

**UNIVERSIDADE FEDERAL DO ESPIRITO SANTO
DEPARTAMENTO DE INFORMATICA
PROJETO FINAL DE GRADUAÇÃO**

BRUNO BORLINI DUARTE

**APLICAÇÃO DO MÉTODO FRAMEWEB NO
DESENVOLVIMENTO DE UM SISTEMA DE
INFORMAÇÃO NA PLATAFORMA JAVA EE 7**

VITÓRIA
2014

BRUNO BORLINI DUARTE

Aplicação do Método FrameWeb no Desenvolvimento de uma Sistema de Informação na Plataforma Java EE 7

Trabalho de Conclusão de Curso
apresentado ao Departamento de
Informática do Centro Tecnológico da
Universidade Federal do Espírito Santo,
como requisito parcial para obtenção do
grau de Bacharel em Ciência da
Computação.

Orientador: Prof. Vitor Estevão Silva
Souza

VITÓRIA

2014

BRUNO BORLINI DUARTE

**Aplicação do Método FrameWeb no
Desenvolvimento de uma Sistema de Informação
na Plataforma Java EE 7**

COMISSÃO EXAMINADORA:

Prof. Vitor Estevão Silva Souza
Departamento de Informática, CT/Ufes
Orientador

Profª. Monalessa Perini Barcellos
Departamento de Informática, CT/Ufes

Prof. Fabiano Borges Ruy
Instituto Federal do Espírito Santo

Vitória, 11 de março de 2014.

Agradecimentos

A Deus por guiar meus passos nesse caminho.

À minha mãe pelo amor incondicional e por todo suporte que ela me dá todos os dias de minha vida, sem ela eu não teria conseguido chegar até esse momento.

Ao meu orientador Pr. Vitor Estêvão Silva Souza pelo incentivo, ajuda e principalmente por todo conhecimento que me passou, um bem inestimável.

À minha irmã Karina Duarte por sempre me apoiar e sempre acreditar em mim mesmo quando eu mesmo não acreditei.

À minha namorada, Aline Gasparini por todo amor, carinho e compreensão.

A todos os professores do Curso de Ciência da Computação da UFES, por tudo que me ensinaram durante todos esses anos.

Aos meus amigos pelos momentos de descontração e principalmente pelo companheirismo.

RESUMO

Inicialmente, a construção de aplicações Web se dava de forma *ad-hoc*. Entretanto, com o crescimento do número e da complexidade das *Webapps* (aplicações para a Web), foi necessário aplicar à construção deste tipo de aplicação os métodos da Engenharia de *Software*, adaptados a esta nova plataforma de implementação. Nasceria, assim, a Engenharia Web.

Nesse contexto, e com o objetivo de propor uma abordagem de construção de sistemas para Web, nasce então o método *FrameWeb*. A metodologia de desenvolvimento proposta pelo *FrameWeb* sugere a utilização de vários *frameworks*, visando a construção rápida e eficiente de sistemas de informação para a Web através de uma arquitetura básica para o desenvolvimento de uma Webapp e um perfil UML para quatro tipos de modelos de projeto que trazem conceitos utilizados por algumas categorias de frameworks.

A proposta de *FrameWeb* considera algumas categorias de *frameworks*. Dentro dessas categorias existem diversos frameworks que podem ser utilizados no desenvolvimento de um *software*. Esse trabalho aplica o método *FrameWeb* utilizando os frameworks da especificação Java EE 7 na criação do SCAP (Sistema de Controle de Afastamento de Professores) do departamento de informática da UFES e propõe melhorias nos modelos propostos pelo método, quando necessário.

O SCAP é um sistema que permite o controle dos afastamentos que os professores do departamento de informática da UFES, o sistema visa auxiliar os professores no processo de solicitação de afastamento, tornando-o mais dinâmico e eficiente.

ÍNDICE

Agradecimentos	3
RESUMO.....	4
1 Introdução	9
1.1 Objetivos.....	10
1.2 Metodologia.....	10
1.3 Organização da Monografia	11
2 Engenharia Web e o Método FrameWeb	12
2.1. Engenharia <i>Web</i>	13
2.2. <i>FrameWeb</i>	17
3 Especificação de Requisitos do SCAP	21
3.1. Descrição do Escopo	22
3.2. Modelo de Casos de Uso	23
4 Análise do SCAP	31
4.1 Modelagem e Criação do Diagrama de Classes	31
4.2 Restrições de Integridade do SCAP.....	33
4.3 Dicionário de Dados	34
5 Projeto e Implementação do SCAP	37
5.1 Arquitetura do Sistema	38
5.2. Frameworks Utilizados.....	41
5.2.1 <i>FrameWork JSF</i>	42
5.2.2 <i>FrameWork CDI</i>	42
5.2.3 <i>Framework Hibernate</i>	42
5.2.4 <i>Framework PrimeFaces</i>	43

5.3 Modelo de Domínio.....	44
5.4 Modelo de Persistência.....	47
5.5 Modelo de Navegação	50
5.6 Modelo de Aplicação.....	54
5.7 Implementação do SCAP.....	55
6 Propostas de Melhorias para o Método FrameWeb.....	60
6.1 Propostas de Melhorias para o Método FrameWeb a partir de experiências com diferentes tipos de Frameworks	60
7 Considerações Finais	64
7.1 Conclusões.....	64
7.2 Trabalhos futuros	65
Referências	67

LISTA DE FIGURAS

Figura 1 Modelo em três camadas proposto pelo FrameWeb (SOUZA & FALBO, 2007).	18
Figura 2 Casos de Uso do Secretário	24
Figura 3 Casos de Uso do Professor	25
Figura 4 Diagrama de Classes do SCAP	32
Figura 5 Pacotes do SCAP	41
Figura 6 Modelo de Domínio do SCAP	44
Figura 7 Enumerados do SCAP	46
Figura 8 Modelo de Persistência do SCAP	48
Figura 9 Modelo de Navegação para Cadastrar Usuário	51
Figura 10 Modelo de Navegação para Cadastrar Solicitacao	53
Figura 11 Modelo de Aplicação do SCAP	55
Figura 12 Tela de Login do SCAP	56
Figura 13 Tela de Cadastro de Solicitação de Afastamento.	57
Figura 14 Tela de Manifestar-se Contra Afastamento	58
Figura 15 Caixa de Dialogo Registrar Manifestação	59
Figura 16 Modelo de Navegação com melhorias propostas	61

LISTA DE TABELAS

Tabela 1 Atores	23
Tabela 2 Casos de Uso Cadastrais/Consultas.....	25
Tabela 3 Registrar Parecer CT - Fluxo de Eventos Normais	26
Tabela 4 Registrar Parecer PRPPG- Fluxo de Eventos Normais.....	26
Tabela 5 Arquivar Solicitação - Fluxo de Eventos Normais.....	27
Tabela 6 Solicitar Afastamento Nacional - Fluxo de Eventos Normais	27
Tabela 7 Solicitar Afastamento Internacional - Fluxo de Eventos Normais	28
Tabela 8 Solicitar Afastamento - Fluxo de Eventos Variantes	28
Tabela 9 Deferir Parecer - Fluxo de Eventos Normais.....	29
Tabela 10 Cancelar Solicitação - Fluxo de Eventos Normais.....	29
Tabela 11 Tecnologias utilizadas na implementação do SCAP.....	39
Tabela 12 Tabela de Estereótipos UML utilizados pelo Modelo de Navegação (SOUZA, 2007)	50

1 Introdução

Nos últimos anos, temos presenciado um crescimento no número de aplicações desenvolvidas para a *Web*. Essa plataforma de desenvolvimento tem uma grande vantagem, pois permite que as aplicações sejam acessadas de qualquer lugar e a qualquer momento, pelas mais diferentes pessoas ao redor do mundo.

Inicialmente, a construção desse tipo de aplicação se dava de forma *ad-hoc*. Entretanto, com o crescimento do número e da complexidade das *Webapps* (aplicações para a *Web*), foi necessário aplicar à construção deste tipo de aplicação os métodos da Engenharia de *Software*, adaptados a esta nova plataforma de implementação. Nasceria, assim, a Engenharia *Web* (SOUZA & FALBO, 2005).

Com o objetivo de propor uma abordagem de construção de sistemas para *Web*, nasce então o método *FrameWeb* (*Framework-based Design Method for Web Engineering*) (SOUZA, 2007). A metodologia de desenvolvimento proposta pelo *FrameWeb* sugere a utilização de vários *frameworks*, visando a construção rápida e eficiente de sistemas de informação para a *Web*.

FrameWeb propõe recomendações para o aumento da agilidade nas fases de desenvolvimento (Levantamento de Requisitos, Análise, Projeto e outras) e um perfil UML (extensão da linguagem UML) para a construção de quatro tipos de modelos na fase de projeto. Esses modelos trazem conceitos usados por algumas categorias de *frameworks*, o que facilita a comunicação entre projetistas e programadores sobre a arquitetura do sistema e como o mesmo deve ser implementado.

Neste projeto, foi desenvolvido um Sistema de Controle de Afastamentos de Professores (SCAP), uma *Webapp* para auxiliar um departamento de universidade a controlar solicitações de afastamento de seus professores efetivos. O projeto seguiu técnicas de Engenharia de *Software* e Engenharia *Web*, aplicando o método *FrameWeb* para o projeto arquitetural e desenvolvendo a aplicação na plataforma Java EE 7 (DEMICHIEL, 2013).

1.1 Objetivos

O objetivo geral deste trabalho é a aplicação do método *FrameWeb* (SOUZA, 2007) utilizando os *frameworks* da plataforma Java EE 7 (DEMICHIEL, 2013), em especial seus componentes, CDI¹, JSF² e Facelets, ao invés dos *frameworks* propostos originalmente pelo *FrameWeb*, respectivamente: Spring, Struts2 e Sitemesh para desenvolver uma aplicação para *Web* que possa gerenciar e controlar as solicitações de afastamentos de professores no departamento de informática da UFES (SCAP).

Os objetivos principais desse trabalho são: demonstrar a necessidade de evolução de *FrameWeb*, proposto originalmente em 2007, utilizando as tecnologias e *frameworks* existentes na época e através da experiência de criação do SCAP, criar propostas de melhorias para os modelos que são definidos pelo método.

1.2 Metodologia

Esta seção define a metodologia adotada para a criação do projeto de graduação proposto neste documento:

- Pesquisa: pesquisar sobre o *FrameWeb* e sobre os diversos *frameworks* nos quais o método foi inicialmente testado para o desenvolvimento de uma aplicação *Web* Java. Estudar a plataforma Java EE 7 e o desenvolvimento de *Webapps* nesta plataforma;
- Análise do Problema Proposto: tendo os professores do Departamento de Informática da UFES como *stakeholders*, os requisitos do SCAP, Sistema de Controle de Afastamentos de Professores, foram levantados e analisados;
- Projeto: a partir dos resultados obtidos no item acima, foi desenvolvido o projeto do sistema SCAP, utilizando a metodologia e os modelos de projeto propostos por *FrameWeb* ;
- Desenvolvimento: seguir a arquitetura de desenvolvimento proposta pelo método para a criação da aplicação SCAP;

¹ Contexts and Dependency Injection for the Java EE platform, <https://jcp.org/en/jsr/detail?id=299>

² JavaServer Faces, <https://jcp.org/en/jsr/detail?id=314>

- Finalização: a finalização do projeto contou com os testes finais do sistema *Web* além dos pequenos ajustes ou melhorias que foram realizadas;
- Apresentação do Projeto: apresentação final do projeto. O projeto e a ferramenta foram entregues e demonstrados.

1.3 Organização da Monografia

Essa monografia esta dividida em 7 capítulos incluindo a introdução.

No capítulo 2 é feito um levantamento geral sobre temas relevantes a esse trabalho: Engenharia *Web* e o Método *FrameWeb*.

No capítulo 3 é feita a análise de requisitos do SCAP (Sistema de Controle de Afastamento de Professores) uma aplicação *web* que será desenvolvida utilizando o método *FrameWeb*.

O capítulo 4 apresenta a especificação de análise do SCAP dentro do paradigma de Orientação à Objetos.

O capítulo 5 aplica as indicações do método *FrameWeb* no projeto do SCAP. O projeto mostra os vários modelos propostos pelo método.

O capítulo 6 propõe melhorias para o método *FrameWeb* através da experiência de utiliza-lo na construção do SCAP.

O Capítulo 7 apresenta as conclusões desse trabalho e as expectativas para trabalhos futuros.

2 Engenharia Web e o Método FrameWeb

Nos últimos anos a *Web* tornou-se presente em muitas situações na vida de milhares de pessoas em todo mundo, ultrapassando muitos desenvolvimentos tecnológicos presentes na história (GINIGE & MURUGESAN, 2001). A *Web* rapidamente cresceu e diversificou seu uso, afetando assim os aspectos da vida cotidiana.

O extraordinário crescimento da *Web* teve um impacto significativo nos negócios, comércio, indústria, finanças, entretenimento, na vida pessoal e profissional das pessoas. A *Web* tornou-se o aspecto central de muitas aplicações em diferentes áreas. Hoje, diversos setores de negócio realizam suas operações no ambiente da Internet, muitos sistemas de informações considerados legados estão sendo migrados para aplicações na *Web*. Uma vasta abrangência de novas e complexas aplicações corporativas está emergindo na *Web* (LI et al, 2000).

Tais aplicações executam em um servidor conectado à *World Wide Web* (WWW) e podem ser acessadas de qualquer computador ou dispositivo conectado à Internet através de um *browser* (navegador de internet), que utiliza-se de protocolos como o HTTP (*HyperText Transfer Protocol*) e da linguagem HTML (*HyperText Markup Language*) para exibir em páginas *Web* uma interface gráfica para o usuário. Essa forma facilitada de acesso vem removendo as limitações espaciais antes existentes: anteriormente o usuário só poderia acessar um sistema ou aplicação caso essa estivesse previamente instalada em seu computador ou caso ele estivesse conectado fisicamente à rede onde a aplicação esta instalada.

Este capítulo mostra a adaptação da Engenharia de *Software* para satisfazer necessidades das *Webapps*, área que recebeu o nome de Engenharia *Web*, e apresenta o método *FrameWeb*, proposto em (SOUZA, 2007) para auxiliar na construção dessas aplicações, baseado no uso de *frameworks* que agilizam e facilitam o trabalho de projeto e desenvolvimento das aplicações.

2.1. Engenharia Web

Recentemente, observamos acontecer uma nova “Crise de *Software*” (GIBBIS, 1994), porém dessa vez ocorrida no desenvolvimento de aplicações para *Web* e por esse motivo chamada de “*Crise Web*” (HANSEN et al., 1999). Esse fato motivou a criação da Engenharia *Web* (*Web Engineering*), que é a Engenharia de *Software* voltada para o desenvolvimento de aplicações *Web* (PRESSMAN, 2005). A Engenharia *Web* trata de abordagens disciplinadas e sistemáticas para o desenvolvimento, implantação e manutenção de sistemas baseados da *Web* (DESPHANDE, 2001).

A Engenharia *Web* está se tornando cada vez mais necessária devido ao aumento do número e da complexidade de aplicações feitas para a *Web*. Com esse aumento de complexidade surge a necessidade de um maior planejamento e de uma maior organização na hora de se construir uma *Webapp*. Tais aplicações estão se tornando cada vez mais importantes para o funcionamento de uma organização, muitas vezes sendo responsável até por gerar o lucro da empresa. Por exemplo, no caso das lojas virtuais, caso a aplicação pare de funcionar a loja não será mais capaz de vender seus produtos, o que pode até mesmo causar o fim da organização. Ou seja, sistemas *Web* tem se tornado cada vez mais ferramentas críticas de negocio para diversas organizações (GINIGE, 2002). Com isso há a necessidade de construir aplicações de fácil utilização, confiáveis e capazes de atender as expectativas dos usuários, que crescem exponencialmente. Para isso é necessário que haja uma abordagem disciplinada em seu desenvolvimento (PRESSMAN, 2005).

