

Ramon Pereira de Matos

**Aplicação do método FrameWeb no  
desenvolvimento de um sistema de informação  
usando os *frameworks* Spring MVC e Vaadin**

Vitória, ES

2017

Ramon Pereira de Matos

**Aplicação do método FrameWeb no desenvolvimento de um sistema de informação usando os *frameworks* Spring MVC e Vaadin**

Monografia apresentada ao Curso de Engenharia da Computação do Departamento de Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Bacharel em Engenharia da Computação.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Departamento de Informática

Orientador: Prof. Dr. Vítor E. Silva Souza

Vitória, ES

2017

---

Ramon Pereira de Matos

Aplicação do método FrameWeb no desenvolvimento de um sistema de informação usando os *frameworks* Spring MVC e Vaadin/ Ramon Pereira de Matos. – Vitória, ES, 2017-

45 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Vítor E. Silva Souza

Monografia (PG) – Universidade Federal do Espírito Santo – UFES  
Centro Tecnológico  
Departamento de Informática, 2017.

1. Engenharia Web. 2. FrameWeb. I. Souza, Vítor Estêvão Silva. II. Universidade Federal do Espírito Santo. IV. Aplicação do método FrameWeb no desenvolvimento de um sistema de informação usando os *frameworks* Spring MVC e Vaadin

CDU 02:141:005.7

---

Ramon Pereira de Matos

# **Aplicação do método FrameWeb no desenvolvimento de um sistema de informação usando os *frameworks* Spring MVC e Vaadin**

Monografia apresentada ao Curso de Engenharia da Computação do Departamento de Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Bacharel em Engenharia da Computação.

Trabalho aprovado. Vitória, ES, 25 de setembro de 2014:

---

**Prof. Dr. Vítor E. Silva Souza**  
Orientador

---

**Prof. Dr. João Paulo A. Almeida**  
Universidade Federal do Espírito Santo

---

**Bruno Borlini Duarte, MSc.**  
Universidade Federal do Espírito Santo

Vitória, ES  
2017

# Agradecimentos

Primeiramente agradeço a Deus, por tudo que Ele tem me proporcionado. Especialmente a oportunidade de viver e sonhar.

Agradeço aos meus pais, Rosangela Pereira e José Brito Ribeiro, pelo amor que me deram e pelo esforço que fizeram para que eu pudesse chegar onde cheguei, pelo apoio e incentivo quando eu quis desistir. Agradeço a toda minha família que sempre esteve ao meu lado e soube compreender a minha ausência em momentos importantes.

A Laudiceia Schuaba e ao seu filho Emanuel Santos que por mais de 5 anos me acolheram voluntariamente em sua residência me fazendo sentir parte de uma nova família, a vocês minha eterna gratidão.

Ao meu orientador Prof. Vitor Estêvão Silva Souza, por ter aceitado ser meu orientador, por ser o melhor exemplo de professor e pela disponibilidade e compreensão em todos os momentos. Agradeço a todos os meus professores de antes e durante a graduação pelo carinho, dedicação e pelo conhecimento a mim proporcionado.

Agradeço à minha namorada Renata Fernandes, por estar ao meu lado e sempre me compreender e apoiar.

Obrigado aos meus amigos “irmãos de coração” pelos momentos de ajuda e de descontração que vivemos juntos. E que se Deus permitir haveremos de viver muito mais.

Agradeço a todos os meus colegas de curso que direto ou indiretamente me ajudaram nessa trajetória, entre os quais destaco o Victor Freitas e Giovani Obermuller que se tornaram amigos.

*“Quando tudo tiver parecendo ir contra você,  
lembre-se que o avião decola contra o vento,  
e não a favor dele.”*  
*(Henry Ford)*

# Resumo

Com ascensão da Internet como um dos principais meios de comunicação, por meio de serviços como a World Wide Web (Web), criou-se uma nova dimensão na utilização de computadores, criando uma demanda por softwares de alta qualidade. Nesse contexto, o desenvolvimento dessas aplicações de maneira *ad-hoc* se tornou insuficiente e surge a necessidade de aplicar os conceitos da Engenharia de Software na criação destas aplicações Web (WebApp), mas de maneira adaptada a esta nova plataforma de implementação. Nasce, assim, a Engenharia Web.

Para atender requisitos de qualidade e agilidade na criação das aplicações Web, foi proposto o método FrameWeb (*Framework-based Design Method for Web Engineering*), que sugere a utilização de uma série de *frameworks* que agilizam o desenvolvimento das WebApps. O método propõe, então, um conjunto de atividades e uma extensão da linguagem UML com quatro tipos de modelos para a fase de projeto. Esses modelos representam conceitos utilizados por algumas categorias de *frameworks*, facilitando, dessa forma, a relação de comunicação entre as equipes de projetistas e a de programadores durante o desenvolvimento da aplicação.

Este trabalho propõe a implementação de duas versões do Sistema de Controle de Afastamentos de Professores (SCAP), uma WebApp que já foi anteriormente implementada utilizando-se o FrameWeb, porém com dois *frameworks* controladores frontais diferentes do que os utilizados anteriormente. Utilizou-se o Spring MVC e Vaadin com o objetivo de testar a aplicabilidade do método a esses *frameworks*.

**Palavras-chaves:** Engenharia Web. WebApp. FrameWeb. Framework. Spring MVC. Vaadin.

# Lista de ilustrações

Figura 1 – Diagrama representando a arquitetura MVC (PRADO, 2015) . . . . .	19
Figura 2 – Funcionamento do padrão arquitetônico “Controlador Frontal” (SOUZA, 2007). . . . .	19
Figura 3 – Arquitetura padrão para WIS baseada no padrão arquitetônico <i>Service Layer</i> (FOWLER, 2002). . . . .	21
Figura 4 – Diagrama de Casos de Uso do subsistema Núcleo (PRADO, 2015). . . . .	26
Figura 5 – Diagrama de Casos de Uso do subsistema Secretaria (PRADO, 2015). . . . .	26
Figura 6 – Diagrama de Classes do SCAP (PRADO, 2015). . . . .	28
Figura 7 – Modelo de Navegação do caso de uso Cadastrar Solicitação. . . . .	31
Figura 8 – Modelo de Navegação do caso de uso Encaminhar Afastamento. . . . .	32
Figura 9 – Modelo de Navegação que cobre os casos de uso Manifestar-se Contra Afastamento e Deferir Parecer. . . . .	32
Figura 10 – Organização do código-fonte do projeto SCAP no Spring MVC. . . . .	34
Figura 11 – Tela de login do SCAP. . . . .	35
Figura 12 – Tela com a formulário de cadastro de solicitação de afastamento. . . . .	35
Figura 13 – Tela que apresenta os dados de um pedido de afastamento. . . . .	36
Figura 14 – Modelo de Navegação do caso de uso Cadastrar Usuário. . . . .	37
Figura 15 – Modelo de Navegação do caso de uso Consultar Afastamento. . . . .	37
Figura 16 – Modelo de Navegação que cobre os Cadastrar Parecer CT e Cadastrar Parecer PRPPG. . . . .	38
Figura 17 – Tela de login do SCAP . . . . .	40
Figura 18 – Tela formulário de busca e lista de solicitações de afastamento. . . . .	40
Figura 19 – Tela com o formulário de cadastro de usuário. . . . .	41
Figura 20 – Tela que apresenta os dados de um pedido de afastamento. . . . .	41



# Lista de tabelas

Tabela 1 – Atores do SCAP (PRADO, 2015). . . . .	25
--	----

# Lista de abreviaturas e siglas

AOP	Aspect Oriented Programming
API	Application Programming Interface
DAO	Data Access Object
DSL	Domain Specific Language
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
JDBC	Java Database Connectivity
JPA	Java Persistence API
MVC	Model-View-Controller
ORM	Object-relational Mapping
SCAP	Sistema de Controle de Afastamento e Professores
UI	User Interface
UML	Unified Modeling Language
WIS	Web Information System

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>1.1</b>	<b>Motivação e Justificativa</b>	<b>13</b>
<b>1.2</b>	<b>Objetivo</b>	<b>13</b>
1.2.1	Objetivos Específicos	13
<b>1.3</b>	<b>Metodologia</b>	<b>13</b>
<b>1.4</b>	<b>Organização da Monografia</b>	<b>14</b>
<b>2</b>	<b>ENGENHARIA WEB, FRAMEWEB E FRAMEWORKS</b>	<b>15</b>
<b>2.1</b>	<b>Engenharia Web</b>	<b>15</b>
<b>2.2</b>	<b>Frameworks</b>	<b>17</b>
2.2.1	Frameworks MVC	18
2.2.2	Spring	20
2.2.3	Vaadin	20
<b>2.3</b>	<b>FrameWeb</b>	<b>20</b>
2.3.1	Arquitetura Web utilizada no FrameWeb	21
2.3.2	Linguagem de modelagem de FrameWeb	22
<b>3</b>	<b>PROJETO SCAP</b>	<b>24</b>
<b>3.1</b>	<b>Descrição do Escopo</b>	<b>24</b>
<b>3.2</b>	<b>Modelos de Casos de Uso</b>	<b>25</b>
<b>3.3</b>	<b>Análise do SCAP</b>	<b>27</b>
<b>4</b>	<b>IMPLEMENTAÇÃO SCAP</b>	<b>30</b>
<b>4.1</b>	<b>Spring MVC</b>	<b>30</b>
4.1.1	Modelo de Navegação	30
4.1.2	Tecnologias Utilizadas	33
4.1.3	Pacotes SCAP	33
4.1.4	Resultado da Implementação	35
<b>4.2</b>	<b>Vaadin</b>	<b>36</b>
4.2.1	Modelo de Navegação	36
4.2.2	Tecnologias Utilizadas	38
4.2.3	Pacotes SCAP	38
4.2.4	Resultado Implementação	39
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>42</b>

<b>REFERÊNCIAS</b>	<b>44</b>
--------------------	-----------

# 1 Introdução

No início o desenvolvimento das aplicações Web se dava de forma *ad-hoc*, porém com o aumento da complexidade dos softwares criados para a Internet este método não era mais suficiente. Logo, para o desenvolvimento de sistemas e aplicações que funcionem na Web (*WebApps* - *Web Applications*) passou-se a aplicar os conceitos de Engenharia de Software, porque embora essas *WebApps* possuam características e requisitos exclusivos, elas não deixam de ser um tipo de software.

