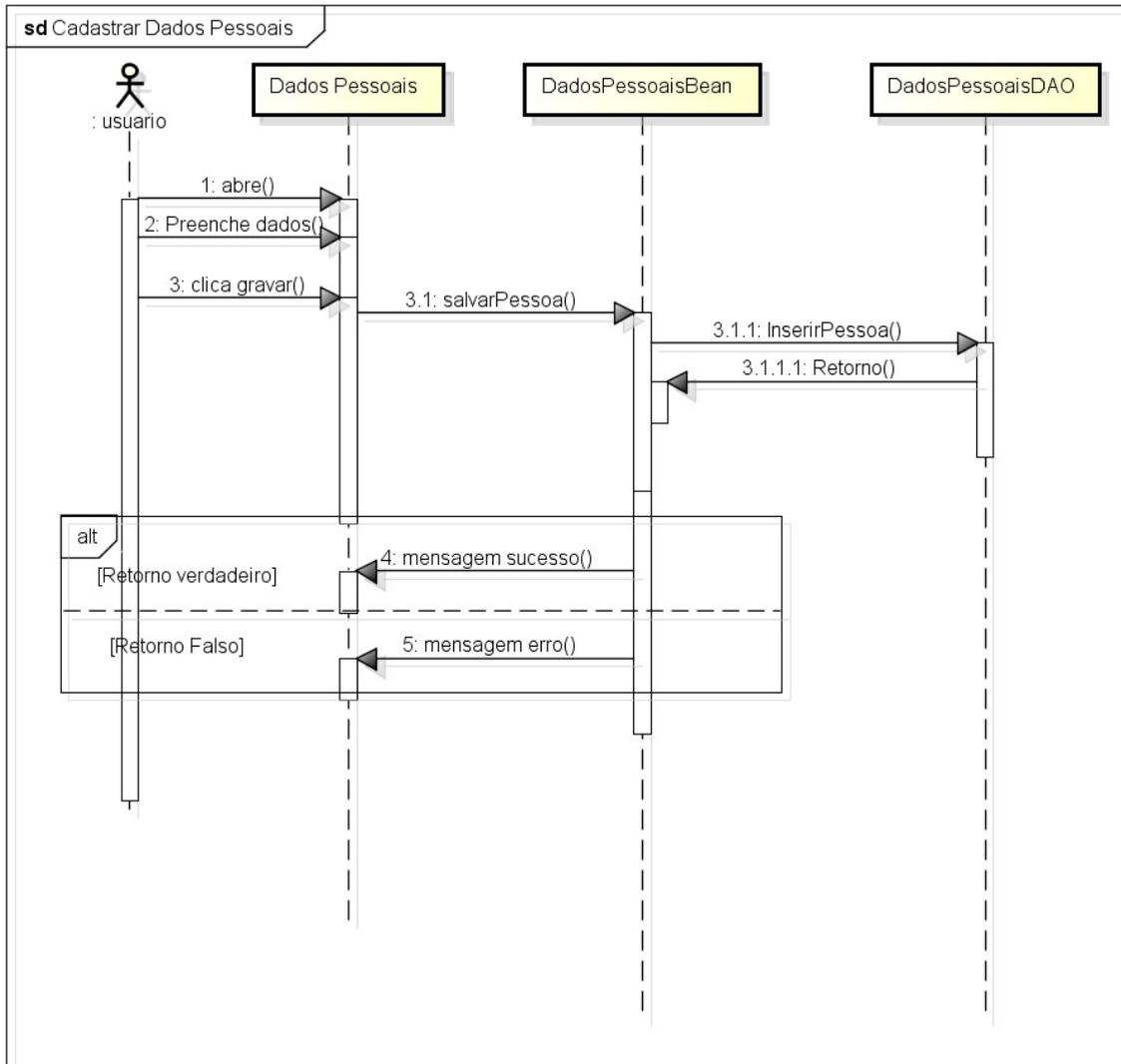
Haverá casos em que um docente deverá ter acesso a mais de um local, como o caso de um docente com perfil de avaliador de propostas que deverá avaliar submissões dos pesquisadores proponentes de duas cidades distintas, por exemplo. Para esses casos, o vínculo não seria do campus de trabalho e sim de perfil de acesso, cuja interface será mostrada mais adiante. A figura 12 mostra a interface de cadastro de dados pessoais, seu vínculo e campus de lotação que a mesma possuirá. Para entendimento, é mostrado nas figuras 8 a 11 os diagramas de seqüência das etapas de consulta, inclusão, alteração e exclusão da interface de dados pessoais, cujo esquema de funcionamento é similar às demais interfaces desenvolvidas.