A Engenharia *Web* lida diretamente com todos os aspectos do desenvolvimento de sistemas *Web*, desde a concepção e seu desenvolvimento até a implementação, avaliação e sua evolução. Sua essência é gerenciar a diversidade e a complexidade das aplicações *Web* e, assim, evitar falhas de projeto que possam causar grandes problemas (GINIGE & MURUGESAN, 2001).

Para que haja qualidade no desenvolvimento das aplicações é altamente recomendável o uso da Engenharia *Web*, que incorpora vários conceitos da

Engenharia de Software, porém, com características próprias do meio aonde essas aplicações serão executadas.

As aplicações criadas para a plataforma *Web*, encontram os mesmos problemas que as aplicações *desktop* em seu desenvolvimento: erros de estimativas, falta de uso de modelos de processos já previamente estabelecidos, planejamento equivocado por parte dos analistas responsáveis, inexperiência dos desenvolvedores, falta de documentação, *layout* inamistoso para o usuário, prazos apertados, dentre outros.

Da mesma forma, todos os atributos de qualidade de software se aplicam também a sistemas baseados na *Web*. Entretanto, há aqueles que são mais relevantes e fornecem uma base mais útil para avaliar a qualidade de um software. Em (PRESSMAN, 2005) é apresentada uma lista de requisitos de qualidade que identificam um conjunto de atributos e levam à alta qualidade de aplicações *Web*.

- **Usabilidade:** refere-se à facilidade de utilização da aplicação, mesmo para usuários com pouco conhecimento técnico;
- **Eficiência:** o tempo de resposta do *site* ao carregar páginas, imagens ou gráficos deve ser o menor possível;
- **Confiabilidade:** refere-se à recuperação de erros, validação das entradas passadas pelo usuário e funcionamento correto dos links do *site*;
- **Manutenibilidade:** devido à constante evolução das tecnologias de desenvolvimento de sistemas e à necessidade de atualização de informações o sistema deve ser construído de forma intuitiva e deve ser de fácil manutenção e evolução;
- **Funcionalidade:** o sistema deve buscar informações corretamente e ter todas as suas operações funcionando.

Vários outros trabalhos foram e têm sido apresentados na literatura sobre como e o que deve ser avaliado em aplicações *Web*. Alguns desses trabalhos podem ser vistos em (BEEVAN, 1997; BORGES, 1996; OLSINA, 1999; LOWE, 1999). Para que uma *Webapp* seja capaz de atingir os requisitos de qualidade mencionados acima, algum processo que garanta tais

atributos deve ser seguido. Como já dito anteriormente, a Engenharia *Web* utiliza-se dos métodos propostos pela Engenharia de Software, devidamente adaptados à criação de *Webapps*.

O modelo de projeto proposto pela Engenharia *Web* é dividido em várias fases e segue uma abordagem iterativa, ao final de cada fase tem-se um produto importante para a construção do sistema e ao final de todas as fases tem-se o *software* devidamente implementado e testado e atendendo aos requisitos e expectativas do cliente.

A primeira fase da Engenharia *Web* é a Análise de Requisitos. Requisitos de um sistema são descrições dos serviços que devem ser fornecidos por esse sistema e as suas restrições operacionais (SOMMERVILLE, 2007).

Essas características podem ser referentes ao conteúdo, à estruturação, ao acesso ou mesmo a estética de uma aplicação. Requisitos de estruturação especificam como o conteúdo da aplicação deve ser organizado, requisitos de acesso definem o estilo de acesso à informação, o que inclui propriedades na apresentação de conteúdos, facilidades de pesquisas e suporte. Requisitos de estética definem as propriedades da aparência em geral (CODA et al, 1998).

A análise de requisitos contem as atividades responsáveis pelo mapeamento do problema que a *Webapp* deve resolver e coletar os requisitos com os *stakeholders* (usuários ou pessoas interessadas na criação do sistema) e os principais objetivos dessa fase são: definir as diversas categorias de usuário, definir os objetivos da aplicação *Web* e estabelecer um conhecimento básico respondendo as seguintes questões (BEZERRA, 2004):

- Quais são os objetivos que a *Webapp* deve atingir?
- Quem vai utilizar a *Webapp*?
- Qual a principal necessidade de negócio que a *Webapp* deve suprir?

De posse dessas informações os analistas responsáveis já um problema bem definido e os dados necessários para dar início a próxima fase do desenvolvimento.

A próxima fase sugere a criação de um modelo de análise. O modelo de análise pode ser dividido em quatro tópicos: a análise de conteúdo que identifica as classes de conteúdo que devem ser fornecidas para a *Webapp*, a análise de interação que descreve a forma pelo qual o usuário interage com a *Webapp*, a análise funcional define as operações que são aplicadas ao conteúdo da *Webapp* e a sequência de processamentos que ocorrem como consequência e por último, a análise de configuração que é responsável por descrever o ambiente operacional e a infraestrutura na qual a *Webapp* reside (PRESSMAN, 2005).

Após terminar com o modelo de Análise da *Webapp*, A Engenharia *Web* propõe que seja elaborado o modelo de projeto. Guiado pelas informações obtidas no modelo de análise, o projeto da *Webapp* foca-se em seis tópicos principais: projeto de conteúdo, projeto arquitetural, projeto de estética, projeto de navegação, projeto de interface e projeto de componentes (PRESSMAN, 2005). Vale ressaltar que a não utilização desses modelos provavelmente resultará em uma aplicação fraca, insegura e ineficiente. Portanto é extremamente importante que o modelo de projeto seja devidamente planejado e desenvolvido.

Segundo o processo da Engenharia *Web*, é a fase de implementação, que é onde o projeto é devidamente construído utilizando uma linguagem de programação escolhida de acordo com as características e requisitos do projeto. Por último, temos a fase de testes que é responsável por garantir a qualidade e garantir que a *Webapp* foi construída de acordo com os requisitos obtidos na fase de levantamento de requisitos. Os testes em *Webapps* enfocam primariamente conteúdo, navegabilidade, segurança, carga, eficiência e interoperabilidade (entre diferentes navegadores *Web*).

Dentro do contexto da Engenharia *Web*, foi proposto o método *FrameWeb*, um método que propõe a otimização dos processos da Engenharia *Web*. Este método será discutido a seguir.

2.2. FrameWeb

A todo tempo tentamos otimizar os processos da Engenharia *Web*. *FrameWeb* surge com este propósito. Além de propor a utilização dos diversos tipos de *frameworks* que existem para se possa conseguir agilidade principalmente na fase de codificação da aplicação, *FrameWeb* nos apresenta também (SOUZA & FALBO, 2005):

- Recomendações para aumentar a agilidade nas principais fases de desenvolvimento (requisitos, análise projeto);
- Uma arquitetura básica para o desenvolvimento de *Webapps*;
- Um perfil UML para um conjunto de modelos de projeto que trazem conceitos utilizados por algumas categorias de *Frameworks*.

FrameWeb é então um método para a modelagem de *Webapps* que assume a utilização de algumas categorias de *frameworks* durante o processo de software. A principal ideia é aumentar a agilidade nas principais fases do desenvolvimento, principalmente nas fases de levantamento de requisitos, análise, projeto e desenvolvimento.

As fases de levantamento de requisitos e análise podem ser conduzidas da forma que a organização considerar mais a adequada. No entanto, *FrameWeb* sugere que sejam utilizados o modelo de casos de uso para apresentação de requisitos e diagramas UML para a modelagem conceitual. O método sugere a utilização do diagrama de classes para a modelagem dos conceitos do domínio do problema. Esse modelo é criado com um alto nível de abstração e sem levar em consideração a plataforma de implementação que será utilizada. Caso o analista responsável ache necessário, outros diagramas UML podem ser utilizados. Após essas fases, serão realizadas as atividades da fase de projeto.

O método da uma maior importância à fase de projeto (SOUZA, 2007). Levando em consideração os diversos tipos de *frameworks* existentes, *FrameWeb* propõe uma arquitetura básica no desenvolvimento de *Webapps*. Essa arquitetura divide o sistema em três camadas, como mostra a figura 1.

Temos a camada de lógica da apresentação, lógica do negócio e lógica de acesso à dados.

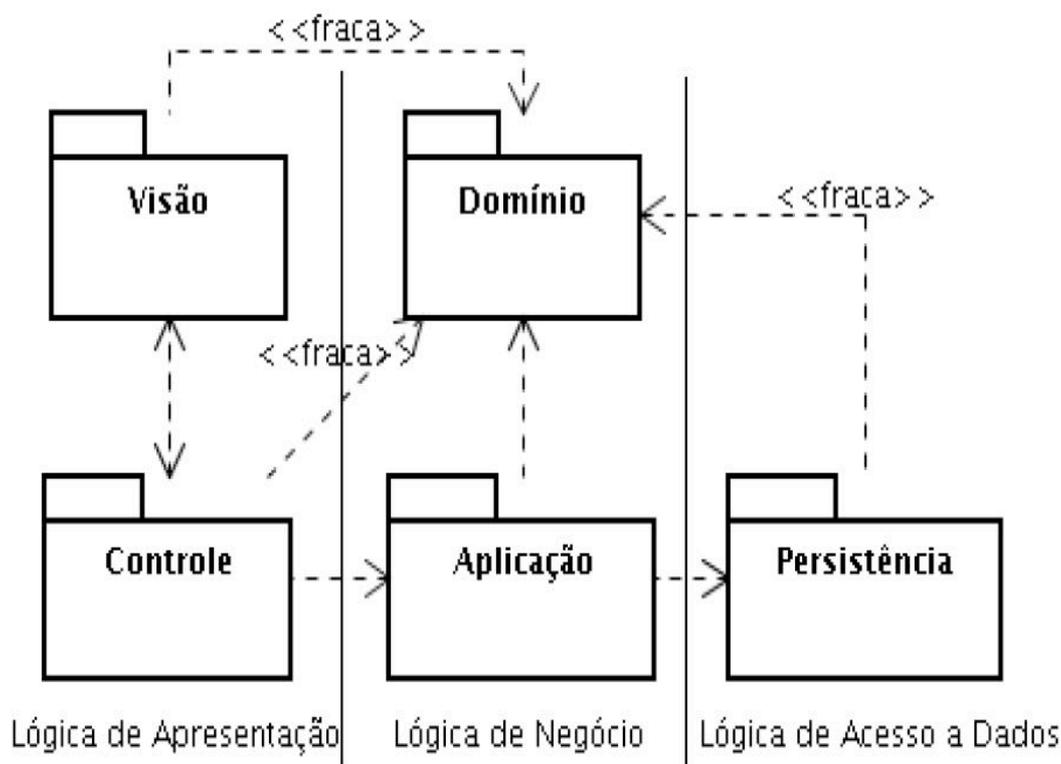


Figura 1 Modelo em três camadas proposto pelo FrameWeb (SOUZA & FALBO, 2007).

A primeira camada, a camada de lógica da apresentação tem o objetivo de prover as interfaces gráficas para a comunicação da *Webapp* com o usuário. O pacote *Visão* contém as páginas web, imagens e scripts que são executados do lado do cliente e outros arquivos que relacionados a exibição de informações para o usuário. O pacote *Controlador* contém classes de ação que são responsáveis por tratar a interação do usuário com os elementos presentes no pacote de *Visão*.

A segunda camada, a camada de lógica de Negócios contém dois pacotes: *Domínio* e *Aplicação*. O primeiro contém as classes que foram identificadas na fase de análise como sendo conceitos do domínio do problema e foram modeladas nos diagramas de classes criados na fase de análise. O segundo pacote é onde estão implementados os casos de uso definidos na

fase de análise de requisitos provendo assim uma camada de serviços que deve ser independente da interface com o usuário.

A terceira e última camada, lógica de Acesso a Dados contém o pacote de persistência, que contém as classes responsáveis por armazenar os objetos persistentes em um banco de dados.

Pela figura 1, vemos que os pacotes Visão e Controlador tem uma interdependência, visto que os elementos presentes no pacote de Visão enviam estímulos causados pelo usuário (clique de um botão, preenchimento de um campo texto, acionamento de uma caixa de seleção) e esses estímulos devem ser devidamente tratados e respondidos pelas classes presentes no pacote Controle. O pacote Controle também possui uma interdependência com o pacote Aplicação, que prove ao usuário acesso as principais funcionalidades do sistema.

O pacote Aplicação possui duas dependências, uma com o pacote de Domínio, pois manipula objetos desse pacote e a outra com o pacote de persistência, para que se possa recuperar, gravar, alterar e excluir objetos de domínio de acordo com a execução dos casos de uso. Os pacotes Visão, Controle e Persistência possuem uma dependência fraca com o pacote de Domínio, o que representa um baixo acoplamento entre os pacotes. O relacionamento <<fraco>> indica que não são feitas alterações nos objetos de domínio. No pacote de Visão, objetos de domínio são utilizados apenas para exibir seus dados em uma tela para o usuário, no pacote Persistência os objetos de domínio são utilizados para realizar o mapeamento do banco de dados relacional (SOUZA, 2007).

O *FrameWeb* apresenta também uma linguagem de modelagem baseada em UML para modelar classes e outros artefatos para a integração com diversos *frameworks* a fim de representar os componentes que são tipicamente utilizados no desenvolvimento *Web* e os componentes relacionados com o uso de *frameworks* (SOUZA, 2007). A ideia é criar quatro diagramas: modelo de domínio, modelo de persistência, modelo de navegação e modelo de aplicação.

- O **modelo de domínio** representa os objetos que fazem parte do domínio do problema e seu mapeamento para a persistência;
- O **modelo de persistência** representa os objetos que são responsáveis pela persistência dos dados gerados pelo sistema;
- O **modelo de aplicação** representa as classes de serviço que são responsáveis por implementar a lógica de negócio do sistema e suas dependências, independentemente das telas (interface com o usuário);
- O **modelo de navegação** representa as páginas *Web* do sistema e os diversos atributos que as compõem e que interagem com a camada de controle do sistema para, assim, controlar e enviar os estímulos mandados e recebidos do usuário.

Esses modelos serão mais amplamente demonstrados e discutidos no capítulo 5 desse trabalho. Uma descrição mais completa do método pode ser lida em (SOUZA, 2007).

Nos próximos capítulos desse trabalho apresentaremos o levantamento de requisitos, análise, projeto e implementação do SCAP seguindo a especificação do *FrameWeb* porém utilizando os *frameworks* presentes na plataforma Java EE 7 e não os *frameworks* originalmente propostos. Esse desenvolvimento será feito com o intuito de analisar como o método se comporta ao utilizarmos os *frameworks* mais recentes da plataforma Java.

3 Especificação de Requisitos do SCAP

A especificação de requisitos envolve as atividades relacionadas com o levantamento e a definição dos requisitos de software de um sistema. . Nessa fase, um esforço conjunto de clientes, usuários e especialistas de domínio é necessário, com o objetivo de entender a organização, seus processos, necessidades, deficiências dos sistemas de software atuais, possibilidades de melhorias, bem como restrições existentes. Trata-se de uma atividade complexa que não se resume somente a perguntar às pessoas o que elas desejam, mas sim analisar cuidadosamente a organização, o domínio da aplicação e os processos de negócio no qual o sistema será utilizado (KOTONYA; SOMMERVILLE, 1998).

O principal objetivo da Especificação de Requisitos é entender como o sistema será utilizado pelo usuário. Seu principal produto deve ser uma modelagem coerente das funcionalidades solicitadas.

Os requisitos podem ser funcionais ou não funcionais. Requisitos funcionais, como o próprio nome indica, apontam as funções que o sistema deve prover e como o sistema deve se comportar em determinadas situações. Já os requisitos não funcionais descrevem restrições sobre as funções a serem providas, restrições essas que limitam as opções para criar uma solução para o problema, tais como restrições de tempo e de recursos, ou restrições às quais o sistema como um todo (ou mesmo o projeto de desenvolvimento) está sujeito. Há ainda as regras de negócio, as quais são derivadas do domínio de aplicação e podem restringir ou estabelecer requisitos funcionais existentes, como cálculos específicos que devem ser realizados, refletindo fundamentos do domínio de aplicação (FALBO, 2011).