Com a rápida evolução da Internet e ascensão como um dos principais meios de comunicação existentes, por meio de serviços como a *World Wide Web* (Web), criou-se uma nova dimensão na utilização de computadores, fazendo surgir a demanda por softwares de alta qualidade para suprir requisitos tais como segurança, velocidade, compatibilidade, confiabilidade, usabilidade, etc. Neste contexto nasce a Engenharia Web (PRESSMAN, 2011).

Em 2007 foi proposto o método FrameWeb (*Framework-based Design Method for Web Engineering*) (SOUZA, 2007), em que é sugerida a utilização de uma série de *frameworks* para agilizar o desenvolvimento de uma aplicação Web (*WebApp*). O método propõe, então, um conjunto de atividades e uma extensão da linguagem UML com quatro tipos de modelos para a fase de projeto. Esses modelos representam conceitos utilizados por algumas categorias de *frameworks*, facilitando dessa forma a relação de comunicação entre as equipes de projetistas e a de programadores durante o desenvolvimento da aplicação.

Desde sua criação, o método vem sendo aplicado no desenvolvimento de diferentes *WebApps*. Em particular, para avaliar a aplicabilidade do método a diferentes *frameworks*, propõe-se desenvolver um mesmo sistema, denominado Sistema de Controle de Afastamento de Professores (SCAP), porém modificando um dos *frameworks* utilizados a cada vez. Uma primeira versão do SCAP (DUARTE, 2014) foi desenvolvida na plataforma Java EE 7 (DEMICHIEL; SHANNON, 2013), na qual o JSF foi utilizado como *framework* controlador frontal (*framework* MVC). Já em (PRADO, 2015) foi utilizado o *framework* VRaptor 4 (CAVALCANTI, 2014).

A partir dos requisitos levantados por Duarte (2014) e revisados por Prado (2015), foram realizadas duas novas implementações do SCAP, aplicando novamente o método FrameWeb, entretanto utilizando os *frameworks* Spring MVC e Vaadin como controladores frontais.

## 1.1 Motivação e Justificativa

As motivações desse projeto incluem: (a) prover um estudo e aprendizado sobre o método FrameWeb, que possa agregar conhecimento pessoal e extrair as vantagens de utilização do mesmo; (b) prover implementações do SCAP com diferentes *frameworks* controladores frontais, de modo a contribuir para futuras pesquisas ou em andamento; (c) exercitar o conteúdo aprendido nas disciplinas Programação III (DEITEL; DEITEL, 2010), Engenharia de Software (PRESSMAN, 2011), Engenharia de Requisitos de Software (FALBO, 2017a), Projeto de Sistemas de Software (BLAHA; RUMBAUGH, 2006) e Banco de Dados (ELMASRI; NAVATHE, 2011).

## 1.2 Objetivo

Construir um sistema de informação Web baseado no método FrameWeb (SOUZA, 2007) utilizando dois diferentes *frameworks* MVC (Controladores Frontais), a saber: Spring MVC e Vaadin.

### 1.2.1 Objetivos Específicos

- Implementar o sistema de Informação Web SCAP seguindo requisitos definidos nos trabalhos anteriores (DUARTE, 2014; PRADO, 2015), utilizando os *frameworks* Spring MVC e Vaadin;
- Produzir diagramas de navegação FrameWeb para as principais funcionalidades do SCAP em suas duas novas implementações;
- Sugerir proposta de modificações para o método FrameWeb, baseadas na experiência adquirida no desenvolvimento do Projeto de Graduação ou constatar aplicabilidade do mesmo aos *frameworks* escolhidos.

## 1.3 Metodologia

O desenvolvimento deste trabalho se deu da seguinte forma:

1. **Revisão bibliográfica:** Estudo do método FrameWeb e estudos dos projetos de graduação de Duarte (2014) e Prado (2015), já realizados anteriormente seguindo o método;
2. **Implementação do sistema:** a partir do estudo da Análise de Requisitos e do Projeto de Sistema, implementar o sistema seguindo o método FrameWeb, com os dois *frameworks* escolhidos e produção dos novos diagramas de navegação FrameWeb;

3. **Análise dos resultados:** analisar, a partir das etapas anteriores, a aplicabilidade do método FrameWeb, seguindo os modelos proposto pelo método para os *frameworks* escolhidos. Elaboração de proposta de modificações, se for o caso.

## 1.4 Organização da Monografia

Essa monografia está organizada em 5 capítulos incluindo a introdução.

No Capítulo 2 apresenta-se um levantamento dos estudos e pesquisas sobre os principais temas abordados ao longo da monografia: Engenharia de Software e a Engenharia Web, *frameworks* e FrameWeb.

O Capítulo 3 apresenta uma especificação e análise dos requisitos do SCAP (Sistema de Controle de Afastamento de Professores) definidos nos trabalhos anteriores (DUARTE, 2014; PRADO, 2015) e o seu desenvolvimento utilizando o FrameWeb.

No Capítulo 4 é demonstrado o desenvolvimento do SCAP com os *frameworks* propostos, Spring MVC e Vaadin, utilizando o método FrameWeb.

No Capítulo 5 são apresentadas as conclusões extraídas deste trabalho juntamente com as sugestões para o método FrameWeb.

## 2 Engenharia Web, FrameWeb e Frameworks

No princípio da *World Wide Web*, os sites eram formados por simples conjunto de arquivos de hipertexto ligados que apresentavam informações de texto e gráficos limitados. Com passar do tempo as ferramentas de desenvolvimento foram tornando possível oferecer junto com as páginas de textos e gráficos uma capacidade computacional. Desta forma nascem as aplicações para Web conhecidas e tratadas neste trabalho como *WebApps* (*Web Applications*).

Durante a primeira década a partir do surgimento da Web o desenvolvimento das aplicações era feito de maneira *ad-hoc*, sem uma metodologia ou processo para apoiar os seus criadores, assim o projeto se desenvolvia de uma maneira artística. Logo, torna-se essencial o uso dos conceitos apresentados na Engenharia de Software, pois apesar de podermos afirmar que elas são diferentes, as *WebApps* são apenas uma dentre as diferentes categorias de software. O ambiente Web, no entanto, possui como características particulares: uso intensivo de redes, simultaneidade, disponibilidade, evolução contínua, segurança, estética, etc. (PRESSMAN, 2011).

Desde sua criação podemos perceber a evolução da Web transformando sofisticados ambientes computacionais e com a expansão da Internet podemos ter verdadeiros sistemas de informação dentro da Web espalhados por diversos lugares, os quais denominamos como WIS (*Web-based Information Systems*).

Nas próximas seções vamos apresentar melhor o conceito da adaptação da Engenharia de Software aplicada na Web, os conceitos dos *frameworks* que surgiram para agilizar e facilitar o desenvolvimento de *WebApps* e o método FrameWeb proposto para auxiliar na construção dessas aplicações.

### 2.1 Engenharia Web

Engenharia Web (*Web Engineering - WebE*), é a Engenharia de Software aplicada ao desenvolvimento Web (PRESSMAN, 2011). Segundo Murugesan et al. (2001) “é o estabelecimento do uso de um tom científico, engenharia, princípios de gestão, abordagens disciplinadas e sistemáticas para o sucesso do desenvolvimento, implantação e manutenção de sistemas e aplicativos baseados na Web de alta qualidade.”

Em um curto espaço de tempo, a Internet e a *World Wide Web* tonaram-se onipresentes nas nossas vidas, superando todos os avanços tecnológicos que presenciamos na nossa história e nos afetando diretamente. Aumentou enormemente o seu alcance e extensão e, hoje, está presente nas indústrias, viagens, bancos, educação e governo, que



a utilizam para melhorar e aprimorar suas operações. O comércio eletrônico também cresceu e se expandiu além das fronteiras nacionais, hoje com o avanço das tecnologias sem fio e dispositivos habilitados para acesso à Internet permite que realizemos várias tarefas do nosso cotidiano através desses dispositivos, o que desencadeou uma nova onda de aplicativos Web móveis.