powered by Astah

Figura 8 : Diagrama de seqüência referente ao processo de consulta de Dados Pessoais

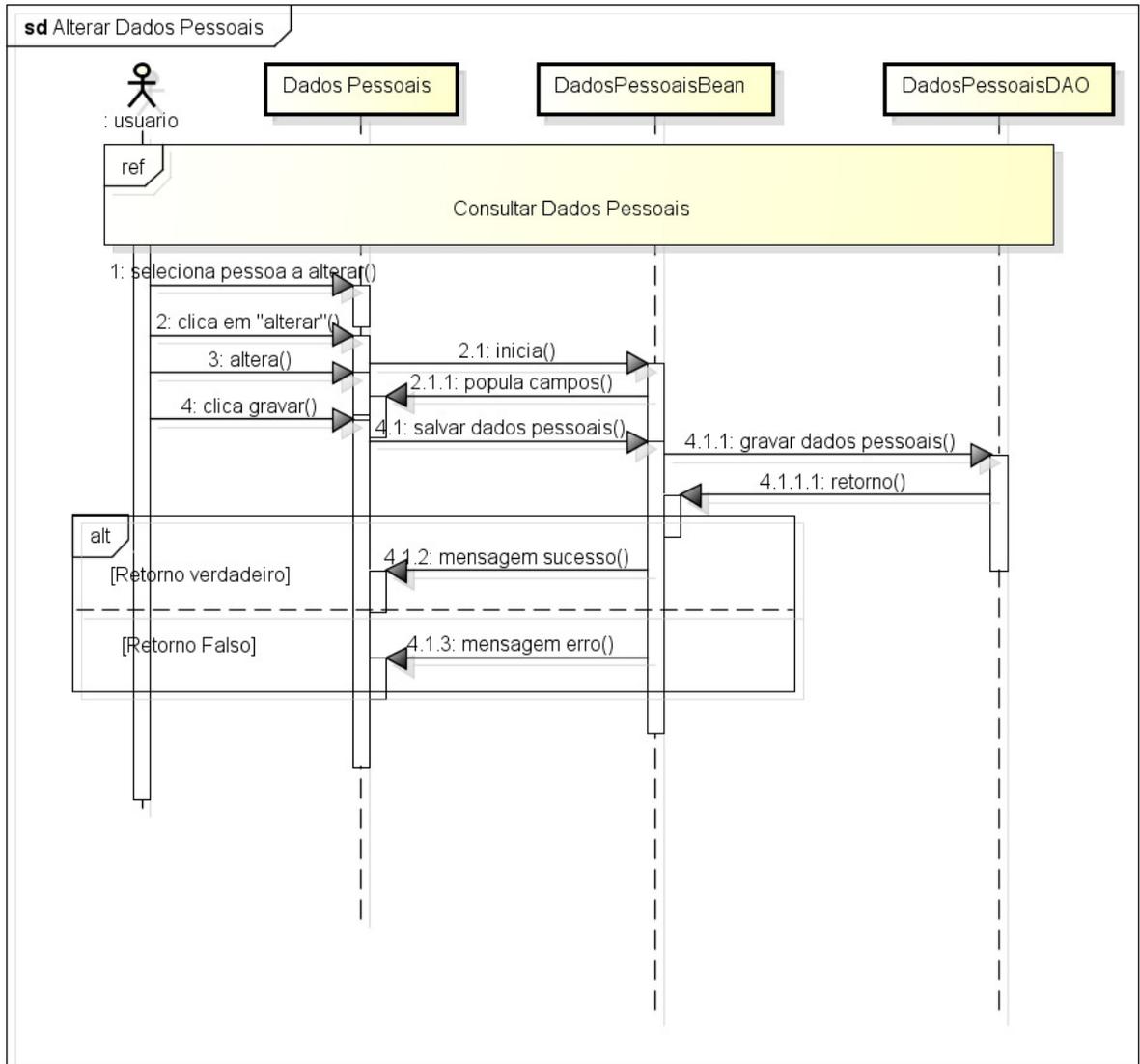
Fonte: arquivo do autor



powered by Astah

Figura 9 : Diagrama de seqüência referente ao processo de cadastro de Dados Pessoais

Fonte: arquivo do autor



powered by Astah

Figura 10 : Diagrama de seqüência referente ao processo de alteração de Dados Pessoais