Nesse capítulo descrevemos a especificação de requisitos funcionais do SCAP – o Sistema de Controle de Afastamento de Professores, uma aplicação *Web* implementada seguindo o método *FrameWeb* (SOUZA, 2007), porém utilizando a plataforma Java EE 7 no lugar dos *frameworks* utilizados pela proposta original.

Essa atividade foi desenvolvida usando a técnica de modelagem de casos de uso (SILVA, 2007) e, portanto, este capítulo apresenta, na seção 3.2, os diagramas de caso de uso gerados, bem como as descrições dos atores e dos casos de usos identificados nesses diagramas. Antes disso, o escopo do sistema é descrito na seção 3.1.

3.1. Descrição do Escopo

No SCAP devem estar disponíveis informações sobre as solicitações de afastamento dos professores do Departamento de Informática (DI) da UFES. As solicitações de afastamento podem ocorrer tanto para eventos no Brasil quanto no exterior. Essas solicitações precisam ser avaliadas pelos professores do DI e, em alguns casos, pela diretoria do Centro Tecnológico (CT) e pela Pró-reitora de Pesquisa e Pós-Graduação (PRPPG) para que sejam aprovadas e o professor possa realizar sua viagem.

Caso o pedido de afastamento realizado para o professor seja para um evento que será realizado no Brasil o professor precisa da aprovação de seu chefe direto, que é o professor que está exercendo o mandato de Chefe do Departamento ao qual pertence. Dessa forma, para realizar um afastamento para um evento nacional a avaliação da solicitação não sai de dentro do DI.

No caso de um pedido de afastamento para um evento que acontecerá no exterior, será necessário que o professor receba a aprovação do CT e da PRPPG para que a solicitação de afastamento seja aprovada e publicada em Diário Oficial da União, conforme art. 95 da lei 8.112³.

No entanto, o SCAP trata da tramitação de solicitação de afastamentos apenas dentro do Departamento de Informática (DI) da UFES, de forma que os processos de tramitação utilizados pelo CT e pela PRPPG fogem ao escopo do sistema e não há nenhuma integração com o sistema de acompanhamento de Processo da UFES. Ele deverá, no entanto, lidar com a tramitação de pedidos de afastamento no exterior enquanto os mesmos encontrarem-se dentro do departamento.

³ http://www.planalto.gov.br/ccivil_03/leis/l8112cons.htm

O SCAP foi projetado para que possa auxiliar o professor nessas solicitações e para tornar o processo de solicitação de afastamento mais dinâmico, através do envio de e-mails automáticos para os interessados e do uso de formulários para ajudar na criação dos documentos necessários para se realizar uma solicitação.

O SCAP também permite que os demais professores e funcionários do DI consultem sobre as solicitações de afastamento em tramitação.

3.2. Modelo de Casos de Uso

O modelo de casos de uso visa capturar e descrever as funcionalidades que um sistema deve prover para os atores que interagem com o mesmo, isto é, os usuários do sistema (SILVA, 2007). Em um diagrama de casos de uso é comum algum dos casos de uso ser considerado mais crítico do que os outros para a lógica de negócio do sistema.

Os casos de uso do sistema estão descritos de forma tabular de acordo com o modelo proposto por (FALBO, 2010). Na tabela 1 estão descritos os atores que foram identificados no contexto deste projeto.

Tabela 1 Atores

Ator	Descrição
Professor	Professores efetivos do DI/UFES.
Chefe de Departamento	Professores do DI/UFES que estão realizando a função administrativa de chefe/subchefe do departamento
Secretário	Secretário do DI/UFES.

O Secretário é o responsável por realizar os casos de uso cadastrais do sistema (Cadastrar Usuário e Cadastrar Chefe de Departamento) e os casos de uso referentes a registrar pareceres externos ao DI no SCAP (Registrar Parecer CT e Registrar Parecer PRPPG). O secretário também é responsável

por arquivar uma solicitação de afastamento quando esta estiver devidamente regularizada.

O professor é o ator principal do sistema, ele é o responsável por realizar os pedidos de solicitações de afastamento, por realizar o papel de relator em uma solicitação de afastamento, deferindo um parecer sobre a mesma e ter o poder de se manifestar contra solicitações de afastamento abertas por outro professor do departamento.

O chefe de departamento também é um professor do Departamento de Informática da UFES, ele pode executar todos os casos de uso associados a um professor e ainda é o responsável por encaminhar uma solicitação de afastamento a seu relator.

As figuras 2 e 3 apresentam respectivamente os diagramas de casos de uso relacionados ao secretário e a um professor do DI.

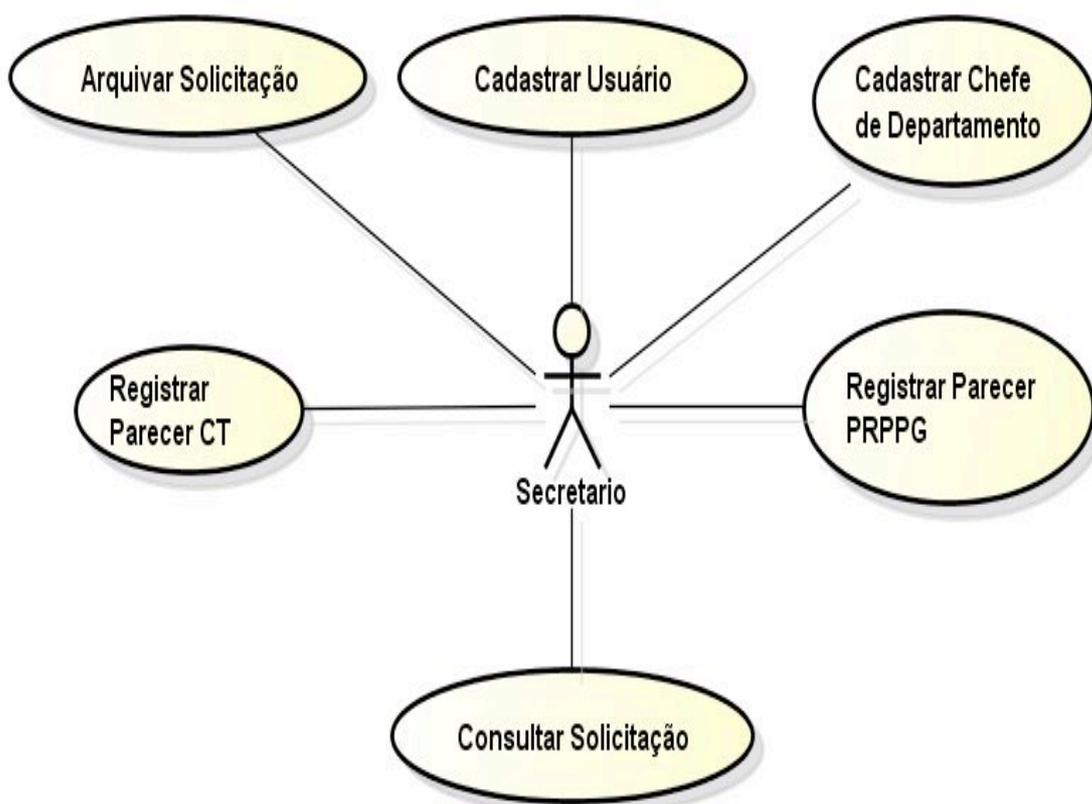


Figura 2 Casos de Uso do Secretário

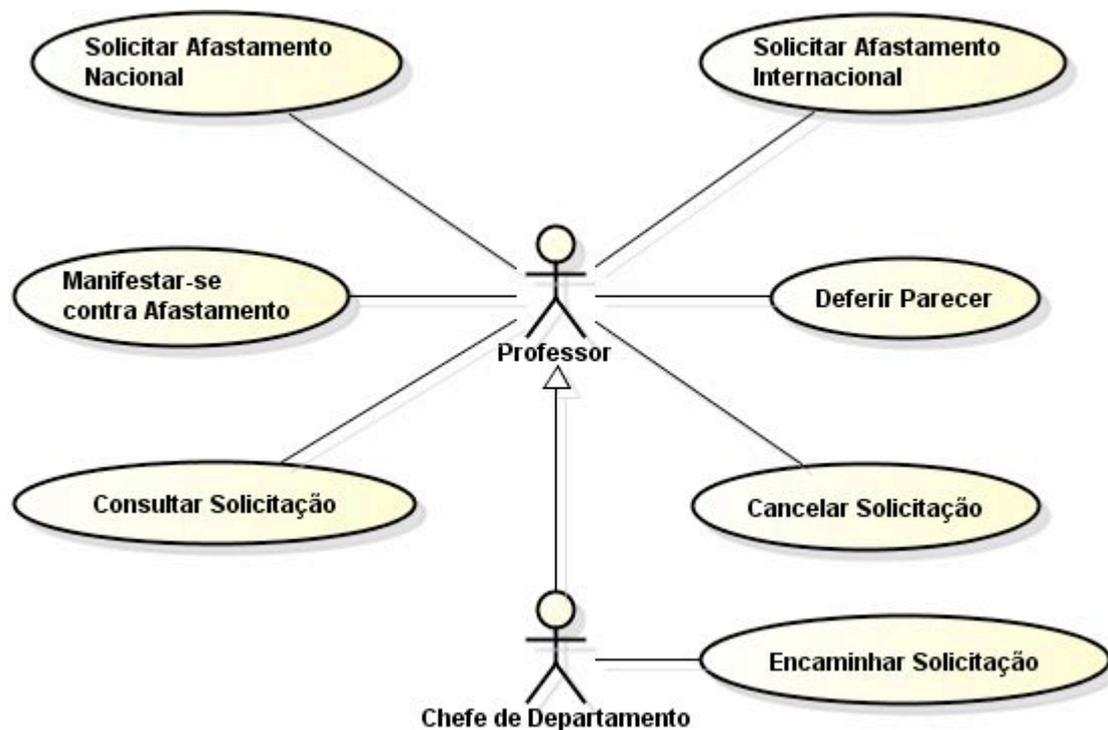


Figura 3 Casos de Uso do Professor

A seguir, são apresentadas as descrições de cada um dos casos de uso identificados. Os casos de uso cadastrais, considerados de baixa complexidade, envolvendo inclusão, alteração, consulta e exclusão, são descritos na tabela 2.

Tabela 2 Casos de Uso Cadastrais/Consultas

Caso de Uso	Ações Possíveis	Observações	Requisitos	Classes
Cadastrar Usuário	I, A,C,E	[!] Informar: nome, matrícula, e-mail e password e o tipo do usuário		Professor, Pessoa, Secretário

Cadastrar Chefe de Departamento	I, A,C,E	[I] Informar: data de início de mandato para um professor já previamente cadastrado	O professor estar cadastrado no sistema	Pessoa, Professor, Mandato.
Consultar Solicitação	C	[C] Consultar uma solicitação de Afastamento já existente no sistema exibindo seus dados na tela.	A solicitação consultada estar cadastrada no sistema.	Solicitacao, Pessoa, Parecer

A seguir, são apresentados os casos de uso de maior complexidade que não puderam ser descritos segundo os formatos tabulares simplificados. Esses casos de uso são descritos segundo o padrão de descrição completa de casos de uso definido.

Caso de Uso: Registrar Parecer CT

Descrição Sucinta: Este caso de uso representa o cadastramento do parecer do Centro Tecnológico sobre a solicitação de afastamento.

Tabela 3 Registrar Parecer CT - Fluxo de Eventos Normais

Nome do Fluxo de Eventos Normal	Precondição	Descrição
Registrar Parecer CT	1 – Status da solicitação estar marcado como: “Aprovado-DI”	1 - Secretário faz login no sistema, seleciona a solicitação desejada e registra o parecer do Centro Tecnológico com relação à solicitação de afastamento. 2 - Secretário Faz upload do documento enviado pelo CT referente ao parecer. 3 - Caso o parecer seja positivo status da solicitação é alterado para “Aprovada-CT” 4 – Caso o parecer seja negativo o status da solicitação é alterado para “Reprovada”.

Caso de Uso: Registrar Parecer PRPPG

Descrição Sucinta: Este caso de uso representa o cadastramento do parecer da PRPPG sobre a solicitação de afastamento.

Tabela 4 Registrar Parecer PRPPG- Fluxo de Eventos Normais

Nome do Fluxo de Eventos Normal	Precondição	Descrição
---------------------------------	-------------	-----------

Registrar PRPPG	Parecer	1 – Status da solicitação estar marcado como: “Aprovada-CT”	1 - Secretário faz login no sistema, seleciona a solicitação desejada e registra o parecer da PRPPG com relação à solicitação de afastamento 2 - Secretário Faz upload do documento enviado pelo PRPPG referente ao parecer. 3- Caso o parecer seja positivo status da solicitação é alterado para “Aprovada-PRPPG”. 4 – Caso o parecer seja negativo o status da solicitação é alterado para “Reprovada”.
------------------------	----------------	---	---

Caso de Uso: Arquivar Solicitação

Descrição Sucinta: Este caso de uso arquiva uma solicitação de afastamento concluída.

Tabela 5 Arquivar Solicitação - Fluxo de Eventos Normais

Nome do Fluxo de Eventos Normal	Precondição	Descrição
Arquivar Solicitação	1 – Status da solicitação estar marcado como: “Aprovada-PRPPG” 2 – O Professor realiza o upload do relatório de sua viagem.	1- Secretário seleciona a solicitação desejada no sistema e altera seu status para “Arquivada”.

Descrição Sucinta: Este caso de uso cadastra uma nova solicitação de afastamento nacional de professor no sistema.

Tabela 6 Solicitar Afastamento Nacional - Fluxo de Eventos Normais

Nome do Fluxo de Eventos Normal	Precondição	Descrição
Solicitar Afastamento Nacional		1 - O professor preenche o formulário de solicitação de afastamento com as informações de: período de afastamento, período do evento, afastamento para evento no Brasil, o motivo de seu afastamento e o tipo ônus de sua viagem. 2 - O professor faz upload dos documentos necessários para realizar a solicitação. 3 - A solicitação é cadastrada no sistema e um e-mail automático é enviado a todos os professores do DI.
Aprovação Automática de solicitação		1 – Caso nenhum professor do DI se manifeste negativamente contra a solicitação de afastamento para evento realizado no Brasil até a data da reunião semanal do DI (ocorre toda segunda-feira as 17 horas) a solicitação é automaticamente aprovada.

		2 - Um e-mail automático é enviado ao professor solicitante informando o resultado final da solicitação.
--	--	--

Tabela 7 Solicitar Afastamento Internacional - Fluxo de Eventos Normais

Nome do Fluxo de Eventos Normal	Precondição	Descrição
Solicitar Afastamento Internacional		1 – O professor preenche o formulário de solicitação de afastamento com as informações de: período de afastamento, período do evento, afastamento para evento no exterior, o motivo de seu afastamento e o tipo ônus de sua viagem. 2 – O professor faz upload dos documentos necessários para realizar a solicitação. 3 – A solicitação é cadastrada no sistema e um e-mail automático é enviado ao chefe do departamento avisando da nova solicitação. 4 – O status da solicitação é alterado para: "Iniciada".
Encaminhar Solicitação de Afastamento Internacional		1 – Chefe do departamento encaminha a solicitação de afastamento para outro professor do DI que será o relator da solicitação de afastamento. 2 – O status da solicitação é alterado para: "Liberada". 3 – Um e-mail automático é enviado ao professor escolhido como relator informando que o parecer é necessário.
Gerar Ata de Aprovação de Solicitação	1 - Parecer da solicitação estar com status: "Aprovada-DI".	1 - É gerado um trecho de ata contendo o parecer do relator e as informações do afastamento do professor são enviadas para serem publicadas no Diário Oficial

Tabela 8 Solicitar Afastamento - Fluxo de Eventos Variantes

Nome do Fluxo de Eventos Normal Relacionado	Variante	Descrição
Análise de Solicitação (Exterior)	1 - Nem todos os documentos necessários para dar continuidade ao processo de solicitação foram anexados,	1 - Chefe de departamento rejeita a solicitação de afastamento e preenche um campo informando o motivo da rejeição 2 - Um e-mail é enviado ao professor que abriu a solicitação informando o parecer do chefe do departamento e as observações referentes ao parecer.