Neste momento, em que a maioria das pessoas depende de sistemas e aplicativos baseados na Web, estes precisam ter um bom desempenho e serem confiáveis e, para que isso ocorra, os desenvolvedores precisam de uma metodologia sólida, um processo disciplinado e repetitivo, melhores ferramentas de desenvolvimento e um conjunto de boas diretrizes. Para que essas expectativas sejam atendidas usa-se a Engenharia Web para minimizar os riscos, melhorar a capacidade de manutenção e a qualidade das *WebApps*.

Ao contrário da percepção de alguns desenvolvedores de software e profissionais da Engenharia de Software, a Engenharia Web não é um clone da Engenharia de Software, entretanto, esta comunga de muitos princípios e acrescenta novas abordagens, metodologias, ferramentas, técnicas e diretrizes para atender os requisitos específicos dos sistemas baseados na Web. As *WebApps* mudam rapidamente em seus requisitos, conteúdos, e funcionalidades durante o seu ciclo de vida, mais do que acontece normalmente no desenvolvimento dos sistemas tradicionais de software (GINIGE; MURUGESAN, 2001).

Como a Engenharia de Software possui métodos disciplinados, a Engenharia Web adapta esses métodos para a criação de *WebApps*. A complexidade de se projetar e implementar um software com vários requisitos e muitos *stakeholders*, como é o caso das aplicações Web, torna relevante a adoção de uma metodologia que possui uma abordagem incremental e interativa. Um processo desse modelo divide a concepção de software em interações. Em cada interação são realizadas as atividades de análise, projeto, implementação e testes.

O modelo de processo Engenharia Web inicia-se na especificação dos requisitos, onde captura-se tudo aquilo considerado importante para entrega da aplicação. A coleta de requisitos, captura todas as informações possíveis sobre as funções que o sistema deve executar (requisitos funcionais) e as restrições sob as quais este deve operar (requisitos não funcionais). Posteriormente, realiza-se a modelagem de análise.

Na criação do Modelo de Análise o intuito é prover uma descrição dos domínios de informação, funcional e comportamental necessários para um sistema baseado em computadores. Este modifica-se dinamicamente a medida que se aprende mais sobre o sistema a ser construído e o cliente entende melhor aquilo que necessita. Há cinco principais tipos de análise resultantes dessa modelagem, que são: **de conteúdo**, que identifica o espectro completo do conteúdo a ser fornecido pela *WebApp*; **de interação**, descrevendo como os usuários interagem com a *WebApp*; **funcional**, que define as operações aplicadas ao conteúdo as funções de processamento independente do conteúdo, mas necessária para

usuário final; **de navegação**, definido a estratégia geral de navegação para a *WebApp*; e **de configuração**, que descreve o ambiente e a infraestrutura nos quais a *WebApp* reside.

Conduzidos pelos resultados da modelagem de análise, a fase de projeto da *WebApp* foca seis tópicos principais: projeto de interface, projeto de estética, projeto de conteúdo, projeto de navegação, projeto arquitetural e projeto de componentes. Projeto é uma atividade da engenharia que conduz um produto de qualidade (PRESSMAN, 2011).

Segundo Olsina, Lafuente e Rossi (2001), o conjunto de atributos de qualidade a serem considerados na aplicação para Web são:

- **Usabilidade:** caracteriza-se pela facilidade de compreensão geral do site, emprego do recurso de ajuda e *feedback online*, planejamento da interface para que a mesma seja agradável esteticamente e que possua características especiais;
- **Funcionalidade:** a capacidade de busca e recuperação, as características de navegação a adaptação aos *browsers* e as questões relacionadas ao domínio da aplicação;
- **Confiabilidade:** validação e recuperação dos dados de entrada dos usuários, recuperação de erros e processamento correto dos links;
- **Eficiência:** o tratamento das questões relacionadas ao desempenho do tempo de resposta e as velocidades de geração de páginas e imagens;
- **Manutenibilidade:** existe uma rápida evolução tecnológica aliada à necessidade de atualização constante do conteúdo e das informações disponibilizadas na Web, logo o software Web deve ser fácil de corrigir, adaptar e estender.

Existem alguns métodos que podem ser aplicados no desenvolvimento de aplicações Web para facilitar a produção, como WAE (CONALLEN, 2002), OOWS (PASTOR; FONS; PELECHANO, 2003) e OOHD (SCHWABE; ROSSI, 1998). Dentro do contexto da Engenharia Web, foi proposto o método FrameWeb, um método que propõe a otimização dos processos da Engenharia Web para *WebApp* baseadas em *frameworks*. Este método será apresentado na Seção 2.3.

## 2.2 Frameworks

Na década de 1980 houve uma popularização da linguagem Smalltalk, paralelamente às bibliotecas de classes, dando início à construção de *Frameworks* de aplicação, que acrescentam às bibliotecas de classes os relacionamentos e interação entre as diversas classes que o compõem, promovendo a reutilização das linhas de código, assim como o projeto abstrato envolvendo o domínio de aplicação.

Os *frameworks* permitem reutilizar toda arquitetura de um domínio específico, bem como componentes isolados. Com a intenção de promover a melhoria da produtividade, qualidade e manutenibilidade, foram desenvolvidos diversos *frameworks* nas últimas décadas (MALDONADO et al., 2002). Na Engenharia Web, o desenvolvimento de WISs possui uma infraestrutura arquitetônica comum. Portanto, logo depois da criação dos primeiros sistemas foram desenvolvidos vários *frameworks* que generalizavam essa infraestrutura, deste modo o *framework* pode ser visto como um conjunto de componentes prontos que podem ser utilizados, mediante configuração, facilitando a produção de código na construção de sistemas Web (SOUZA, 2007).

Souza (2007) organiza os frameworks de desenvolvimento de WISs e seis categorias diferentes:

- *Frameworks* MVC (Controladores Frontais);
- *Frameworks* Decoradores;
- *Frameworks* de Mapeamento Objeto/Relacional;
- *Frameworks* de Injeção de Dependência (Inversão de Controle);
- *Frameworks* para Programação Orientada a Aspectos (AOP);
- *Frameworks* para Autenticação e Autorização.

Nas próximas subseções apresentaremos uma introdução aos *frameworks* MVC, por ter relação direta com a arquitetura proposta pelo FrameWeb e os *frameworks* utilizados na implementação do SCAP.

### 2.2.1 Frameworks MVC

MVC é a abreviatura de *Model-View-Controller* (GAMMA et al., 1994), ou Modelo-Visão-Controle. Desde o seu desenvolvimento esta arquitetura foi bem aceita nas diversas áreas da Engenharia de Software e é possivelmente, uma das mais utilizadas na atualidade para construção de aplicações Web. A Figura 1 ilustra a arquitetura MVC.

O elemento visão é encarregado de realizar a representação gráfica dos dados contidos no modelo, realiza solicitação das atualizações do modelo e é responsável de receber os estímulos do usuário e enviá-los ao controlador para que as devidas solicitações sejam atendidas. O modelo encapsula o estado da aplicação, através das chamadas de métodos ele pode ter o seu estado alterado pelo controlador ou consultado pela visão, no caso de alteração de estado o modelo notifica a visão sobre as alterações. O controlador é responsável por receber os estímulos do usuário enviados pela visão, transformando essas ações em atualizações no modelo e seleciona a visão para a entrega das respostas.

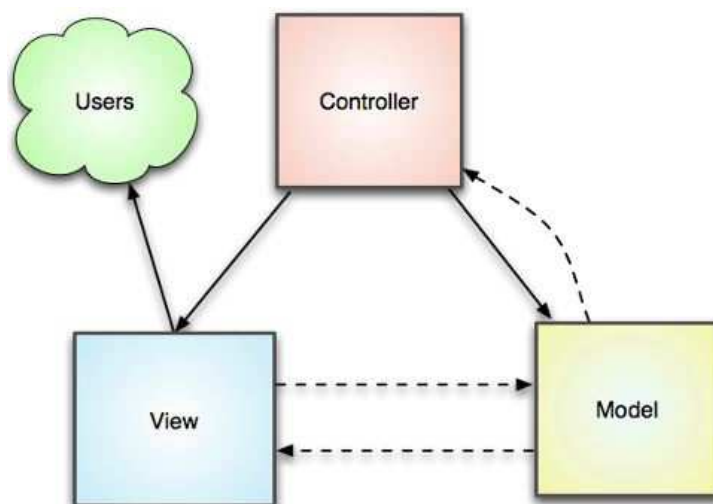


Figura 1 – Diagrama representando a arquitetura MVC (PRADO, 2015)

Apesar de alta aplicabilidade desse padrão MVC em aplicações Web, sua forma teórica não se encaixa perfeitamente nessa plataforma, pois a camada de modelo se encontra no servidor Web e como a comunicação é sempre iniciada na direção cliente – servidor o modelo não pode notificar o elemento visão sobre alterações, visto que este se encontra no navegador. Porém, quando aplicado a Web o nome correto para esse padrão seria “Controlador Frontal” (*Front Controller*) (ALUR; CRUPI; MALKS, 2003). A Figura 2 ilustra esse padrão arquitetônico.