Fonte: arquivo do autor

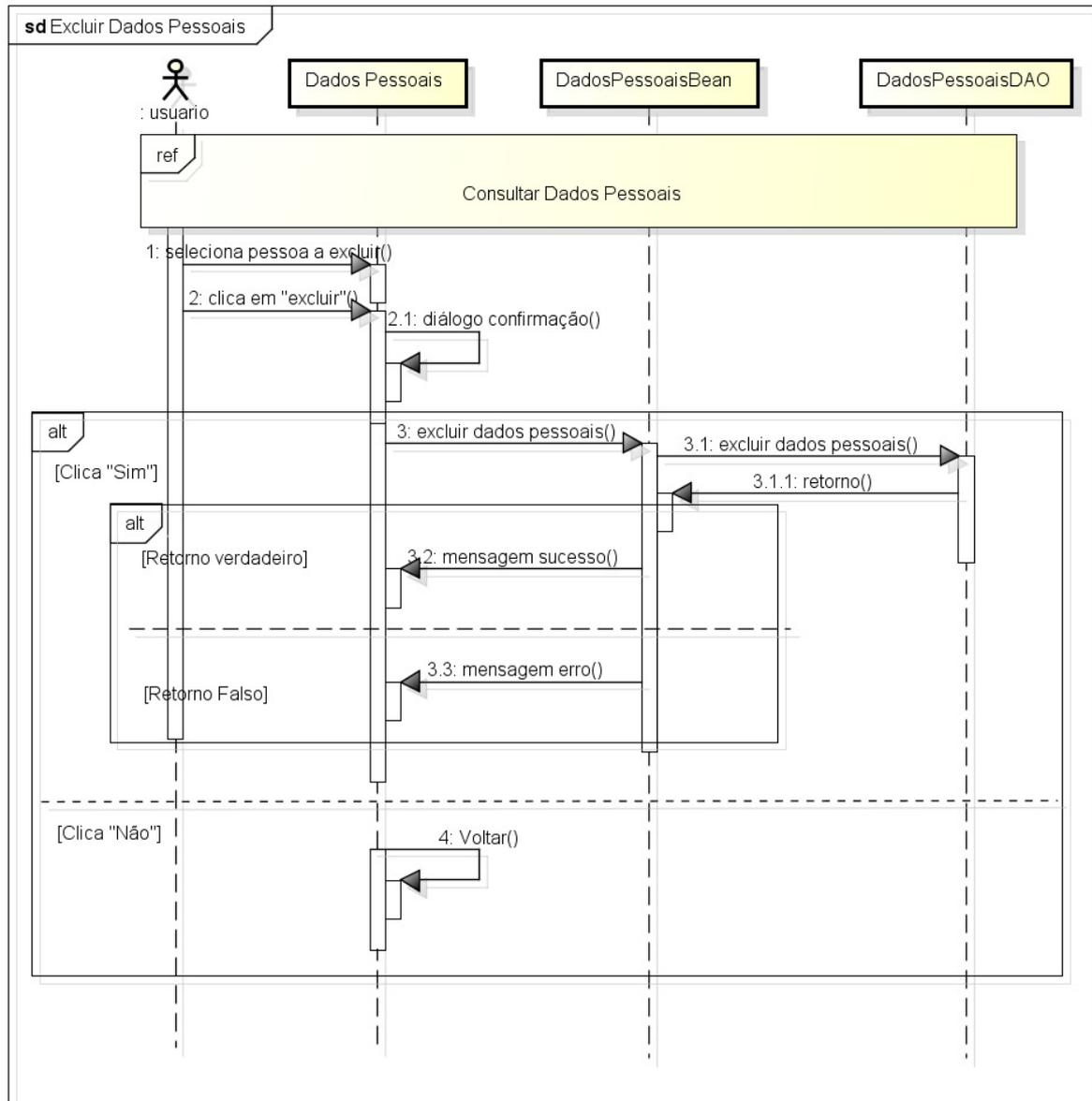


Figura 11 : Diagrama de seqüência referente ao processo de exclusão de Dados Pessoais

Fonte: arquivo do autor

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GUSTAVO D DUTRA  [Início](#) > [Cadastros](#) > [Dados Pessoais](#) > [Cadastro de Dados Pessoais](#)

FGCP

- ▼ **Cadastros**
 - Área do CNPQ
 - Dados Pessoais
 - I.E.S.
 - Nível Acadêmico
 - Outros Cadastros
- ▶ Configurações
- ▶ Projetos

Identificação	
Nome: *	Data de Nascimento: *
Estado Civil: *	Sexo: *
Nacionalidade: *	País de Origem: *
UF Nasc.: *	Município Nasc.: *
Documentos	
CPF: 136.714.722-00	Matrícula: *
RG: *	Órgão Expedidor: *
Data de Expedição: *	UF Expedição: *
Endereço	
País: * Brasil	
UF: *	Município: *
Bairro: *	CEP: *
Logradouro: *	Nº: ***
Complemento: *	

* Campos obrigatórios
 ** Para nacionalidade diferente de 'BRASILEIRA', no campo 'UF Nasc.' selecione 'EX' e no campo 'Cidade Nasc.' selecione 'ESTRANGEIRO'
 *** O número é obrigatório. Caso o endereço não possua este dado, favor informar 'S/N' (Sem número)

✔
✘

UTFPR
Setor de Tecnologia

Figura 12 : Interface de cadastro de dados pessoais.

Fonte: arquivo do autor

É possível alimentar as informações das tabelas de apoio do banco de dados da aplicação através do menu de "Cadastros", onde os usuários com o perfil de acesso "administrador" poderão manipular informações que refletem no funcionamento geral da aplicação. Os usuários com este perfil de acesso também tem acesso ao menu "Configurações" onde é possível atribuir perfis de acesso aos demais usuários da aplicação como mostra a figura 13.