Descrição Sucinta: Este caso de uso cadastra no sistema o parecer de um

professor escolhido como relator de uma solicitação de afastamento.

Tabela 9 Deferir Parecer - Fluxo de Eventos Normais

Nome do Fluxo de Eventos Normal	Precondição	Descrição
Deferir Parecer Positivo	<p>1 – Solicitação de afastamento ser uma solicitação para evento no exterior.</p> <p>2 – Solicitação de afastamento estar marcada com status: “Liberada”</p>	<p>1 – O professor do DI escolhido como relator preenche um formulário onde informa o parecer favorável ao afastamento.</p> <p>2 – O parecer é cadastrado no sistema e um documento é gerado automaticamente contendo o parecer.</p> <p>3 – O status da solicitação é alterado para: “Aprovada-DI”.</p> <p>4 – Um e-mail automático é enviado ao professor solicitante informando do parecer.</p> <p>5 – Após ser marcada com o status “Aprovada-DI” a solicitação sai do escopo do SCAP onde é enviada para a aprovação do Centro Tecnológico e da PRPPG.</p>
Deferir Parecer Negativo		<p>1 – O professor relator preenche um formulário onde informa o parecer contrário ao afastamento.</p> <p>2 – O parecer é cadastrado no sistema e um documento é gerado automaticamente contendo o parecer.</p> <p>3 – O status da solicitação é alterado para: “Reprovada”.</p> <p>4 - Um e-mail automático é enviado ao professor solicitante informando do parecer.</p>
Manifestar-se contra Afastamento	<p>1- Solicitação de afastamento ter tipoAfastamento “Nacional”.</p> <p>2 – Solicitação de afastamento estar marcada com status: “Liberada”</p>	<p>1 – Algum professor do DI se manifesta contra uma solicitação de afastamento.</p> <p>2 – Após a reunião o secretário cadastra o resultado da votação no sistema alterando o status da solicitação para “Aprovada-DI” ou “Reprovada”.</p>

Tabela 10 Cancelar Solicitação - Fluxo de Eventos Normais

Nome do Fluxo de Eventos Normal	Precondição	Descrição
Cancelar Solicitação		<p>1 - O professor busca no sistema a solicitação que deseja cancelar.</p> <p>2 - Os dados da solicitação são apresentados e é solicitada confirmação do cancelamento.</p>

		<p>3 – Professor confirma que quer cancelar a solicitação de afastamento.</p> <p>4 - O status da solicitação é alterado para: "Cancelada"</p> <p>5 – Um e-mail é enviado ao chefe do Departamento avisando que a solicitação foi cancelada.</p>
--	--	---

4 Análise do SCAP

A atividade de Análise dentro do paradigma da Orientação a Objetos tem por objetivo identificar objetos do mundo real, características destes objetos e relacionamentos entre eles que são relevantes para o problema a ser resolvido, especificando e modelando o problema de forma que seja possível criar um projeto orientado a objetos efetivo (PRESSMAN, 2005).

Nesse capítulo apresenta-se a análise para o projeto de desenvolvimento do Sistema de Controle de Afastamento de Professores (SCAP) que foi proposto anteriormente. Nas próximas páginas serão apresentadas as classes de domínio que foram modeladas como resultado dessa fase.

4.1 Modelagem e Criação do Diagrama de Classes

A modelagem é uma atividade essencial no processo de engenharia dos requisitos e cuida da elaboração de modelos descrevendo o que o software tem de fazer (e não como fazê-lo). Até este momento, a ênfase está sobre o domínio do problema e não se deve pensar na solução técnica, computacional, a ser adotada (FALBO, 2011).

Na modelagem de um problema, tenta-se abstrair os conceitos do mundo real em classes de objetos, atributos e seus relacionamentos. Essas informações são essenciais para o mapeamento do problema a ser resolvido, são formados então diagramas de classes que mostra os atributos e associações entre as classes de domínio do sistema.

A seguir, na figura 4 são mostradas as classes de domínio que compõem o SCAP.

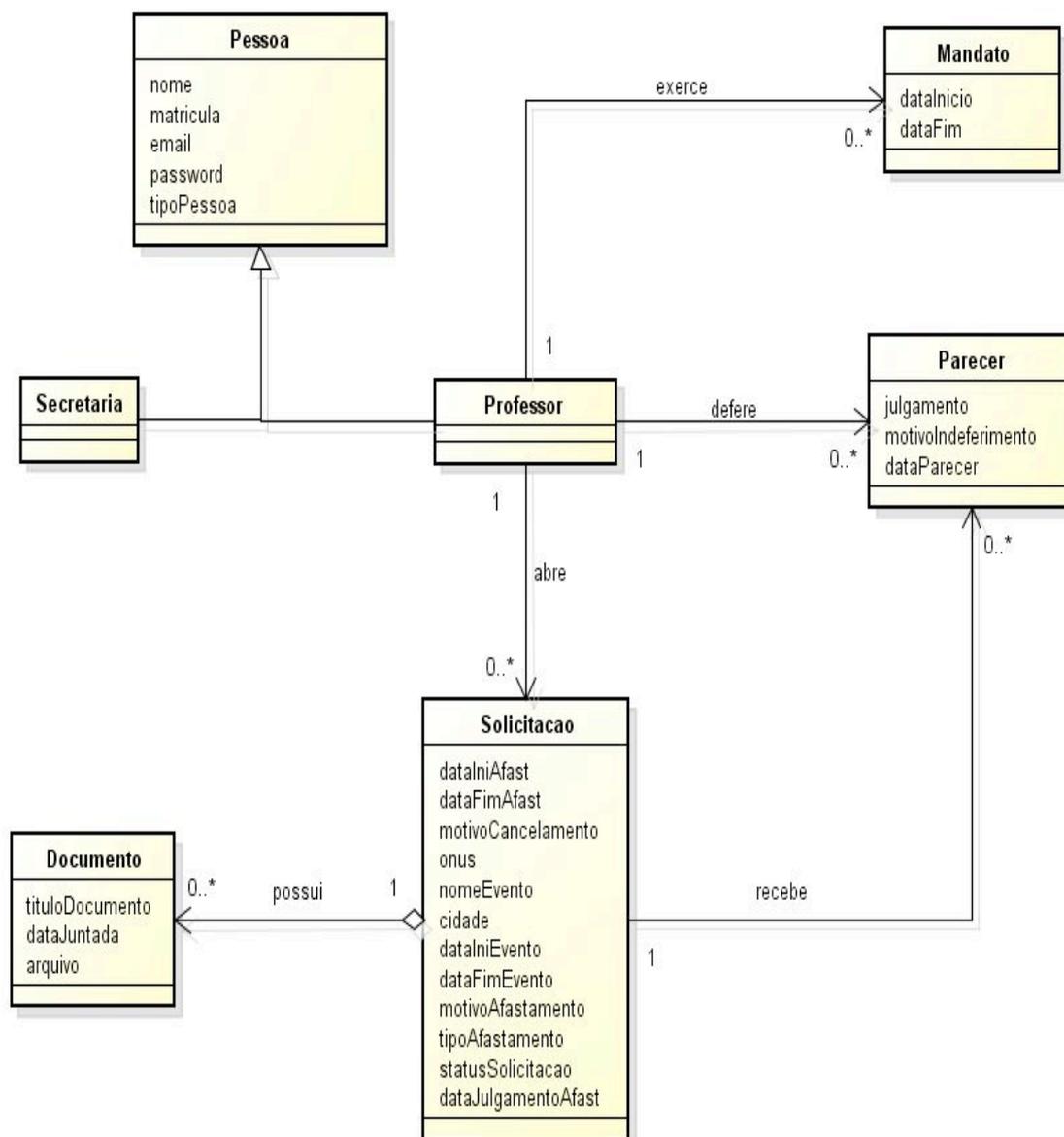


Figura 4 Diagrama de Classes do SCAP

A classe **Pessoa** representa todos os usuários cadastrados no sistema, dela herdam duas outras classes que representam os dois tipos de usuários do SCAP, as classes **Professor** e **Secretário** que representam respectivamente um Professor do Departamento de Informática e o Secretário do departamento.

Por fim, chefe de departamento do DI é representado pela associação entre as classes **Professor** e **Mandato**. Na classe **Solicitacao**, a principal classe do sistema, estão todas informações necessárias para que o documento de solicitação de afastamento seja gerado. Além do relacionamento

obvio com a classe **Professor**, **Solicitação** também se relaciona com as classes **Parecer**, que é a responsável por guardar as informações referentes a um parecer de um professor do DI sobre uma solicitação de Afastamento e **Documento** que é responsável por guardar o caminho, o título e a data de juntada, que é a data em que o documento é anexado à solicitação, de um documento ao processo de solicitação de afastamento.

4.2 Restrições de Integridade do SCAP

Como o próprio nome indica, restrições de integridade restringem as ações que o sistema ou seus usuários podem realizar. Elas complementam as informações de um modelo conceitual e capturam restrições relativas a relacionamentos entre elementos de um modelo que normalmente não são passíveis de serem capturadas pelas notações gráficas utilizadas na elaboração de modelos conceituais estruturais (FALBO, 2012).

Restrições de integridade também podem ser vistas como regras de negócio importantes para a compreensão e eliminação de ambiguidades do modelo conceitual, dessa forma, elas ajudam a tornar o modelo mais fiel à realidade. Elas podem ser relativas às multiplicidades presentes no modelo ou a atributos das classes do sistema (FALBO, 2012).

A seguir estão listadas as principais restrições de integridade capturadas através do modelo presente na figura 4.

1. Um professor não pode ser solicitado para dar um parecer sobre sua própria solicitação de afastamento.
2. A data de início de um afastamento não pode ser posterior a data de fim do mesmo afastamento.
3. A data de início de um mandato de professor não pode ser posterior a data de fim do mesmo mandato.
4. Não pode haver mais de dois professores (chefe e subchefe de departamento) exercendo um mandato ao mesmo tempo.
5. O secretário do departamento não pode abrir uma solicitação de Afastamento
6. Um usuário do SCAP não pode possuir os dois perfis de acesso (professor e secretario) ao mesmo tempo.

4.3 Dicionário de Dados

Um dicionário de dados consiste numa lista organizada de todos os elementos de dados que são pertinentes para o sistema. O dicionário de dados visa descrever o significado dos conceitos e atributos das classes de domínio capturadas nos diagramas de classes obtidos durante a fase de análise (FALBO, 2011).

A seguir temos o dicionário de dados do SCAP, onde descrevemos suas principais classes com seus respectivos atributos.

- **Pessoa:** representa qualquer usuário do SCAP. Os usuários podem ser um(a) secretário(a) e um professor.
 - nome: o nome completo do usuário;
 - matricula: representa a matrícula do usuário na universidade;
 - email: o e-mail institucional do usuário.
- **Solicitacao:** representa uma solicitação de afastamento realizada por um professor do DI.
 - dataIniAfast: representa a data de início do afastamento de um professor, ou seja, o primeiro dia útil em que ele não estará presente na UFES;
 - dataFimAfast: representa a data de fim do afastamento de um professor, ou seja, o ultimo dia útil em que ele não estará presente na UFES;
 - dataIniEvento: representa da data de início do evento para o qual o professor esta pedindo afastamento;
 - dataFimEvento: representa a data de fim do evento para o qual o professor esta pedindo afastamento;
 - nomeEvento: o nome do evento para o qual o professor esta pedindo o afastamento;
 - cidade: A cidade onde ocorrerá o evento mencionado acima;
 - onus: o tipo de ônus que o afastamento do professor trará para a universidade. Esse ônus pode ser total, parcial ou inexistente;
 - tipoAfastamento: o tipo de afastamento que o professor esta solicitando, esse afastamento pode ser nacional, para um evento

que ocorra no Brasil, ou internacional, para eventos ocorridos fora do país;

- statusSolicitacao: o status em que a solicitação de afastamento se encontra, esse status deve ir sendo atualizado a medida que a solicitação vai avançando no fluxo de aprovação. Os status que uma solicitação pode assumir são: Iniciada, Liberada, Bloqueada, Adequada, Cancelada, Aprovada-DI, Aprovada-CT, Aprovada-PRPPG, Arquivada e Reprovada;
- motivoCancelamento: caso uma solicitação de afastamento seja cancelada, o professor deve informar o motivo pelo qual esta solicitando o cancelamento (opcional).
- dataJulgamentoAfast: é a data em que a câmara departamental realiza seu julgamento sobre a solicitação de afastamento, aprovando-a ou reprovando-a.
- **Mandato:** representa o mandato de um professor do DI que esta atuando como chefe de departamento.
 - dataInicio: representa a data de início do mandato de um professor como chefe de departamento;
 - dataFim: representa a data do fim do mandato de um professor como chefe de departamento.
- **Parecer:** representa o parecer de um professor do DI, sobre uma solicitação de afastamento de realizada por outro professor e as principais informações relacionadas a esse parecer.
 - julgamento: representa o parecer de fato de um professor sobre a solicitação de afastamento de outro professor. Um julgamento pode ser favorável ou desfavorável à uma solicitação de afastamento;
 - dataParecer: a data do dia em que um professor desferiu um parecer sobre uma solicitação de afastamento;
 - motivoIndeferimento: o motivo pelo qual o professor deferiu um parecer.
- **Documento:** representa um documento a ser anexado a uma solicitação de afastamento e seus principais atributos

- tituloDocumento: o nome do documento;
- dataJuntada: a data em que o documento foi anexado à solicitação de afastamento;

arquivo: o arquivo binário do documento.

5 Projeto e Implementação do SCAP

Com os requisitos capturados e especificados na forma de modelos, pode-se começar a trabalhar no domínio da solução. Muitas soluções são possíveis para o mesmo conjunto de requisitos e elas são intrinsecamente ligadas a uma dada plataforma de implementação (linguagem de programação, mecanismo de persistência a ser adotado etc.).

A fase de projeto tem como um de seus objetivos definir e especificar uma solução a ser implementada. É uma fase de tomada de decisão, tendo em vista que muitas soluções são possíveis. Além disso, o projeto é um processo de refinamento. Inicia-se com o projeto da arquitetura do sistema, que visa descrever a estrutura de nível mais alto da aplicação, identificando seus principais elementos ou componentes e como eles se relacionam uns com os outros. Uma vez definida a arquitetura, o projeto passa a se concentrar no detalhamento de cada um desses elementos, até atingir o nível de unidades de implementação (ex., classes no desenvolvimento orientado a objetos) (FALBO, 2011).

O projeto pode então ser visto como o processo criativo de transformar uma especificação de um problema em uma especificação de uma solução. No projeto de software utilizam-se a especificação e os modelos de requisitos gerados na fase de análise e especificação de requisitos. A partir dos requisitos, muitas soluções são possíveis e, portanto, muitos projetos diferentes podem ser produzidos (FALBO, 2011). Uma solução é considerada adequada ao problema, se ela satisfizer a todos os requisitos especificados (PFLEEGER, 2004). Assim, o projeto é também uma atividade de tomada de decisão. Em suma, após ter analisado o problema, pode-se decidir como projetar uma solução.

O desenvolvimento desse sistema segue a aplicação do processo de software para sistemas Orientados a Objetos para a *Web* baseados em *frameworks* segundo o método *FrameWeb* (v. Seção 2.2).

A seção 5.1 desse trabalho detalha a Arquitetura de Sistema,

demonstrando as tecnologias utilizadas na implementação do SCAP e os pacotes criados para modularizar o sistema. A seção 5.2 apresenta uma descrição sucinta dos *frameworks* utilizados no desenvolvimento do SCAP. Já as seções 5.3, 5.4, 5.5 e 5.6 demonstram os modelos definidos pelo método *FrameWeb*, que são respectivamente: o modelo de domínio, o modelo de persistência, o modelo de aplicação e o modelo de navegação. Por fim, na seção 5.7 é demonstrado o funcionamento do SCAP.