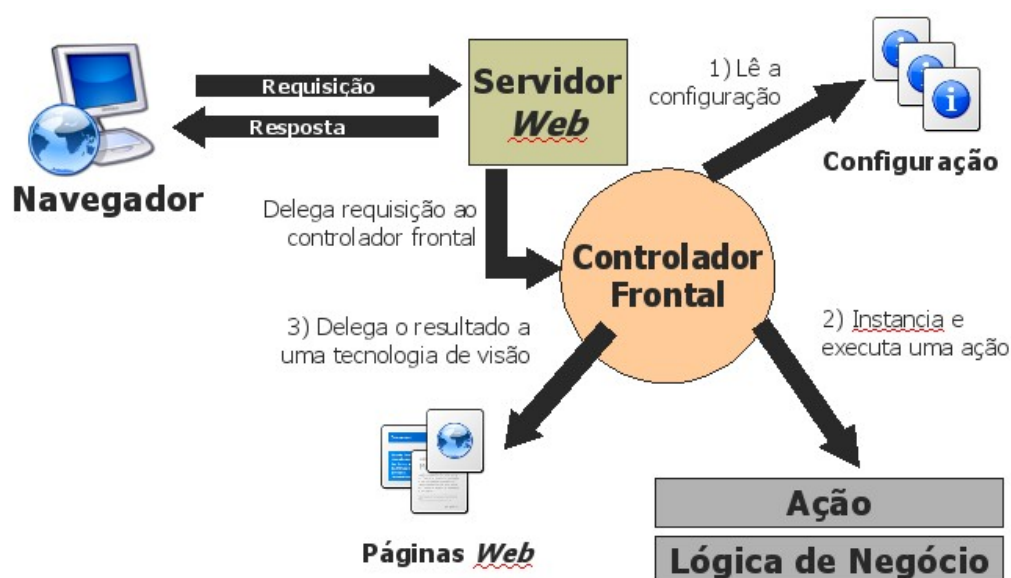


Figura 2 – Funcionamento do padrão arquitetônico “Controlador Frontal” (SOUZA, 2007).

O cliente realiza uma requisição HTTP por meio do navegador, que pode ser um pedido de leitura de uma página ou envio de dados para processamento. O servidor Web encarrega o controlador frontal de tratar a requisição e este passa a gerenciar todo o processo, realizando as devidas ações para que seja emitida uma resposta à requisição, encaminhando

para a visão. Os *frameworks* MVC oferecem a implementação dessa arquitetura facilitando com que esse seja aplicado diante de configurações (SOUZA, 2007).

### 2.2.2 Spring

O Spring *Framework* (JOHNSON et al., 2004) fornece um modelo abrangente de programação e configuração para aplicativos corporativos modernos baseados em Java em qualquer tipo de plataforma de implantação. Um elemento chave do Spring é o suporte a infraestrutura no nível de aplicação para que as equipes possam se concentrar na construção da lógica de negócio do aplicativo, sem vínculos desnecessários com ambientes de implantação específicos.

Destacam-se entre as suas características, a injeção de dependências, programação orientada a aspectos (AOP - *Aspect Oriented Programming*), Spring MVC, suporte para JDBC e JPA, etc. Desde o início já incluso no Spring Framework, o Spring Web MVC é uma estrutura original criada na API *Servlet*. O Spring MVC, como é mais conhecido, foi planejado a partir da ideia de um padrão de controlador frontal suportando diversos fluxos de trabalho.

### 2.2.3 Vaadin

Vaadin (GRÖNROOS, 2017) é uma plataforma de desenvolvimento que oferece todas as ferramentas necessárias para criar facilmente um aplicativo da Web. O Vaadin Framework é uma estrutura UI e tem seu código escrito em Java e executado na JVM do servidor, enquanto a sua UI é renderizada como HTML5 no navegador. Este *framework* também torna automática a comunicação entre o servidor e navegador.

Provê uma forte abstração protegendo o desenvolvedor das dificuldades na criação de aplicativos Web, pois ele não precisa se preocupar com a comunicação cliente-servidor ao desenvolver a sua aplicação. Ele ainda possui uma série de componentes de UI reutilizáveis e é extensível com HTML e JavaScript.

Dentre as diferenças do Vaadin e os demais *frameworks* podemos destacar que este em suas aplicações padrão não possui uma navegação tradicional de uma aplicação Web, pois geralmente são executados em uma única página. Mas isso não se torna um problema devido ao Vaadin dispor de uma classe que pode normalmente implementar e gerenciar a navegação de uma aplicação.

## 2.3 FrameWeb

O FrameWeb (SOUZA, 2007) define-se como método de projeto para a construção de sistemas de informação Web (*Web Information System – WISs*) baseado em *frameworks*.

O método admite que tipos diferentes de *frameworks* serão utilizados no desenvolvimento da aplicação, definindo uma arquitetura básica para o WIS e propondo modelos que aproximam o uso dos *frameworks* a implementação do sistema. Um dos motivos para a sua criação se deve ao fato que dentre as diversas propostas para Engenharia Web, como metodologias, métodos, linguagens, etc., nenhuma delas leva em conta o relacionamento dos aspectos característicos dos *frameworks* comumente utilizados na construção de WISs.

A fase de Projeto concentra as propostas principais do método: (i) definição de uma arquitetura padrão que divide o sistema em camadas, de modo a se integrar bem com os *frameworks* utilizados; (ii) proposta de um conjunto de modelos de projeto que trazem conceitos utilizados pelos *frameworks* para esta fase do processo por meio da definição de uma linguagem específica de domínio que faz com que os diagramas fiquem mais próximos da implementação (SOUZA, 2007; MARTINS, 2016).

Na fase de implementação a produção do código é facilitada pela utilização dos *frameworks*, principalmente porque os modelos de projeto elaborados remetem a componentes relacionados a eles (SOUZA, 2007).

### 2.3.1 Arquitetura Web utilizada no FrameWeb

A arquitetura lógica padrão WISs definida no FrameWeb é baseada no padrão arquitetônico *Service Layer* (Camada de Serviço) (FOWLER, 2002). Como pode ser visto na Figura 3, esse sistema arquitetônico é dividido em três camadas: lógica de apresentação, lógica de negócio e lógica de acesso a dados.

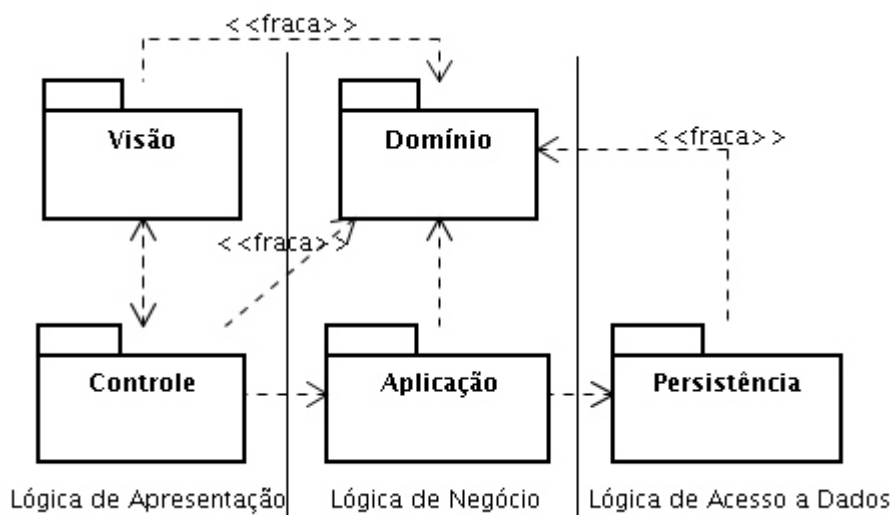


Figura 3 – Arquitetura padrão para WIS baseada no padrão arquitetônico *Service Layer* (FOWLER, 2002).

A primeira camada é formada pelos pacotes de Visão e Controle. O pacote Visão tem como principal objetivo a interação do cliente com o sistema e nela encontram-se as

páginas Web, folhas de estilo, imagens e *scripts* que executam do lado cliente. O pacote Controle é responsável pelo monitoramento dos estímulos enviados pelo cliente através dos elementos da Visão, nesse pacote também abrange as classes de controle e outros arquivos relacionados ao *framework* Controlador Frontal. Estes dois pacotes possuem mutua dependência pois as respostas esperadas pelo cliente através dos estímulos provocados acionam a classe controle que apresenta informações contidas nas classes do Domínio.

Domínio e Aplicação são os pacotes que compõem a segunda camada, denominada Lógica de Negócio. O pacote Domínio contém classes que representam os conceitos do domínio do problema, identificados e modelados na fase de análise pelos diagramas de classes que são revisados na fase de projeto. A Aplicação tem como objetivo a implementação dos casos de uso definidos na fase de especificação de requisitos, para prover uma camada de serviço independente da camada de apresentação responsável pela interface com o usuário.

Apesar da independência da interface com o usuário, o pacote Controle depende do pacote Aplicação para tornar possível que os estímulos provocados pelo usuário na Visão sejam transformados, pelas classes de Controle, em métodos no pacote Aplicação para desta maneira executar os casos de usos.