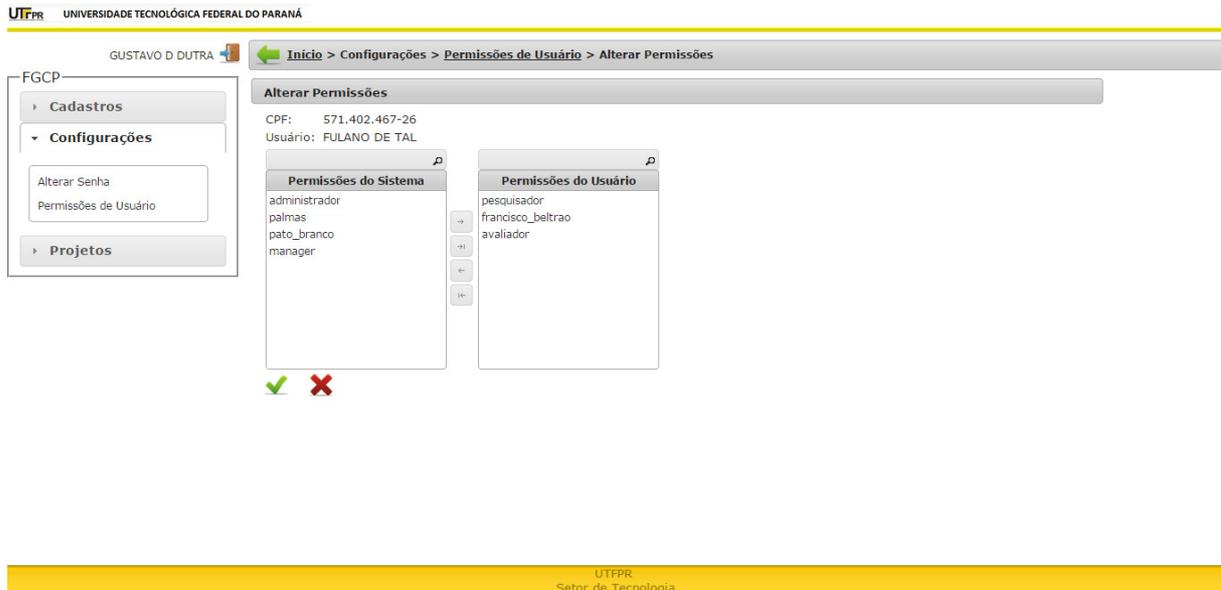


Figura 13 : Interface de atribuição de perfil de acesso à aplicação.

Fonte: arquivo do autor

Nesta primeira etapa de implementação, também encontra-se disponível o cadastro de projetos, dos quais faz-se necessário manter este cadastro para posteriormente vincular os editais de chamadas. Esta interface possui um filtro de busca seguida de uma lista de resultados, onde após selecionar um registro para edição abre-se o formulário de cadastro e edição conforme mostra a figura 14.

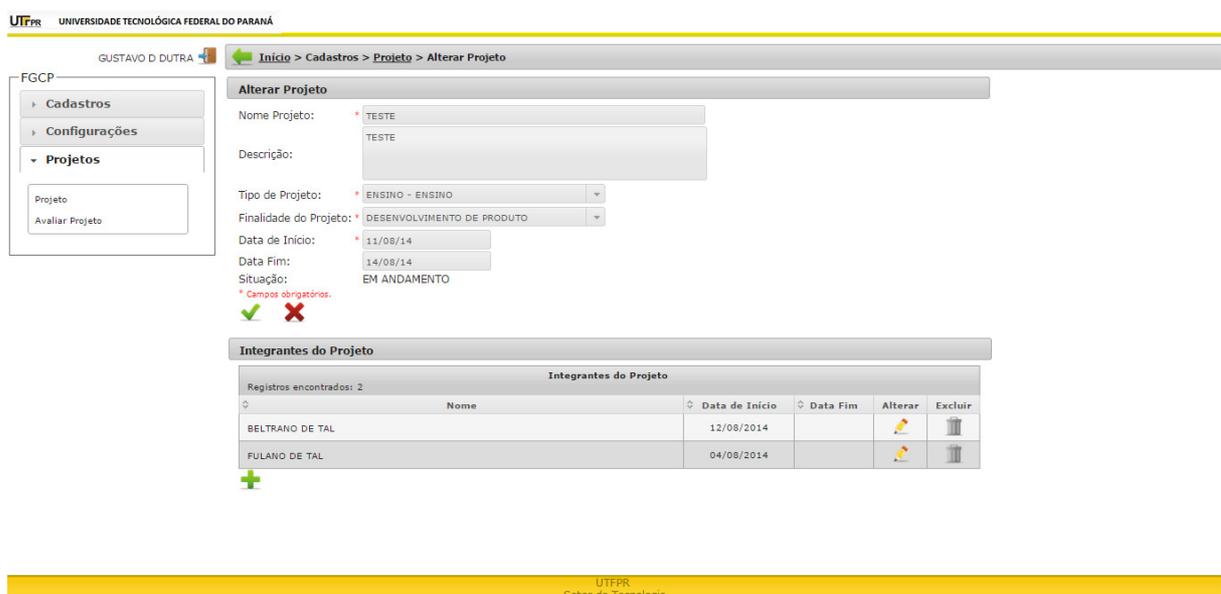


Figura 14 : Interface de cadastro e edição de projetos.