5.1 Arquitetura do Sistema

A fase de projeto começa com a definição da arquitetura do sistema, que envolve dentre outras coisas, questões relativas à organização e estrutura geral do sistema, seleção de alternativas de projeto, atribuição de funcionalidades a elementos de projeto e atendimento a atributos de qualidade (MENDES, 2002). É nessa fase que os conceitos obtidos na fase de análise são mapeados para uma plataforma tecnológica de implementação, diminuindo assim os níveis de abstração elevados presentes na fase de análise e se aproximando do nível de uma linguagem de programação.

5.1.1 Tecnologia Utilizada

O SCAP foi implementado utilizando a linguagem de programação Java, na plataforma Java EE 7 (*Java Enterprise Edition 7*). A plataforma foi utilizada em conjunto com vários *frameworks* gratuitos e de código aberto que foram desenvolvidos para auxiliar no desenvolvimento de uma aplicação *Web* utilizando a linguagem Java. Para realizar a persistência dos dados foi utilizado um banco de dados relacional para a persistência de dados, já que esse é o modelo mais conhecido e utilizado atualmente, por possuir uma forte teoria por trás de sua operação e uma indústria consolidada que dá suporte a seus produtos.

A tabela 11 descreve as principais tecnologias utilizadas na implementação do SCAP:

Tabela 11 Tecnologias utilizadas na implementação do SCAP

Tecnologia	Descrição	Propósito
Java SE	Linguagem de programação orientada a objetos e plataforma de desenvolvimento JAVA.	Linguagem de programação utilizada na implementação do SCAP.
Java EE	Especificação da plataforma Java para a criação de aplicações distribuídas, incluindo a API de <i>Servlets</i> para o desenvolvimento <i>Web</i> .	Criação e implementação de páginas <i>Web</i> para o desenvolvimento de sistemas que utilizam a arquitetura cliente-servidor.
Oracle Glassfish 4 JEE Container	Servidor criado pela Oracle que implementa dentre outras coisas a API de <i>servlets</i> e permite a criação de aplicações <i>Web</i> baseadas em Java. O Glassfish 4 é totalmente compatível com a especificação Java EE 7.	Prover um servidor para as páginas <i>web</i> e ser compatível com a especificação Java EE 7.
Banco de Dados Relacional	Banco de Dados MySQL versão 5.5.	Permitir a persistência dos dados em tabelas.
Hibernate ORM	Framework gratuito de mapeamento que faz a persistência de objetos em um banco de dados relacional.	Realizar o mapeamento Objeto-Relacional e realizar a persistência de objetos.
CDI	Framework gratuito existente na especificação do Java EE 7 e presente desde o Java EE 6 para a injeção de contextos e dependências.	Realizar a injeção de dependências entre as classes do sistema afim de reduzir a quantidade de código utilizado.
JSF	Framework que permite a elaboração de interfaces com o usuário na plataforma <i>Web</i> .	Criar a interface com o usuário das páginas do SCAP.

PrimeFaces	Framework gratuito e de código aberto baseado no JSF que possui uma biblioteca maior de componentes gráficos para complementar o JSF.	Tornar mais rica a Interface com o usuário do SCAP através da maior quantidade de componentes gráficos oferecidos pelo PrimeFaces.
Jasper Reports	Framework para a criação de relatórios.	Criação de relatórios e documentos que serão exportados pelo sistema.

O *FrameWeb* foi originalmente criado para ser utilizado com o *framework* Struts², que é um *framework* gratuito mantido pela *Apache Software Foundation* e cujo propósito é atuar como um *framework* controlador frontal para facilitar o desenvolvimento de aplicações *Web*. O Struts² auxilia na adequação de aplicações feitas para *Web* ao padrão MVC além de prover outras funcionalidades como conversão de dados.

Nesse trabalho, é realizada a experiência de criar uma aplicação utilizando o método *FrameWeb* porém utilizando os *frameworks* presentes na especificação do Java EE 7. As informações referentes a essa experiência estão presentes no capítulo 6 desse trabalho.

5.1.2 Pacotes do SCAP

A figura 5 apresenta a divisão de pacotes do SCAP. A divisão foi feita primariamente baseado na arquitetura de camadas proposta pelo *FrameWeb*.

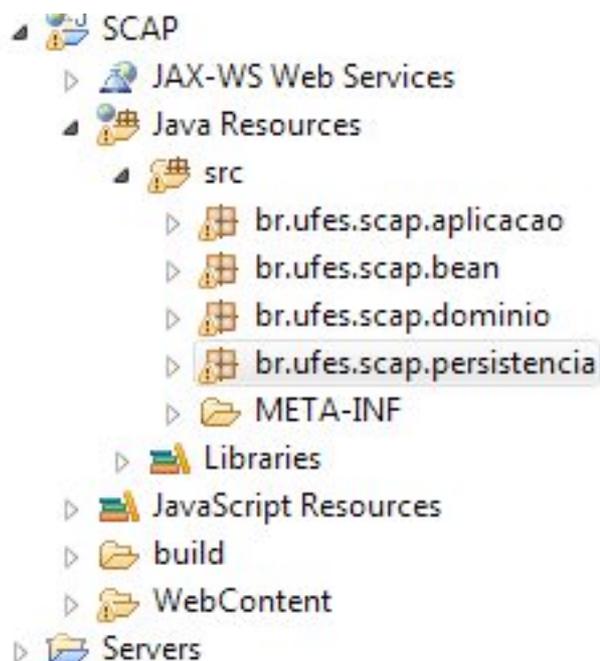


Figura 5 Pacotes do SCAP

A figura mostra os pacotes criados na implementação do SCAP. O pacote `br.ufes.scap.aplicacao` contém as classes de aplicação do SCAP, responsáveis por implementar a lógica de negócio da aplicação.

O pacote `br.ufes.scap.bean` possui as classes de controle do SCAP, que são responsáveis por tratar os estímulos criados pelo usuário e por realizar a comunicação entre o pacote de visão, que contém as páginas *Web* do sistema, localizadas na pasta `WebContent`.

O pacote `br.ufes.scap.dominio` contém as classes de domínio do sistema, que são as responsáveis por definir as principais entidades do sistema e seus respectivos atributos.

Por fim, o pacote `br.ufes.scap.persistencia` contém as classes responsáveis por gravar no banco de dados os objetos persistentes do sistema.

5.2. Frameworks Utilizados

Um *framework* é uma aplicação reutilizável e semi-completa que pode ser especializada para produzir aplicações personalizadas (HUSTED et al., 2004).

No desenvolvimento do SCAP foram utilizados vários frameworks visando à criação rápida de uma aplicação *Web* estável e eficiente.

Nas seções 5.2.1, 5.2.2, 5.2.3 e 5.2.4 são feitas descrições dos frameworks utilizados na implementação do SCAP.

5.2.1 Framework JSF

O JSF (*JavaServer Faces*) é um *framework* Web de interface com o usuário baseado no uso de componentes criados especificamente para aplicações Web Java. O JSF foi projetado para aliviar significativamente o ônus e manutenção de aplicações que são executadas em um servidor de aplicações Java (BURNS, 2013).

O JSF faz parte da especificação Java EE desde a versão 5, estando atualmente na sua versão 2.2, que foi lançada junto com a especificação Java EE 7. Neste trabalho ele foi utilizado como *framework* controlador frontal no lugar do Struts2, que é o *framework* original proposto pelo método *FrameWeb*.

5.2.2 Framework CDI

CDI (*Contexts and Dependency Injection*) é um *framework* de injeção de dependências que fornece um poderoso conjunto de serviços para componentes Java EE. Estes serviços permitem que os componentes Java EE, incluindo os *beans* de sessão EJB e JavaServer Faces (JSF) estejam vinculados a contextos do ciclo de vida, a serem injetados e de interagir de forma fracamente acoplada disparando e observando eventos. O CDI unifica e simplifica a especificação EJB (*Enterprise Java Beans*) (ORT, 2009).

O CDI faz parte da especificação Java EE desde a versão 6 e foi o *framework* utilizado para substituir o Spring que é o *framework* de injeção de dependências proposto pelo método *FrameWeb*.

5.2.3 Framework Hibernate

Hibernate é um *framework* de mapeamento objeto relacional escrito em linguagem Java cujo objetivo é diminuir a complexidade envolvida no desenvolvimento de aplicações que necessitam trabalhar com banco de dados relacional. O framework realiza a intermediação entre o banco de dados e sua aplicação, poupando o desenvolvedor de ter que se preocupar com instruções SQL para recuperar ou persistir os dados do seu software (BAUER & KING, 2005).

Como demonstrado anteriormente, o objetivo desse trabalho é a utilização do método *FrameWeb* para o desenvolvimento de uma aplicação utilizando os *frameworks* da especificação Java EE 7. Dessa forma, teríamos então que utilizar o padrão JPA (*Java Persistence API*), proposto pela especificação Java EE 7 e, em particular, a implementação incluída no servidor de aplicação Glassfish, que é o *framework* EclipseLink. No entanto, uma série de motivos nos levou a optar pelo uso do Hibernate no desenvolvimento do SCAP:

1. Restrições de tempo de conclusão desse trabalho;
2. Similaridade da implementação do Hibernate com o padrão JPA;
3. Ferramentas disponibilizadas pelo Hibernate como, por exemplo, a capacidade de utilização da linguagem SQL, que tornaram o desenvolvimento do SCAP mais ágil;
4. O fato desse trabalho ser focado no modelo de navegação e não no modelo de persistência, que seria o mais afetado no caso da substituição do *framework*.

Por fim, o uso de SQL para consultas no banco de dados ao invés da linguagem específica do Hibernate (HSQL) ou do JPA (JPQL) nos permitiu também colocar em prática conhecimentos adquiridos durante o curso na disciplina de Banco de Dados.

5.2.4 Framework PrimeFaces

PrimeFaces⁴ é um framework baseado na tecnologia JavaServer Faces que se foi criado com o objetivo de tornar mais simples o uso de elementos Ajax em aplicações *Web* desenvolvidas em Java.

O PrimeFaces conta com uma biblioteca de mais de 100 componentes e mais de 30 temas abertos que permitem a criação de interfaces ricas de forma simplificada e eficiente além de ter seus componentes construídos com a tecnologia Ajax por *default*, ou seja, não é necessário nenhum esforço do desenvolvedor para fazer chamadas assíncronas ao servidor, dessa forma, sendo possível a criação de validações do lado do cliente sem a necessidade

⁴ PrimeFaces, <http://www.primefaces.org>

de um recarregamento total da página.

5.3 Modelo de Domínio

O modelo de domínio é na verdade um diagrama de Classes representando os objetos do domínio do problema e seu mapeamento para que possa ser armazenado em um banco de dados relacional.

Por meio de um perfil UML definido pelo *FrameWeb*, o mapeamento da persistência é inserido no modelo conceitual, indicando assim como converter classes em tabelas e atributos em colunas.

O modelo de domínio do pacote é demonstrado na figura 6.

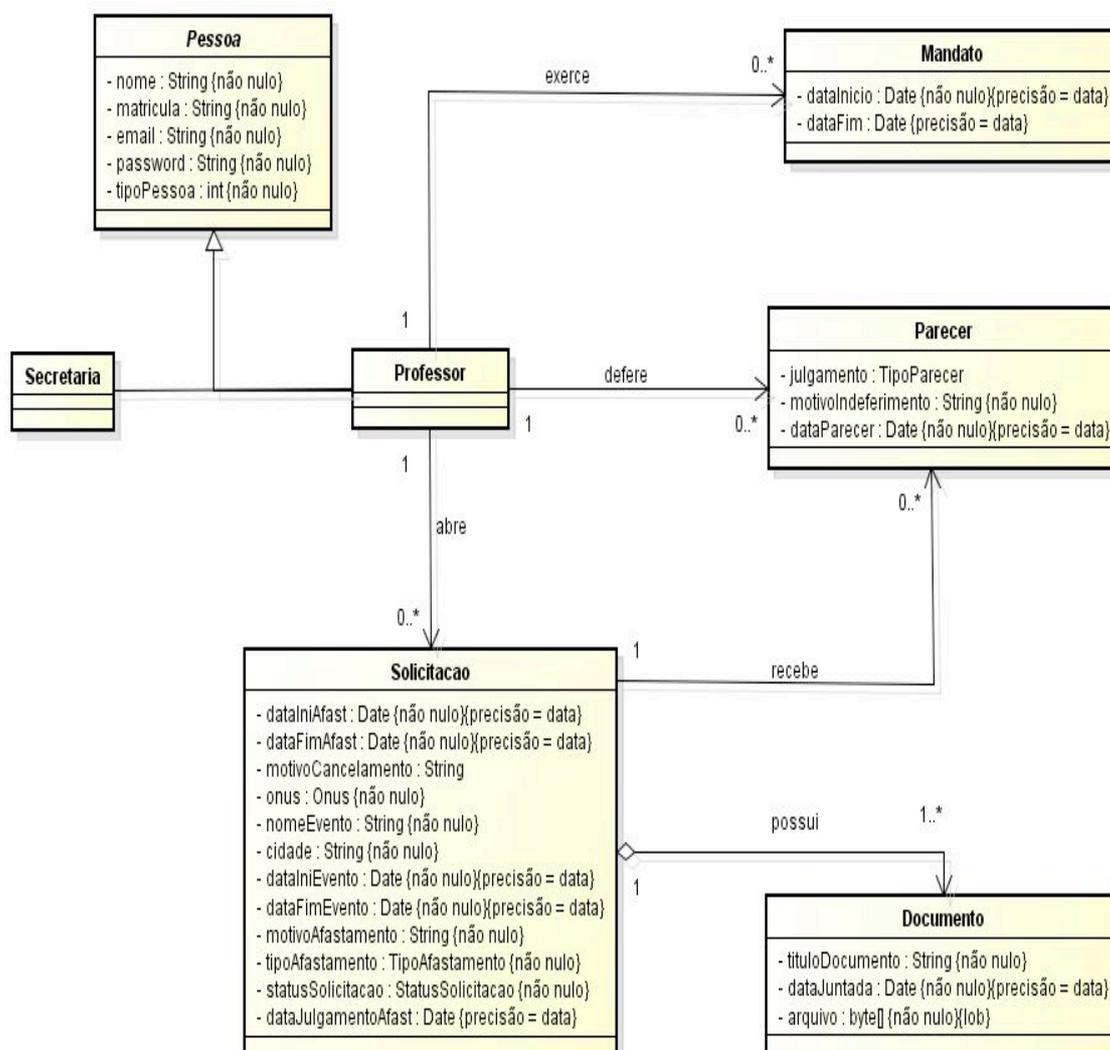


Figura 6 Modelo de Domínio do SCAP

A partir do diagrama de classes acima, foi construído o código para as classes da camada de domínio do SCAP e seus respectivos mapeamentos objetos/relacionais. É possível perceber que, diferentemente do diagrama de classes utilizado na fase de análise, nesse diagrama os tipos de dados definidos pelo Java foram adicionados aos atributos, as navegabilidades entre as classes foram determinadas e os nomes foram dados às navegabilidades.

A maioria dos atributos já recebe as configurações do mapeamento objeto/relacional: por exemplo, os atributos que possuem a restrição {não nulo} não podem possuir valores vazios (`null`) ao serem inseridos no banco de dados relacional. Os demais atributos, que não receberam a restrição, podem receber valores nulos ao serem inseridos no banco de dados. Os atributos `dataIniAfast`, `dataFimAfast`, `dataIniEvento`, `dataFimEvento`, `dataParecer`, `dataInicio`, `dataFim`, `dataJuntada`, `dataParecer` e `dataJulgamentoAfast` possuem a restrição {precisão = data}, pois por serem do tipo `Date`, armazenam data e hora, mas devido ao escopo do SCAP, só é de interesse a parte da data. Os atributos `onus`, `tipoAfastamento` e `statusSolicitacao` são tipos enumerados cujo os valores são demonstrados na figura 7.