A lógica de acesso a dados, terceira e última camada, conta com um único pacote, Persistência, responsável pelo armazenamento de informações dos atributos de classes que precisam ser persistidas em mídia de longa duração, como banco de dados, arquivos etc. Com aplicação do FrameWeb espera-se a utilização de um *framework* de mapeamento objeto/relacional por meio do padrão de projeto *Data Access Object* (DAO) (ALUR; CRUPI; MALKS, 2003), este adiciona uma camada de abstração que promove um independência da lógica de acesso a dados da tecnologia de persistência possibilitando uma fácil substituição do *framework* ORM se necessário.

### 2.3.2 Linguagem de modelagem de FrameWeb

Na fase de projeto também são gerados componentes que serão implementados na próxima fase de produção do software. Para agilizar o trabalho dos programadores foi determinada uma linguagem de modelagem específica que representasse diretamente os conceitos existentes nos *frameworks*, devido à grande integração entre eles.

Em sua proposta original (SOUZA, 2007), o FrameWeb utilizava a mesma abordagem utilizada por outras linguagens de modelagem para Web, como WAE (CONALLEN, 2002) e UWE (KOCH et al., 2000), definindo extensões leves (*lightweight extensions*) ao meta-modelo da UML para representar componentes típicos da plataforma Web e dos *frameworks* utilizados, criando um perfil UML. Entretanto, a evolução do FrameWeb proposta por Martins (2016) define uma linguagem Específica de Domínio (*Domain Specific*



*Language* ou DSL) formal para o FrameWeb, por meio de metamodelos que estendem uma parte do metamodelo da UML. Tais metamodelos especificam a sintaxe abstrata da linguagem, descrevendo os conceitos e as regras de modelagem específicas do método e, conseqüentemente, permitem a criação de ferramentas e mecanismos para que esta linguagem possa ser utilizada de forma correta e direcionada.

A seguir é apresentado um breve resumo dos modelos propostos no FrameWeb, todos eles baseados no diagrama de classes da UML:

- **Modelo de Entidades:** é a representação dos objetos de domínio do problema e o seu mapeamento para persistência em um banco de dados relacional;
- **Modelo de Persistência:** é um diagrama que conduz a implementação das classes DAO existentes, mostrando todas as interfaces e suas implementações, juntamente com os métodos contidos em cada classe;
- **Modelo de Navegação:** é um diagrama que representa os diferentes componentes que formam a camada de lógica de apresentação, como páginas Web, formulários HTML e as classe de controle do *framework Front Controller*. Além de guiar a implementação das páginas e das classes de controle, dá clara noção da interação entre os pacotes de Visão e Controle;
- **Modelo de Aplicação:** é um diagrama que representa as classes de serviços do sistema para conduzir a implementação dos casos de uso e a suas dependências. Revela ainda a configuração de dependência entre os pacotes de Controle, Aplicação e Persistência, bem como, determina quais as classes de ação dependem de quais classes de serviço.



## 3 Projeto SCAP

Neste capítulo apresenta-se o os requisitos do SCAP levantados anteriormente por Duarte (2014) e Prado (2015). A Seção 3.1 descreve o escopo do sistema; a Seção 3.2 apresenta os requisitos levantados na forma de casos de uso; e a Seção 3.3 mostra o diagrama de classes resultante da análise de tais requisitos.

O leitor interessado encontrará mais detalhes, bem como o resultado da fase de projeto de sistemas, incluindo modelos FrameWeb considerando as arquiteturas propostas por estes autores, em seus trabalhos originais (DUARTE, 2014; PRADO, 2015). Os requisitos apresentados aqui foram usados como base para construção de novos modelos FrameWeb e implementação de novas versões do SCAP, considerando os *frameworks* Spring MVC e Vaadin, a serem apresentados no capítulo seguinte.

### 3.1 Descrição do Escopo

Para participar de eventos oficiais no Brasil ou no exterior os professores do Departamento de Informática (DI) da UFES precisam submeter um pedido de afastamento ao departamento. As solicitações de afastamento podem ser avaliadas em até três instâncias, por professores do DI, pela diretoria do Centro Tecnológico (CT) e a Pró-reitoria de Pesquisa e Pós-Graduação (PRPPG).

No DI existe a Câmara Departamental, formada por funcionários do departamento e pelos representantes discentes. Quando a solicitação de e afastamento é para algum evento nacional, cabe a esta a responsabilidade pela avaliação do pedido de afastamento. Assim, o pedido é realizado por meio de lista de e-mails dos funcionários do DI e, caso nenhum membro da câmara departamental se manifeste contra no prazo de dez dias, o pedido é aprovado. Desta forma, para eventos no Brasil o processo ocorre completamente dentro do departamento.

Para eventos no exterior, um professor que não tenha relação de parentesco com o solicitante é escolhido para ser o relator do pedido. O relator emite um parecer que também é avaliado pela Câmara Departamental e, se aprovado, é encaminhado para a diretoria do CT e posteriormente para a PRPPG. Se aprovado em todas as instâncias, o afastamento é concedido e publicado no Diário Oficial da União. Entretanto o SCAP trabalha apenas com a tramitação realizada dentro do DI, não existindo uma integração com os processos do CT e da PRPPG. Logo, o escopo do sistema se delimita a tratar da solicitação de afastamento na parte que cabe ao DI.

## 3.2 Modelos de Casos de Uso

O processo de Engenharia de Requisitos determina que depois da definição do escopo sejam modelados os casos de uso, sendo este processo elaborado conforme a proposta de Falbo (2017a). A Tabela 1 lista e descreve os atores identificados.

Tabela 1 – Atores do SCAP (PRADO, 2015).

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

Os professores cadastram as suas solicitações de afastamento no sistema, podem se manifestar em relação ao pedido de afastamento solicitados por outros professores dentro do prazo determinado. Se escolhidos para relatoria de um afastamento internacional, é sua responsabilidade produzir um parecer de aprovação ou não para o afastamento.

O chefe de departamento é o responsável pela definição de um relator para solicitação de afastamento internacional, ele é um professor selecionado para exercer essa função administrativa e possui um prazo determinado para exercê-la.

Os secretários são responsáveis pela parte administrativa do sistema, realizando o cadastro dos professores, além das relações de parentesco ente eles, bem como o cadastro do mandato do chefe do departamento. Como já exposto anteriormente, o sistema auxilia atividades do processo de afastamento referentes ao DI, logo o secretário também é responsável pela inserção de documentos provenientes dos pareceres do CT e PRPPG.

O SCAP foi dividido em dois subsistemas: o Núcleo contém os casos de uso referente aos atores professor e chefe de departamento, enquanto o subsistema Secretaria contempla os casos de usos referentes ao ator secretário.

Nas figuras 4 e 5 são apresentados os diagramas de caso de uso destes subsistemas. Em seguida apresentamos resumidamente a descrição dos casos de uso levantados.

No caso de uso **Solicitar Afastamento** o professor cadastra uma solicitação de afastamento no sistema para que seja analisad, podendo cancelar seu pedido de afastamento por meio do caso de uso **Cancelar Afastamento**.

O caso de uso **Encaminhar Afastamento** é executado quando ocorre uma solicitação de afastamento internacional e é realizado pelo Chefe do Departamento, que determina um professor para ser relator deste afastamento. O caso de uso **Definir Parecer** é realizado quando o professor escolhido como relator de um afastamento internacional cadastra no sistema o parecer sobre a referida solicitação.

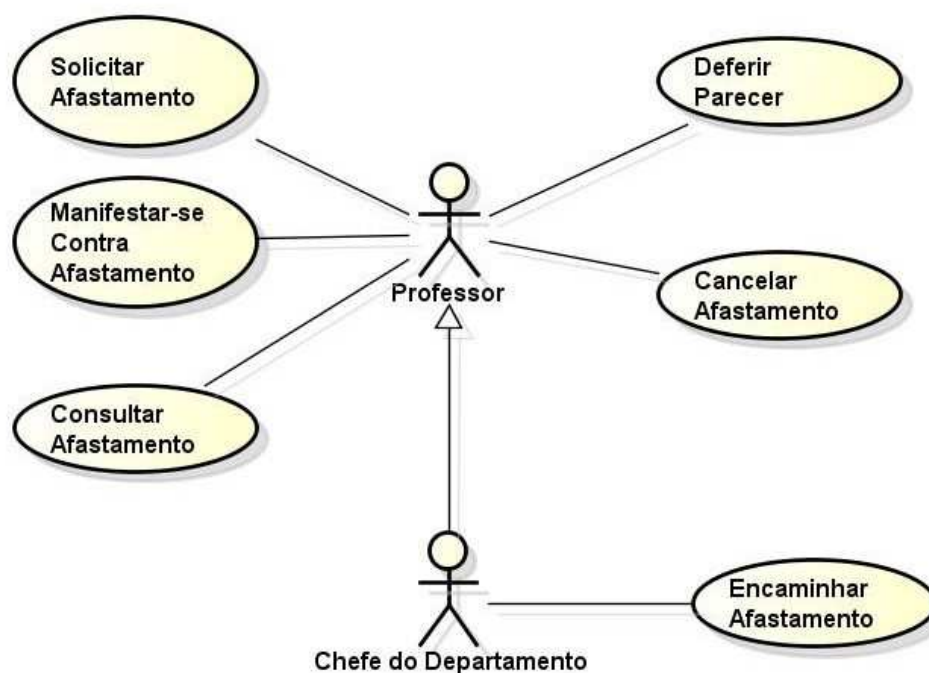


Figura 4 – Diagrama de Casos de Uso do subsistema Núcleo (PRADO, 2015).