Fonte: arquivo do autor

As interfaces foram projetadas para permitir uma rápida adaptação do usuário com a ferramenta, com filtros de buscas que permitem localizar efetivamente a informação que se necessita. Do ponto de vista da segurança da informação, todas as ações realizadas são monitoradas através do registro de *logs* do sistema, onde para cada ação em uma determinada interface da aplicação, são registradas a data, hora, usuário, interface e ação realizada. Isto facilitaria em uma futura implementação de um módulo de registros de segurança de acesso onde o usuário final com a devida permissão de acesso, poderia monitorar as ações de entrada e manipulação de dados realizadas.

5 CONCLUSÃO E TRABALHOS FUTUROS

No decorrer deste projeto, foi finalizada a etapa inicial da construção do Sistema de Gerenciamento de Chamadas Públicas, contemplando os cadastros básicos, dados funcionais, permissões de acesso, projetos e vínculo da equipe de projeto. Procurou-se também preparar a estrutura do aplicativo para comportar as futuras implementações que deverão ser realizadas, modelando o banco de dados a fim de contemplar todas as tabelas necessárias já nesta etapa assim como as classes Java para atender o mapeamento objeto relacional das tabelas criadas.

A idéia proposta neste aplicativo é unir a experiência pessoal no projeto de novas aplicações corporativas com os conceitos de desenvolvimento de sistemas na plataforma web aprendidos no decorrer do curso. A ferramenta foi idealizada de forma a atender as necessidades de diversas instituições de ensino superior, sejam elas públicas ou privadas, uma vez que ela visa preencher a necessidade em gerir o ciclo de vida das chamadas públicas de seleção de projetos no alcance público e que há poucas opções disponíveis no mercado que visam atender este tipo de demanda.

O sistema desenvolvido até o momento compreende a implementação apenas dos cadastros iniciais e vínculos básicos atendendo ao item 1 - elaboração - do ciclo de vida de uma chamada pública, porém ele foi projetado para que se possa continuar com as demais implementações de forma a atingir o objetivo final de gerir todo o ciclo de vida das chamadas públicas.

Após essa implementação inicial pretende-se estender a implementação para a criação do controle de avaliações das submissões, envio dos convites aos avaliadores do projeto, criação de formulários de avaliação personalizados, o deferimento das avaliações feitas e o controle da prestação de contas de todos os gastos relativos à execução dos projetos executados. Após estas implementações faltantes, pretende-se implementar a área de relatórios com indicadores de produção de pesquisa de cada programa ou docente envolvido.

REFERÊNCIAS

CAMARA. **Projeto de Lei N° 6673 / 2006.** Disponível em: <http://www.camara.gov.br/sileg/integras/390517.pdf>. Acesso em 20/10/2014.

ÇIVICI, Çağatay. **Primefaces User's Guide 3.4.** Turquia, 2014. 475 p.

DEITEL, H.M. **Java: como programar.** 6.ed. São Paulo: Prentice Hall, 2005.

PRESSMAN, Roger S. **Engenharia de Software.** 7.ed. Porto Alegre: McGraw-Hill, 2011.

SILVEIRA, Paulo et al. **Introdução à arquitetura e design de software: Uma visão sobre a plataforma Java.** Rio de Janeiro: Elsevier, 2012.

SILBERCHATZ, Abraham; KORTH, Hank; SUDARSHAN, S. **Sistema de Banco de Dados:** Tradução da 5a. edição: Elsevier, 2006.

IBM. **Introdução a Plataforma Eclipse.** Disponível em: <http://www.ibm.com/developerworks/br/library/os-eclipse-platform/>. Acesso em: 25/09/2014.

PRIMEFACES. **Ultimate JSF Framework.** Disponível em: www.primefaces.org. Acesso em: 25/09/2014.