Figura 7 Enumerados do SCAP

O tipo enumerado `StatusSolicitacao` apresenta os status pelos quais uma solicitação de afastamento passa até que seja concluída. O primeiro status de uma solicitação de afastamento é o status INICIADA, que é recebido após a solicitação ser cadastrada no sistema. O status INICIADA é válido tanto para solicitações de afastamento nacionais e internacionais. Caso a solicitação de afastamento seja de tipo NACIONAL, ou seja, para um evento que será realizado dentro do Brasil, a solicitação fica com status INICIADA até que algum professor do DI se manifeste contra ela, onde essa receberá o status REPROVADA e irá para votação na reunião de departamento, onde será

decido se essa deverá ser aprovada ou não e assim receber o status APROVADA-DI, que será cadastrado pelo secretário do DI.

No caso de uma solicitação de afastamento de tipo INTERNACIONAL, ou seja, para um evento que será realizado no exterior, a solicitação precisa passar pelo parecer do Chefe do Departamento, onde recebe o status LIBERADA. Após receber o status LIBERADA, a solicitação deve ser julgada por um professor do DI que emitirá seu parecer FAVORAVEL ou DESFAVORAVEL sobre a mesma. Caso o parecer seja DESFAVORAVEL a solicitação recebe o status REPROVADA e deverá ser julgada na reunião do DI, caso seja a solicitação seja julgada com parecer FAVORAVEL pelo professor relator, esta receberá o status APROVADA-DI e ficará aguardando os pareceres do CT e da PRPPG, pareceres esses que fogem ao escopo do sistema e devem ser cadastrados manualmente pelo secretário do DI após seu recebimento no formato de documento impresso.

Por fim, após o professor entregar o relatório de sua viagem, o secretário altera o status da solicitação para ARQUIVADA e essa tem seu ciclo de vida concluído. A qualquer momento durante o ciclo de vida da solicitação, o professor que registrou o pedido de afastamento pode cancelar a solicitação, ao realizar essa ação, a solicitação recebe o status CANCELADA.

Uma solicitação de afastamento possui um ônus à universidade associado a ela. Esse ônus pode ser TOTAL, que é quando o professor continua recebendo tudo o que lhe é de direito enquanto esta afastado, PARCIAL onde todos os benefícios são retirados com exceção do salário e INEXISTENTE onde não há ônus para universidade, ou seja, todos os benefícios do professor são suspensos no período em que este estiver afastado da universidade.

5.4 Modelo de Persistência

De acordo com o método *FrameWeb*, o padrão de projeto DAO (*Data Access Object*) deve ser utilizado para criar uma camada entre os objetos da camada de domínio e a tecnologia utilizada para realizar a persistência (no caso do SCAP, um *framework* de mapeamento O/R). As classes DAO são

responsáveis pela lógica de persistência de objetos de uma classe de domínio específica.

O modelo de persistência é um diagrama de classes que apresenta as classes DAO e suas respectivas interfaces que compõem a camada de persistência de uma aplicação. O diagrama da figura 8 apresenta para cada classe de domínio, uma interface com sufixo DAO de acesso a uma classe concreta com sufixo DAOImpl que realiza sua implementação. Vale acrescentar também que, todas as interfaces do pacote de persistência herdam de uma interface `BaseDAO` e que todas as classes do pacote herdam de uma classe `BaseDAOImpl` que implementa a interface `BaseDAO`.

A figura 8 apresenta as classes e interfaces do pacote de persistência.

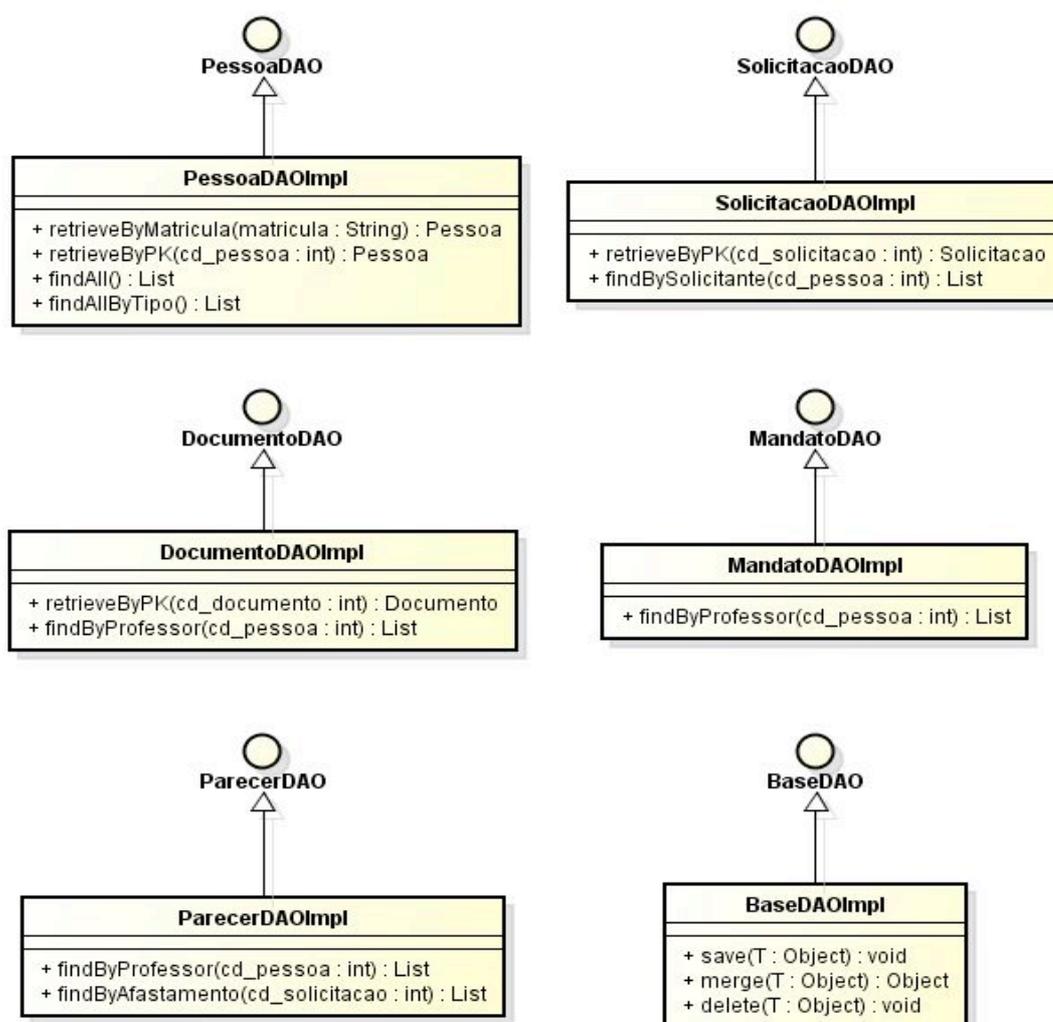


Figura 8 Modelo de Persistência do SCAP

O método *retrieveByMatricula()* da classe *PessoaDAOImpl* é utilizado para autenticar o usuário para que esse possa entrar no SCAP. A matrícula nesse caso é utilizada como *login* do usuário e é feita uma consulta no banco de dados verificando se aquela matrícula existe e se o *password* digitado é igual ao *password* cadastrado. Caso os dados digitados pelo usuário no formulário de *login* estejam corretos o usuário é autenticado e o método retorna os dados do usuário. O método *retrieveByPK()* é uma consulta que retorna um usuário através de sua *primary key* (chave primaria).

Os métodos *findAll()* e *findAllByTipo()* retornam uma lista de usuários através de uma consulta no banco de dados com a diferença que o método *findAllByTipo()* contém na sua clausula *where* uma condição que separa os usuários por seu tipo (Professor e secretário), esse método é utilizado entre outras coisas para preencher *combo boxes* (campo que exibe uma lista de valores pré-existentes) no sistema.

O método *retrieveByPK()* da classe *SolicitacaoDAOImpl* assim como o da classe *PessoaDAOImpl* é uma consulta que encontra uma solicitação de afastamento através de sua chave primaria e a retorna. O mesmo acontece para os métodos de mesmo nome presentes nas classes *DocumentoDAOImpl* e *MandatoDAOImpl*. O método *findBySolicitante()* por sua vez, realiza a consulta no banco e retorna uma lista das solicitações de afastamento de um professor. A consulta é realizada através da *foreign key* (chave estrangeira) de professor que existe dentro de uma solicitação de afastamento. Os métodos *retrieveByPK()* e *findBySolicitante()* da classe *SolicitacaoDAOImpl* são utilizados pelos casos de uso Deferir Parecer, Consultar Solicitação e Manifestar-se contra Afastamento para que a solicitação desejada possa ser encontrada e exibida na tela.

Os métodos *findByProfessor()* presentes nas classes *DocumentoDAOImpl*, *MandatoDAOImpl* e *ParecerDAOImpl* retornam respectivamente uma lista de todos os documentos, mandatos e pareceres associados a um professor.

Assim como o método `findBySolicitante()` da classe `SolicitacaoDAOImpl`, esses métodos realizam a consulta através da chave estrangeira de professor que existe dentro das tabelas Documento e Mandato.

Por fim, os métodos genéricos *save*, *merge* e *delete* presentes na classe `BaseDAOImpl` servem para realizar o cadastro, alteração e exclusão dos objetos presentes no sistema.

5.5 Modelo de Navegação

O modelo de navegação apresenta as páginas *Web* e os demais itens que compõem (formulários, imagens, links etc.) a camada de visão de um sistema que interagem com a camada de controle para enviar os estímulos e ações criados pelo usuário até a camada de serviço.

Nessa subseção são apresentados modelos de navegação para os casos de uso mais relevantes do SCAP, esses casos de uso foram escolhidos por representarem bem os aspectos do modelo proposto pelo método *FrameWeb*.

FrameWeb define na tabela 12 os estereótipos UML utilizados pelos diferentes elementos que podem ser representados no modelo de navegação.

Tabela 12 Tabela de Estereótipos UML utilizados pelo Modelo de Navegação (SOUZA, 2007)

Estereótipo	O que representa
nenhum	Uma classe para qual o framework que realiza o papel de Controlador Frontal delega a execução da ação feita pelo usuário.

<<pagina>>	Uma página web estática ou dinâmica.
<<modelo>>	Um <i>template</i> de uma página <i>web</i> processada por uma <i>engine</i> que retorna um documento <i>html</i> como resultado.
<<form>>	Um formulário.

A figura 9 apresenta o modelo de navegação para o caso de uso Cadastrar Professor.

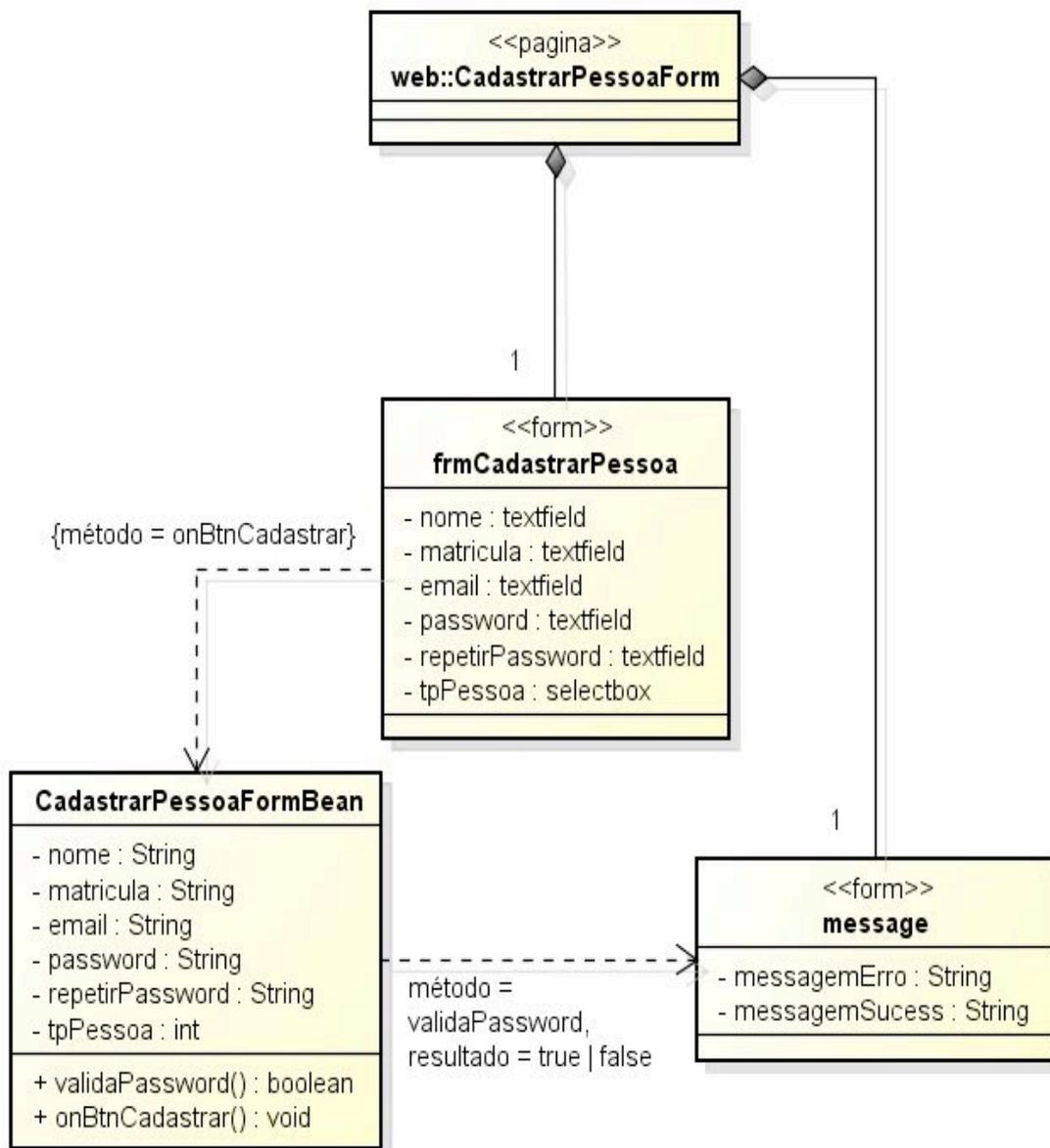


Figura 9 Modelo de Navegação para Cadastrar Usuário

O diagrama acima apresenta os elementos que são relevantes para a execução do cadastro de um novo usuário no SCAP. O diagrama porém não contempla a ação de cadastrar um chefe de departamento visto que esse precisa estar previamente cadastrado como professor no sistema.

Cadastrar Usuário: Na página `CadastrarPessoaForm` será exibido o <<form>> `frmCadastrarPessoa` onde um usuário previamente cadastrado deverá preencher todos os campos do formulário e deverá clicar no botão Cadastrar exibido na tela, ao clicar no botão os dados são enviados para a classe `CadastrarPessoaFormBean` que executará o método `onBtnCadastrar()`, os dois *passwords* digitados pelo usuário serão validados e uma mensagem de sucesso será exibida ao usuário.

A figura 10 exibe o Modelo de Navegação da primeira parte do caso de uso mais importante do sistema, que é o de Solicitar Afastamento. Essa primeira parte é referente ao momento no qual o professor que deseja solicitar o afastamento cadastra a solicitação inserindo as informações necessárias no formulário presente na página `CadastrarSolicitacaoForm` e clicando no botão Cadastrar.

As informações são enviadas para a classe `CadastrarSolicitacaoFormBean` é essa é responsável por tratar as informações recebidas do formulário e envia-las para a camada de serviço na forma de um Objeto do tipo `Solicitacao`, onde por sua vez, será enviada a camada de persistência e persistida.