O caso de uso **Consultar Afastamento** é utilizado quando o professor, chefe de departamento ou secretário busca informação sobre um pedido de afastamento. Quando um professor decide se manifestar contra um afastamento ele faz uma busca por esse afastamento e logo depois utiliza o caso de uso **Manifestar-se Contra Afastamento** para realizar a sua manifestação.

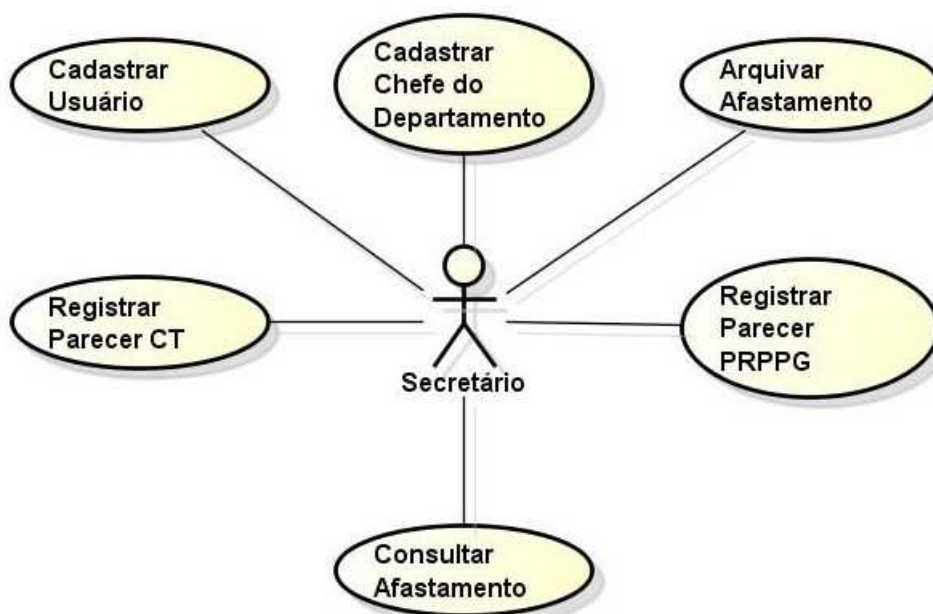


Figura 5 – Diagrama de Casos de Uso do subsistema Secretaria (PRADO, 2015).

Os secretários são responsáveis pela parte administrativa, logo utilizam-se dos casos

de uso **Cadastrar Usuário** e **Cadastrar Chefe do Departamento** para, no primeiro, adicionar um novo professor ou secretário ao sistema e, no segundo, registrar um professor previamente cadastrado como chefe do departamento, informando a data de início e fim do seu mandato.

Para um pedido de afastamento internacional é necessário inserir no sistema os documentos de outras instâncias de aprovação e, para essas tarefas, utiliza-se os casos de uso **Registrar Parecer CT** e **Registrar Parecer PRPPG** para cadastro dos pareceres.

O caso de uso **Arquiva Afastamento** acontece no final da tramitação onde ocorre a mudança de status para “Arquivado”.

### 3.3 Análise do SCAP

Ao longo da fase de análise há a necessidade de adaptar as entidades do mundo real, para que elas possam ser modeladas. Modelos estruturais buscam capturar os principais conceitos do domínio suas relações e propriedades. Dentre esses tipos de modelo, destaca-se o diagrama de classes no paradigma de Orientação a Objetos, assim a modelagem pode representar os artefatos do mundo real em classes e representar também a relação e interação entre classes (FALBO, 2017a). Ao realizar a abstração dos conceitos do mundo real em classe de objetos seus atributos e relacionamentos, são formados diagramas de classes que mostram suas propriedades e as associações existentes entre as classes que compõem o sistema (PERUCH, 2007).

Na criação do diagrama de classes deve se destacar a relevância da multiplicidade das relações que afeta diretamente as classes de projeto relacionadas a persistência de dados. O SCAP foi dividido em dois subsistemas, entretanto as classes de domínio do problema são representadas em um único diagrama, que pode ser observado na Figura 6. Todas as classes, portanto, pertencem ao subsistema Núcleo. Segundo Prado (2015), tal divisão foi realizada para facilitar a implementação e não devido à natureza das classes criadas.

As classes **Professor** e **Secretário** representam os professores e secretários do departamento e herdam os atributos da classe **Pessoa**. Os professores que possuem relação de parentesco têm essa relação representadas pela classe **Parentesco**. A classe **Mandato** representa um professor que é escolhido como chefe de departamento.

A classe **Afastamento** possui os dados referentes a uma solicitação de afastamento realizada por um professor. Esta pode conter um ou mais documentos que são representados pela classe **Documento**. Se o afastamento for para fora do Brasil ele precisa de um relator que está relacionado na classe **Relator**.

A classe **Parecer** representa os pareceres dos professores que foram escolhidos como relatores e cada parecer está relacionado a um afastamento.

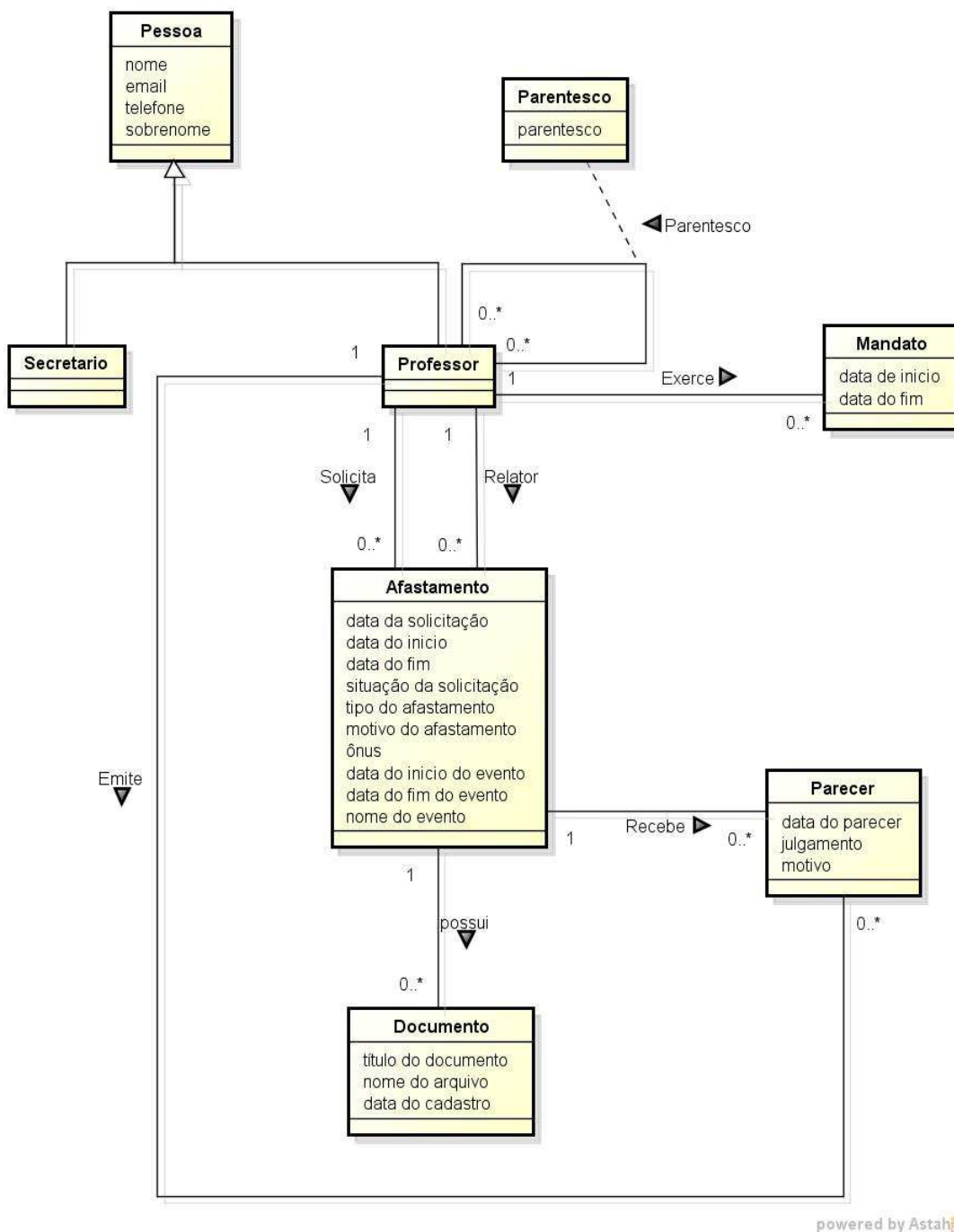


Figura 6 – Diagrama de Classes do SCAP (PRADO, 2015).

Além dos elementos visíveis no diagrama de classes, restrições de integridade têm um papel muito importante, pois elas complementam a informações de um modelo. Tais regras são capturadas a partir de modelos conceituais estruturais e são obtidas através das restrições relativas aos relacionamentos entre elementos de um modelo que não podem ser representadas graficamente. Tais regras devem ser documentadas junto ao modelo conceitual estrutural do sistema (FALBO, 2017a).

Listamos abaixo as restrições de integridade que foram obtidas por Duarte (2014) e também as acrescentadas por Prado (2015).

- Um professor não pode ser solicitado para dar um parecer sobre sua própria solicitação de afastamento;
- A data de início de um afastamento não pode ser posterior à data de fim do mesmo afastamento;
- A data de início de um mandato de professor não pode ser posterior à data de fim do mesmo mandato;
- Não pode haver mais de dois professores (chefe e subchefe de departamento) exercendo um mandato ao mesmo tempo;
- O secretário do departamento não pode abrir uma solicitação de afastamento;
- Um professor não pode ser relator de um afastamento solicitado por um parente.

## 4 Implementação SCAP

Este capítulo descreve as duas novas implementações do SCAP, cada uma utilizando um *framework* Controlador Frontal diferente, a saber: Spring MVC e Vaadin. Ambas foram baseadas nos resultados produzidos anteriormente por Duarte (2014) e Prado (2015) (apresentadas no capítulo anterior) e seguiram o método FrameWeb. Como já concluído por Duarte (2014), a mudança do controlador frontal impacta diretamente no modelo de navegação do FrameWeb, pois existe uma correspondência direta entre esse modelo e esta categoria de *framework*.

Assim, partido dos resultados citados acima, demos apenas um pequeno passo atrás, retornando à fase de projeto para refazer os modelos de navegação para que os mesmos possam incluir as características dos *frameworks* utilizados neste trabalho e guiar as novas implementações do SCAP. Deve-se ressaltar que foram aproveitados para as duas novas implementações os códigos-fonte da implementação do SCAP realizada por Prado (2015), em particular aqueles referentes ao modelo de domínio, aplicação e persistência. Nas próximas seções apresenta-se os novos modelos de navegação propostos para cada uma das novas implementações do SCAP.

Como o intuito deste trabalho não é a entrega da ferramenta SCAP e sim a aplicação do método FrameWeb com diferentes *frameworks*, foram implementados protótipos que não contemplam todas as especificações para a criação da WebApp: algumas regras de negócio não foram implementadas, tampouco os casos de uso relacionados à inserção de documentos.

Este capítulo possui a seguinte estrutura: a Seção 4.1 apresenta a implementação completa do SCAP com o Spring MVC (modelos, arquitetura, demonstração de telas) e, na sequência, a Seção 4.2 apresenta da mesma forma a implementação com Vaadin.

### 4.1 Spring MVC

Apresenta-se a seguir a implementação da aplicação Web SCAP, com o controlador frontal Spring MVC.

#### 4.1.1 Modelo de Navegação

A Figura 7 apresenta o modelo de navegação para o caso de uso **Cadastrar Solicitação**. Logo após realizar o login, o professor cadastrado no sistema é encaminhado para a página principal do sistema, representada no diagrama de navegação pela classe **Index**, podendo realizar uma solicitação de afastamento. A classe **AfastamentoController**

é a classe controladora que lida com a lógica de apresentação referente a este caso de uso.

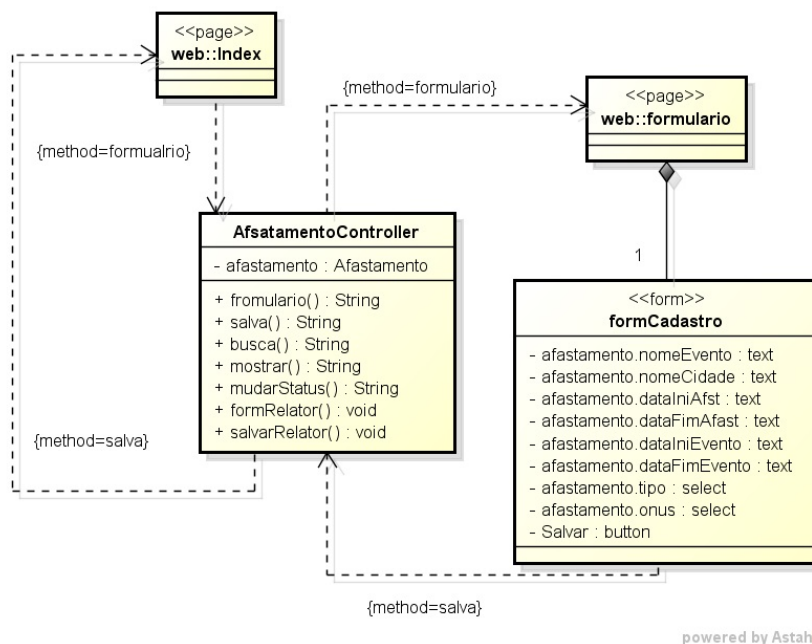


Figura 7 – Modelo de Navegação do caso de uso Cadastrar Solicitação.

Quando ele realiza essa solicitação, a classe controladora chama o método `formulario` que redireciona o usuário para uma página, que neste caso é também denominada `formulario`. Após preenchido o formulário, o usuário clica no botão `salvar` e a classe `AfastamentoController` invoca método `salvar` que, se comunica com a classe de aplicação responsável por esse caso de uso. Em seguida, o usuário retorna à página inicial.

Em alguns casos, o chefe do departamento deve encaminhar uma solicitação de afastamento para outro professor do DI, que será o relator da solicitação. A Figura 8 mostra o modelo de navegação para este cenário. A página `mostrar` é exibida quando o chefe do departamento solicita o afastamento em que se deseja definir o relator. Assim que ele clica no botão `Cadastrar um Relator`, a classe controladora `AfastamentoController` executa o método `formRelator` e encaminha para a página `formRelator`, onde é solicitada a matrícula do relator. Logo depois da confirmação, o método `salvarRelator` é acionado e a classe controladora solicita que a classe de aplicação, responsável por estes casos de uso, faça o registro.

O modelo de navegação da Figura 9, referente aos casos de uso **Manifestar-se contra Afastamento** e **Deferir Parecer**, representa a situação na qual algum professor quer se manifestar contra uma determinada solicitação de afastamento, podendo fazê-lo na página `mostrar`. Quando o botão `Definir um Parecer` é acionado, a classe controladora `ParecerController` chama o método `formulario`, que redireciona para a página em que se preenche as informações relacionadas à manifestação. Devidamente preenchido, o controlador recebe esses dados do formulário e os envia para a aplicação. Cabe ressaltar



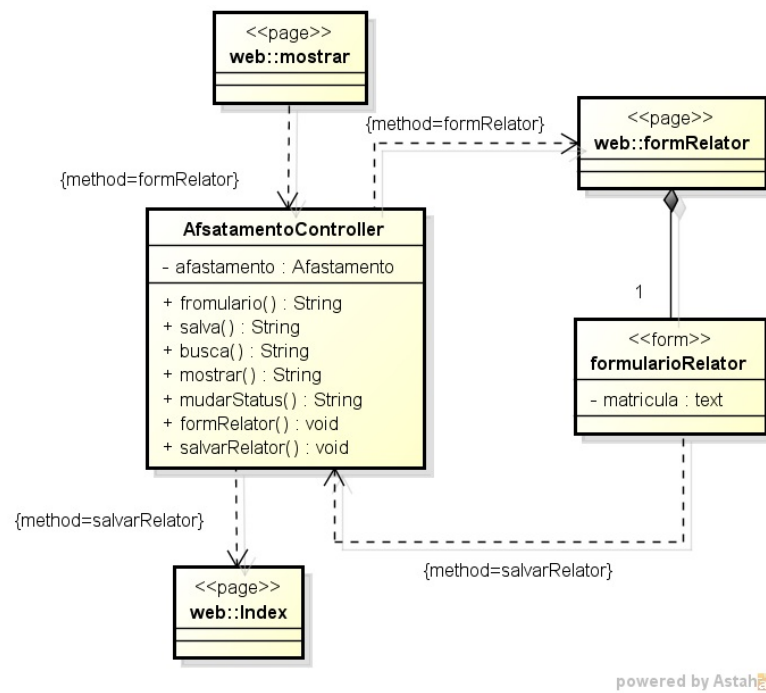


Figura 8 – Modelo de Navegação do caso de uso Encaminhar Afastamento.

que um relator é indicado para cada solicitação de afastamento e ele deve emitir um parecer que segue o mesmo rito de uma manifestação.