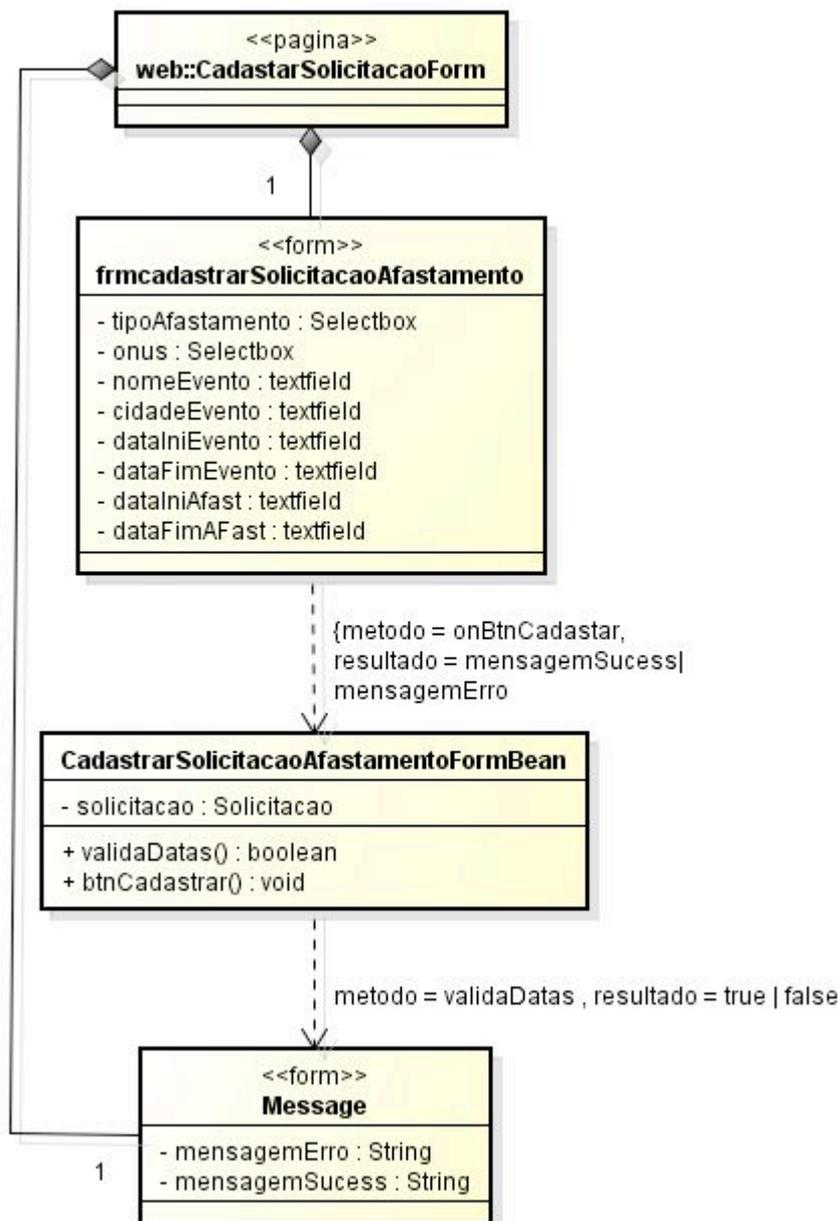


Figura 10 Modelo de Navegação para Cadastrar Solicitacao

Os dois modelos de navegação apresentados nessa seção seguem as recomendações propostas pelo método FrameWeb. No entanto, devido a diferença dos *frameworks* utilizados na implementação do SCAP com os propostos originalmente no método, algumas funcionalidades presentes na página, como eventos disparados através de AJAX (*Asynchronous JavaScript And XML*), que permitem alterações em de apenas parte da página, não podem ser demonstradas no modelo. Dessa forma, na próxima seção desse trabalho, iremos propor melhorias ao método, para inclusão de elementos no modelo de navegação para que esse seja capaz de representar todos os elementos

utilizados na criação das páginas.

5.6 Modelo de Aplicação

O modelo de aplicação é o responsável por reunir as classes que implementam a lógica de negócio do sistema, descrita nos casos de uso e suas devidas dependências com relação aos demais modelos do sistema.

Basicamente, o modelo de aplicação mostra as classes da camada de aplicação (classes presentes no pacote `br.ufes.scap.aplicacao`). O modelo também mostra as relações entre as classes da camada de aplicação com as classes presentes na camada de controle (pacote `br.ufes.scap.bean`), classes essas que dependem das classes de aplicação e as classes de persistência (pacote `br.ufes.scap.persistencia`) que são classes das quais as classes de aplicação dependem para persistir/recuperar registros do banco de dados relacional.

Segundo o método *FrameWeb*, tais dependências entre diferentes camadas da arquitetura são satisfeitas por um *framework* de Injeção de Dependências e o modelo de aplicação pode servir de base para a configuração deste *framework*.

A figura 11 apresenta o modelo de aplicação do SCAP. Ela mostra as classes da camada de Lógica de Aplicação (classes com sufixo *Service*) do SCAP. Foram mapeadas classes de aplicação para as classes mapeadas no modelo de Domínio.

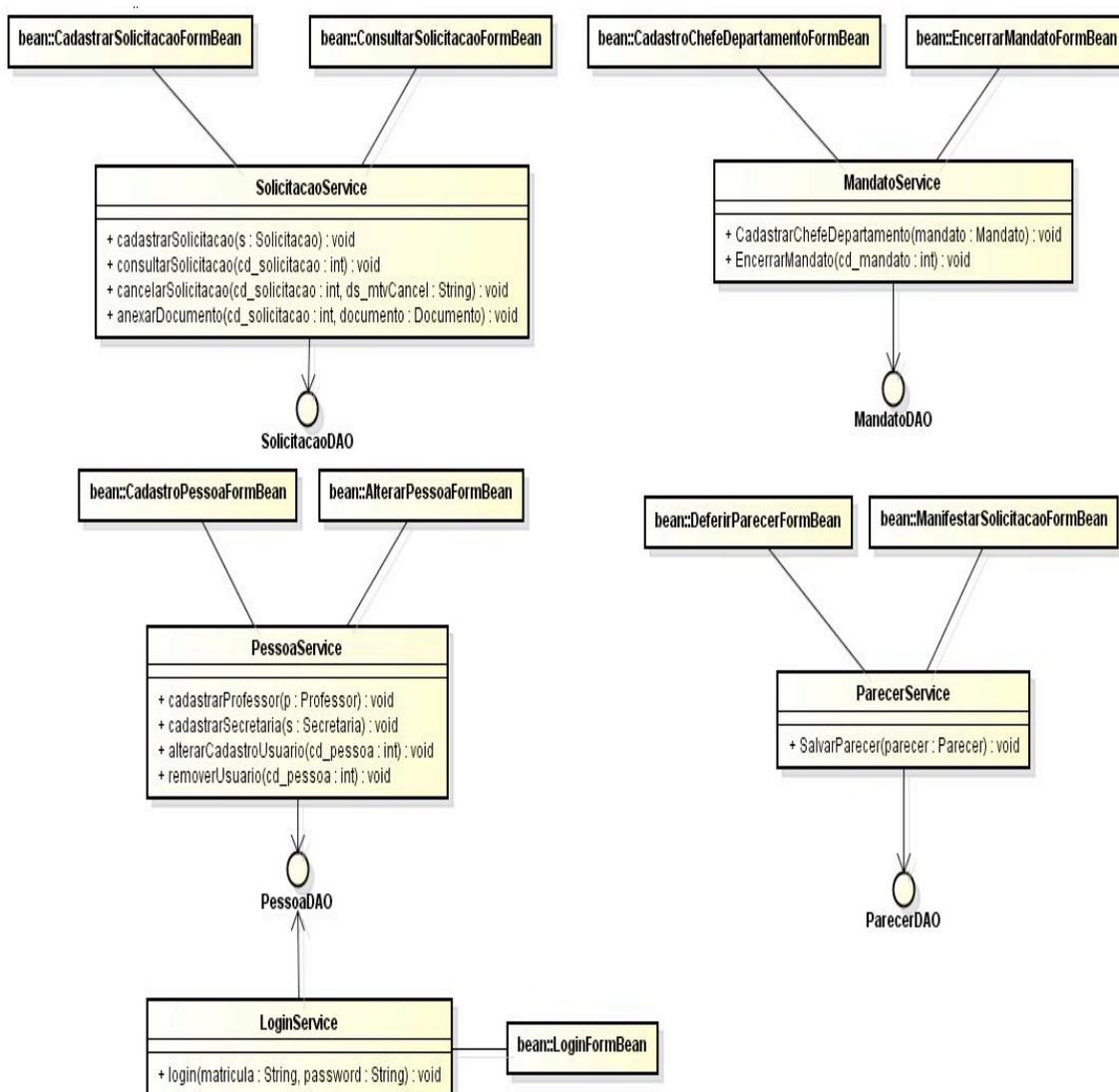


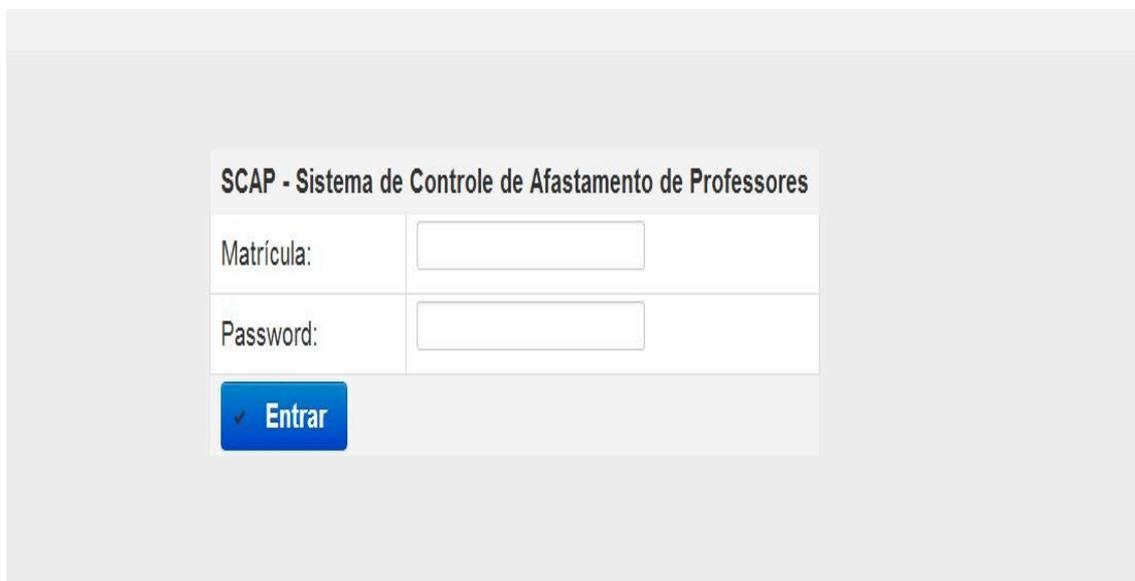
Figura 11 Modelo de Aplicação do SCAP

O modelo também mostra as dependências das classes de aplicação. As classes de controle (sufixo *Bean*) dependem das classes de aplicação e essas por sua vez, dependem das classe com sufixo DAO (representadas no modelo pelas interfaces que as definem) que implementam as interfaces mostradas no modelo para persistir, atualizar, recuperar ou remover (CRUD) os objetos que são afetados pelos estímulos gerados pelo usuário do SCAP.

5.7 Implementação do SCAP

Essa seção visa demonstrar o funcionamento do SCAP, e mostrar as suas principais funcionalidades.

Ao acessar a o endereço do SCAP, a tela de login, presente na figura 12, será exibida para que o usuário possa ser autenticado, quando ao realizar a autenticação o usuário é redirecionado para a tela principal do SCAP.



A imagem mostra a interface de login do sistema SCAP. No topo, há o título "SCAP - Sistema de Controle de Afastamento de Professores". Abaixo dele, há um formulário com dois campos de entrada: "Matrícula:" e "Password:". Cada campo possui um botão de seta para a direita. Abaixo dos campos, há um botão azul com o texto "Entrar" e um ícone de seta para a direita.

Figura 12 Tela de Login do SCAP

Ao se acessar o endereço do SCAP, a tela de login será exibida para o usuário e ele deverá entrar com sua matrícula e senha para que possa ser autenticado e entrar no sistema. A autenticação é muito importante pois, com ela são salvos em um objeto do tipo Pessoa as informações do usuário e essas são utilizadas por outras telas do sistema, como por exemplo na tela de cadastro de solicitação de afastamento, que se utiliza das informações da pessoa que esta autenticada no sistema para preencher as informações do professor que realiza a solicitação de afastamento. A figura 13 demonstra a tela cadastro de solicitação de afastamento.

SCAP Cadastros ▾ Secretaria ▾ Afastamentos ▾

Nome do Evento:*

Data de Início do Evento:*

Data de Fim do Evento:*

Cidade:*

Data de Início do Afastamento:*

Data de Fim do Afastamento:*

Tipo de Afastamento:* Seleccione ▾

Onus:* Seleccione ▾

Cadastrar Solicitação

Figura 13 Tela de Cadastro de Solicitação de Afastamento.

Todas as telas de cadastro do SCAP seguem o mesmo padrão da tela de cadastro de solicitação de afastamento mostrada acima.

Para cadastrar uma solicitação de afastamento, o usuário, que deverá obrigatoriamente ser um professor do Departamento de Informática, deverá preencher todos os campos presentes na tela e clicar no botão “Cadastrar Solicitação”. Então, o número da solicitação de afastamento será informado ao usuário. Caso algum dos campos não seja preenchido ou algum valor inconsistente seja inserido, como por exemplo, a data de fim de um afastamento ser anterior à data de início ou ainda anterior a data de início do evento uma mensagem de erro será exibida para o usuário.

A seguir, a figura 14 mostra a tela onde um professor pode buscar por uma solicitação de afastamento e manifestar-se contra ela.

SCAP Cadastros ▾ Secretaria ▾ Afastamentos ▾

Codigo da Solicitação:

Solicitante:

Pesquisar Solicitação

Afastamentos

Codigo	Tipo de Afastamento	Solicitante	Inicio	Fim	Status	Onus
1	NACIONAL	Bruno Borfini	2014-01-01	2014-01-02	INICIADA	PARCIAL

Manifestar-se

Figura 14 Tela de Manifestar-se Contra Afastamento

Nessa tela o professor realiza uma busca por uma solicitação através de seu código e essa é exibida na tabela mostrada na figura. Ao clicar no botão detalhes uma caixa de dialogo se abre contendo todas as outras informações da solicitação e uma caixa de texto onde o professor registrará seu parecer em relação à solicitação de afastamento. Esse parecer será julgado por todos os professores na próxima reunião de departamento e seu resultado final será registrado através de outra tela do SCAP pelo secretário do departamento.

A tela também tem um filtro para buscar pelo solicitante. Nesse caso a tabela é preenchida com todas as solicitações abertas por aquele solicitante. A tela de Consulta de solicitação é muito parecida com a tela exibida acima, porém não possui a opção de inserir o motivo de manifestação.

A figura 15 mostra a caixa de dialogo aberta ao clicar no botão Manifestar-se.

Manifestar-se Contra Afastamento x

Codigo: 1

Solicitante: Bruno Borlini

Tipo de Afastamento: NACIONAL

Inicio: 2014-01-01

Fim: 2014-01-02

Status: INICIADA

Onus: PARCIAL

Motivo:

1003 characters remaining.

Enviar Parecer

Figura 15 Caixa de Dialogo Registrar Manifestação

A caixa de dialogo mostra alguns detalhes da solicitação de afastamento e possui um campo texto para que o professor possa registrar o motivo de sua manifestação contra outro professor. Ao clicar no botão enviar parecer uma verificação é feita, caso a solicitação seja do tipo Internacional, não esteja com status Iniciada ou Liberada, ou o campo Motivo não tenha sido preenchido, uma mensagem de erro é exibida e a manifestação não é cadastrada no sistema.

6 Propostas de Melhorias para o Método FrameWeb

Esse capítulo descreve a experiência de se desenvolver uma aplicação utilizando o método *FrameWeb*, porém utilizando *frameworks* presentes na especificação do Java EE7 e sugere alterações nos modelos propostos pelo método para melhor adaptação a diferentes tipos de *frameworks*.

O capítulo também apresenta uma breve descrição dos *frameworks* utilizados no desenvolvimento do SCAP.

6.1 Propostas de Melhorias para o Método FrameWeb a partir de experiências com diferentes tipos de Frameworks

Na fase de projeto, o método *FrameWeb* sugere alguns modelos a serem construídos para facilitar na criação e implementação de uma aplicação, esses modelos são: Modelo de Domínio, Modelo de Persistência, Modelo de Aplicação, Modelo de Navegação. Alguns desses modelos foram demonstrados no capítulo 5 quando descrevemos o projeto de implementação do SCAP.

Após o experimento de criação do SCAP utilizando o método *FrameWeb* utilizando os *frameworks* da especificação Java EE 7 concluímos que somente o modelo de navegação que receberá alterações a fim de melhorar sua adaptação a diferentes *frameworks Web*. Segundo o método *FrameWeb* (SOUZA, 2007), existe uma correspondência entre os *frameworks* controlador frontal e o Modelo de Navegação. Dessa forma, esse modelo é o que deve sofrer alterações.

A figura 16 exibe o modelo de navegação da classe `CadastroSolicitacaoFormBean` com as alterações sugeridas ao método.

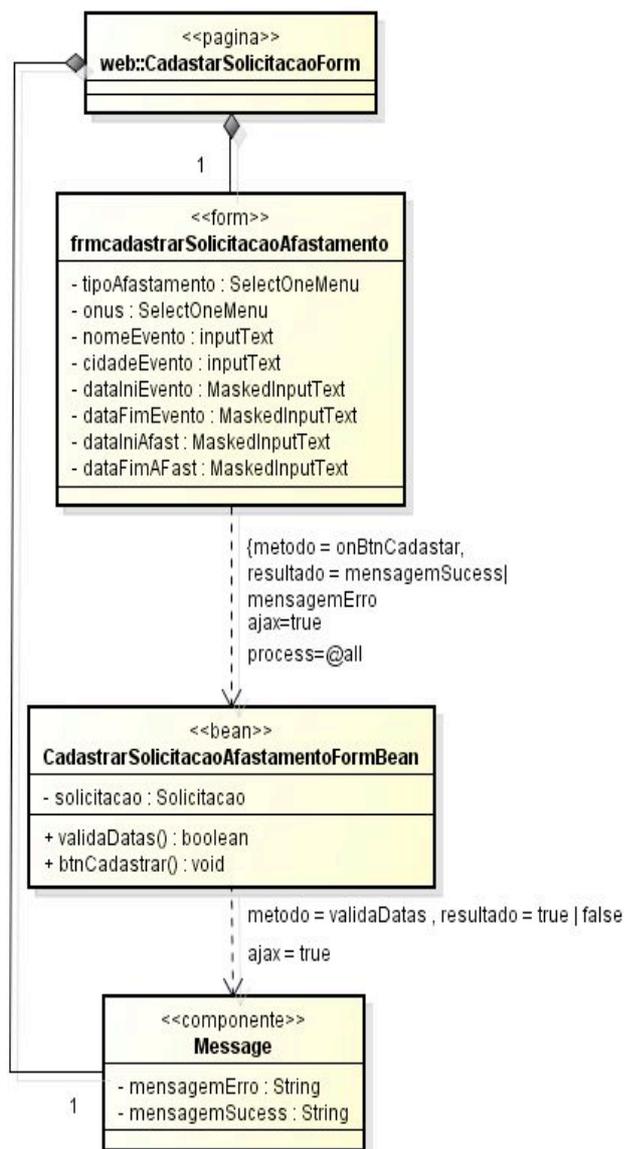


Figura 16 Modelo de Navegação com melhorias propostas

Conforme pode ser visto na figura 16, foram adicionados dois novos estereótipos UML. O estereótipo `<<bean>>` representa a classe controladora utilizada pelo JSF. Essa, é uma classe que é responsável por tratar os formulários presentes nas paginas com extensão `.xhtml` do JSF e suas instâncias são preenchidas com as informações obtidas nos formulários em questão.

O estereótipo `<<componente>>` representa um elemento da página que sofre a ação de algum outro componente. No caso da implementação do SCAP, o AJAX permite o carregamento de apenas uma parte da página. Mensagens de validação são carregadas automaticamente através de uma requisição AJAX que é executada quando o usuário realiza uma requisição ou gera um estímulo. Esse tipo requisição, que carrega dados em *background*, sem a necessidade de um carregamento completo da página só é possível devido ao fato dos componentes do PrimeFaces terem sido construídos com a utilização do AJAX, que está disponível em praticamente todos os componentes do PrimeFaces e pode ser desabilitado através da declaração do atributo `ajax="false"` no componente.

Também foram adicionados as notações `ajax=true`, para determinar a utilização do AJAX no momento em que há uma validação do lado do cliente a ser feita e essa gera um estímulo para o usuário sem a necessidade de um recarregamento da página. A notação `process=@all` determina que todo o conteúdo do formulário está sendo processado. Essa notação é utilizada, pois o PrimeFaces permite que apenas um componente específico, ou uma lista de componentes, nomeados por meio de seus identificadores, sejam processados quando o usuário gera um estímulo na página.

Os atributos da classe controladora representam parâmetros de entrada e saída de dados relacionados à ação executada pelo usuário (os dados são recuperados ou enviados à classe controladora através dos métodos `get` e `set` de cada atributo da classe).

Quando um atributo da classe controladora esta sendo referenciado por algum elemento do formulário presente na página `.xhtml` e o usuário entra com um dado no formulário e executa uma ação de envio dos dados (que pode ser através de um botão por exemplo), os dados do formulário são injetados automaticamente pelo *framework* na classe controladora, através do método `set` do atributo e ficam disponíveis dentro da classe para serem manipulados.

De forma análoga, quando o usuário executa uma ação para recuperar um dado presente no atributo da classe controladora (o dado em questão provavelmente estará persistido no banco de dados e deverá ser trazido da camada de persistência até a classe controladora da tela que recebeu o

estímulo vindo do usuário) e exibi-lo em um componente do formulário, o *framework* age injetando a informação presente no atributo da classe controladora no componente do usuário através do método *get* do atributo.

Vale ressaltar que a ligação da página *Web* com a classe controladora é feita através da anotação `@ManagedBean` do *framework* JSF mas também poderia ser feita através da anotação `@Named` do CDI.

Por fim, é importante notar que não é objetivo principal deste trabalho propor como o método *FrameWeb* deve ser evoluído, mas somente mostrar a necessidade de evolução, indicando algumas das diferenças entre o *framework* utilizado originalmente na proposta do método e o padrão JSF utilizado no desenvolvimento do SCAP. Encontra-se em curso uma pesquisa de mestrado com o objetivo de propor uma evolução do *FrameWeb* e o resultado deste trabalho poderá servir de insumo para esta pesquisa.

7 Considerações Finais

Nesse último capítulo, são apresentadas as conclusões obtidas durante a análise, projeto e desenvolvimento desse trabalho e as perspectivas para possíveis trabalhos futuros.

7.1 Conclusões

O uso de *frameworks* que provém uma infraestrutura base para o desenvolvimento de Sistemas de Informação *Web* (*Web-based Information Systems – WISs*), com o passar do tempo, influenciou a definição de padrões para o desenvolvimento de aplicações. Seu uso auxilia o desenvolvedor a construir um software mais rapidamente (muitos componentes já estão prontos) e com maior segurança (os *frameworks* já foram extensivamente testados por seus desenvolvedores) (SOUZA, 2007).

Visando aproveitar a consolidação de algumas categorias de *frameworks* e aumentar a produtividade no desenvolvimento de aplicações *Web*, SOUZA, (2007) propõe o método *FrameWeb* (*Framework-based Design Method for Web Engineering*). O método assume que determinados *frameworks* serão utilizados durante a construção da aplicação, define uma arquitetura básica para o WIS e propõe modelos de projeto que se aproximam da implementação do sistema usando esses *frameworks*.

Nesse trabalho, aplicou-se o processo proposto pelo método *FrameWeb* no desenvolvimento do SCAP, utilizando os *frameworks* presentes na especificação do Java EE 7, ao invés de utilizar os *frameworks* propostos originalmente pelo método (Struts², Spring *Framework*, Hibernate e Sitemesh).

Para o estudo e compreensão dos *frameworks* utilizados (CDI, Hibernate, JSF, Primefaces) durante o desenvolvimento do SCAP, foram realizadas pesquisas, buscas de exemplos e tutoriais e a leitura da documentação dos *frameworks*.

A partir do desenvolvimento do projeto e da experiência adquirida durante a implementação do mesmo, foi possível realizar uma avaliação do método *FrameWeb*. Os estudos e dificuldades encontradas durante a

realização desse trabalho conferiram ao autor além de um amadurecimento profissional, acadêmico e pessoal, uma maior proficiência no desenvolvimento de sistemas e consolidou bem os conhecimentos obtidos durante todo o curso de Ciência da Computação. A análise e projeto do SCAP também desenvolveram no autor a vontade de se aprofundar mais na área da Engenharia de Software, e se possível, realizar um mestrado nessa área.

O protótipo implementado do SCAP pode ser considerado simples e escalável, ele contém uma boa parte das funcionalidades já devidamente implementadas mesmo com o tempo curto para a finalização do projeto. Grande parte do desenvolvimento do SCAP se deve ao uso do *FrameWeb* que realmente auxiliou no processo de desenvolvimento através dos modelos de projeto e do perfil UML propostos pelo método que se aproximam da fase de implementação por possuírem características de certos tipos de frameworks que foram utilizados durante o processo de desenvolvimento. Dessa forma é possível acrescentar que apesar das várias dificuldades encontradas durante o processo de codificação, como por exemplo, na parte de aprendizado da utilização do CDI, que é um framework cujo autor desse trabalho ainda não tinha tido contato, os processos de projeto e desenvolvimento e da aplicação podem ser considerados satisfatórios, pois atenderam e até superaram as expectativas do autor em alguns casos, se considerarmos o pouco tempo disponível para a conclusão do projeto.

Após o processo de implementação, foram sugeridos melhorias para o método *FrameWeb*, mais precisamente para o Modelo de Navegação, para que o método se torne independente dos *frameworks* de desenvolvimento utilizados.

Por fim é preciso adicionar que esse trabalho visa mostrar que o *FrameWeb* pode ser evoluído e que esse trabalho pode ser utilizado como base para esse processo de evolução, principalmente devido ao fato de ser uma experiência diferenciada de utilização do método

7.2 Trabalhos futuros

Os trabalhos futuros com relação ao tema estudado podem ser:

- Incorporar melhorias ao SCAP, adicionando mais funcionalidades que possam ser utilizadas pelo Departamento de Informática;
- Analisar o método *FrameWeb* utilizando o padrão JPA (*Java Persistence API*) para realizar a persistência dos dados e verificar o impacto que a alteração do *framework* de persistência traz ao método e a seus modelos.

Referências

BAUER, C.; KING, G.; **Hibernate in Action**. 1st Edition. Manning Publications, 2005.

BEEVAN, N. **Usability issues in Web Design**. Proceedings of Usability Professionals Association Conference, Washington DC, EUA. 22-26 June 1998. Disponível online: <http://www.nigelbevan.com/cart.htm>. Último acesso: 06/02/2014.

BEZERRA, E. **Desenvolvimento Incremental e Iterativo**. Artigo disponível online em:

<http://www.mundooo.com.br/php/mooartigos.php?pa=showpage&pid=20>.2004.

Último acesso em 14/01/2014

BORGES, C.B.; MORALES, I.; RODRIGUEZ, N.J.; **Guidelines for Design usable World Web Pages**. CHI '96 Conference Companion on Human Factors in Computing Systems, New York, EUA. 1996. Disponível online: <http://www.cc.gatech.edu/projects/PageSleuth/references/p277-borges.pdf>. Último acesso: 06/02/2014

BURNS, E. **Java Server Faces** Specification v2.2, Oracle America, Inc. Disponível online: <https://jcp.org/en/jsr/detail?id=344>, March 2013. Último acesso: 06/02/2014

CODA, F.; GHEZZI, C.; VIGNA, G.; GARZOTTO, F.; **Towards a Software engineering approach to Web site development Software Specification and Desing**. Proceedings in the Ninth International Workshop on, 16-18 April 1998.

DEMICHIEL, L.; SHANNON, B. **Java™ Platform**, Enterprise Edition (Java EE) Specification, v7. Disponível online: <http://jcp.org/en/jsr/detail?id=342>, Maio 2013. Último acesso: 06/02/2014

DESHPANDE, Y.; MURUGESAN, S.; HANSEN, S.; GINIGE, A. **Web Engineering: A New Discipline for Development of Web-Based Systems**, Web Engineering: Managing Diversity and Complexity of Web Application Development, Springer Berlin Heidelberg, 1st Edition. 2001. Disponível Online: <http://www-itec.uni-klu.ac.at/~harald/proseminar/web11.pdf>. Último acesso: 06/02/2014

- FALBO, R. A. **Projeto de Sistemas**: Notas de aula. Disponível online:
http://www.inf.ufes.br/~falbo/files/Notas_Aula_Projeto_Sistemas.pdf.
 Setembro 2011. Último acesso: 12/01/2014.
- GININE, A. **Web Engineering**: Managing the Complexity of Web Systems Development. IEEE Multimedia. Volume 8. Issue 1 Jan-Mar 2002.
- GINIGE, A.; MURUGESAN, S. **Web Engineering**: An Introduction. IEEE Multimedia. Volume: 8, Issue: 1. Jan-Mar 2001.
- HUSTED, T.; DUMOLIN, C.; FRANCISCUS, G; INTERFELDT, D. **Struts em ação**. 1ª Edição. Editora Ciência Moderna Ltda, Rio de Janeiro, 2004.
- KOTONYA, G., SOMMERVILLE, I., **Requirements engineering**: processes and techniques. Chichester, England: John Wiley, 1998.
- LI, H.; SHI, T.; YANG, S. **An approach of sampling computing for wavelet analysis and its Web-based implementation**. Signal Processing Proceedings, 2000. WCCC-ICSP 2000. 5th International Conference on 21-25 Aug. 2000 volume 1
- LOWE, D. **Website evaluation**. Proceedings of WebNet World Conference on the WWW and Internet, Virginia, EUA. Outubro 1999.
- PRESSMAN, R.S. **Software Engineering**: A Practitioner s Approach. 6th edition. McGraw Hill, 2005.
- PFLEEGER, S.L., **Engenharia de Software**: Teoria e Prática. 2ª edição. São Paulo: Prentice Hall, 2004.
- SILVA, R. P. **UML2 em Modelagem Orientada a Objetos**. 1ª edição. Visual Books, 2007.
- SOMMERVILLE, I. **Engenharia de Software**, 8ª Edição. São Paulo: Pearson – Addison Wesley, 2007.
- SOUZA, V. E. S. **FrameWeb**: um Método baseado em Frameworks para o Projeto de Sistemas de Informação Web. 2007. Dissertação (Mestrado em Informática), Programa de Pós-Graduação em Informática, Universidade Federal do Espírito Santo, Vitória, 2007
- SOUZA, V. E. S., Falbo, R. A.: **FrameWeb**: A Framework-based Design Method for Web Engineering. Euro American Conference on Telematics and Information Systems. EATIS, 2007. Faro, Portugal, 2007.

SOUZA, V.; Falbo, R.A. **An Agile Approach for Web Systems Engineering**, XI Simpósio Brasileiro de Sistemas Multimídia e Web(WebMedia), MG, Brasil, 2005.

OLSINA, L.; ROSSI, L. **Toward web-site quantitative evaluation defining quality characteristics and attributes**. Proceedings of WebNet World Conference on the WWW and Internet, Virginia, EUA. Outubro 1999.

ORT, E. **Introducing the Java EE 6 Platform: Part 1**. Disponível online: <http://www.oracle.com/technetwork/articles/javaee/javaee6overview-full-149950.pdf>, Dezembro 2009. Último Acesso: 06/02/2014