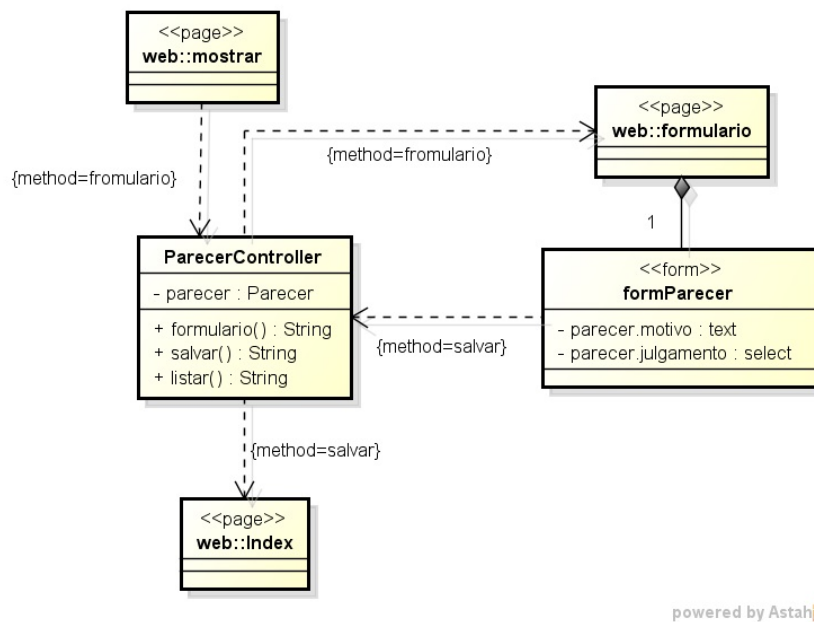


Figura 9 – Modelo de Navegação que cobre os casos de uso Manifestar-se Contra Afastamento e Deferir Parecer.

### 4.1.2 Tecnologias Utilizadas

O SCAP foi implementado utilizando a linguagem de programação Java, na plataforma Java EE 7 (Java Enterprise Edition 7). Para isso usou-se o Eclipse IDE como o ambiente de desenvolvimento com a intenção de facilitar a implementação do software, o servidor de aplicação usado foi o Apache Tomcat que é um servidor Java, utilizou-se também o Apache Maven, uma ferramenta de gerência de projetos que simplifica a obtenção das dependências do projeto.

Utilizou-se Java Persistence API (JPA) para persistência de dados, padronizando o mapeamento objeto-relacional por meio de uma interface de dados comum aos principais *frameworks* desta categoria. Utilizamos o *framework* Hibernate, que implementa o padrão JPA, é escrito em Java e visa facilitar a transição dos objetos usados no sistema para tabelas relacionais utilizadas no banco de dados. Ainda na parte relacionada à persistência, utilizamos o banco de dados MySQL que armazena dados em tabelas, seguindo o modelo relacional.

O Spring MVC foi o *framework* controlador frontal utilizado e, para facilitar e agilizar a implementação do sistema, o Spring framework foi utilizado para injeção de dependências entre as classe do sistema com objetivo de reduzir a quantidade de código utilizado.

Por fim, utilizamos o Bootstrap como *framework* decorador para melhorar a experiência de utilização para os usuários.

### 4.1.3 Pacotes SCAP

A Figura 10 apresenta como foram distribuídas as classes do SCAP: o sistema foi dividido em dois subsistemas Núcleo e Secretária, observados na fase de análise e especificação de requisitos. Em seguida, temos a subdivisões que respeitam a arquitetura em camadas proposta pelo FrameWeb, assim as classes foram separadas em 4 pacotes: Controle, Aplicação, Domínio e Persistência. Ressaltando que os arquivos do pacote de visão encontram-se no diretório **src/main/webapp** e são os responsáveis pela interação direta com o usuário.

Esta figura foi obtida do Package Explore da IDE Eclipse, que foi o ambiente de desenvolvimento utilizado na construção do projeto. Nela podemos observar toda a distribuição das pastas do sistema juntamente com alguns arquivos de configuração e as bibliotecas.

O pacote `br.ufes.scap.nucleo.aplicacao` inclui todas as classes de aplicação do subsistema Núcleo, elas são responsáveis pela execução dos casos de uso referentes a este subsistema e implementam sua lógica de negócio.

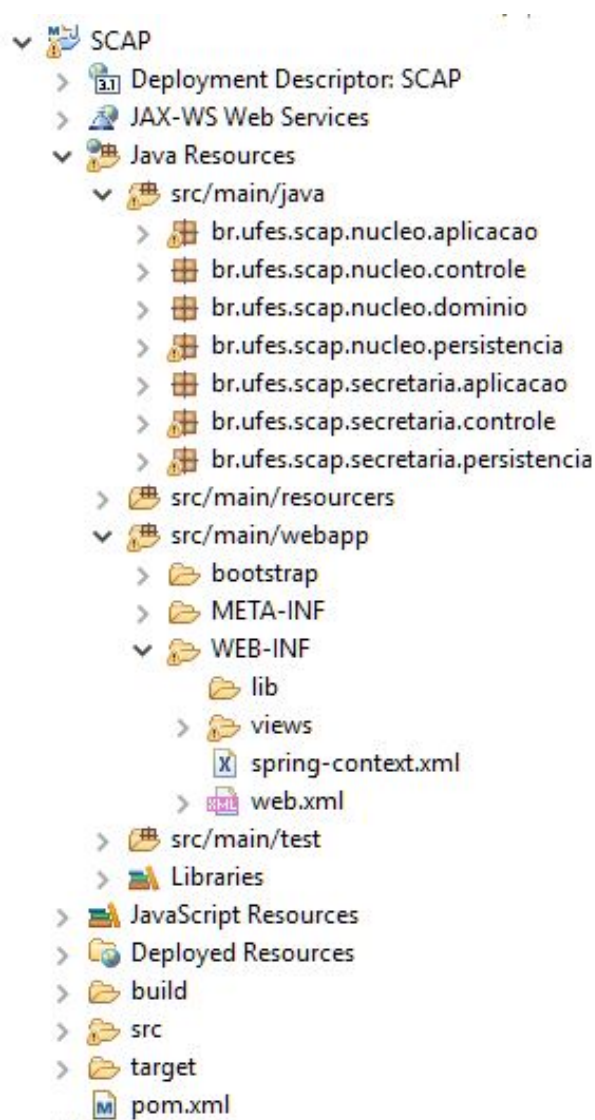


Figura 10 – Organização do código-fonte do projeto SCAP no Spring MVC.

A responsabilidade da comunicação entre o pacote visão e o pacote de aplicação é do pacote `br.ufes.scap.nucleo.controle` que contém as classes de controle que executam a tarefa anteriormente citada, tratando os estímulos enviados e recebidos pelo usuário ou pelo sistema.

O pacote `br.ufes.scap.nucleo.dominio` contém as classes de domínio de todo o sistema, que representam as entidades do mundo real e seus atributos.

E por último temos nesse subsistema o pacote `br.ufes.scap.nucleo.persistencia`, no qual encontra-se as classes responsáveis pela lógica de acesso e armazenamento de dados. Elas são responsáveis em fazer a persistência das classes de Domínio.

#### 4.1.4 Resultado da Implementação

Nesta seção apresentamos algumas capturas de tela de diferentes partes do sistema SCAP implementado com Spring MVC.

A Figura 11 exibe a tela de acesso ao sistema. Por meio dela todos o indivíduos cadastrados tem acesso às funcionalidades do sistema.

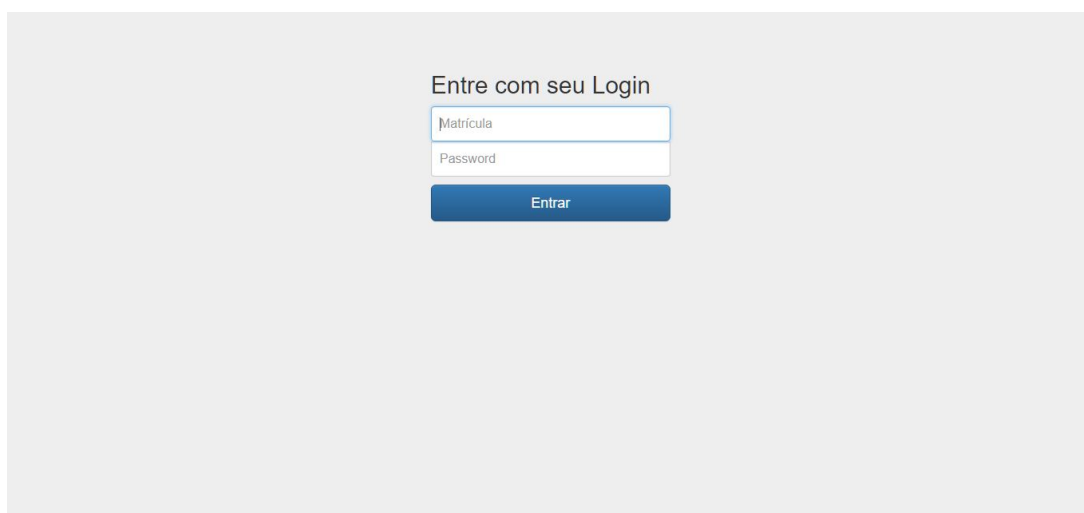
A imagem mostra a tela de login do sistema SCAP. No topo, há o título "Entre com seu Login". Abaixo dele, há dois campos de entrada: "Matrícula" e "Password". Abaixo dos campos, há um botão azul com o texto "Entrar".

Figura 11 – Tela de login do SCAP.

Quando um professor solicita um afastamento ele é encaminhado para página representada na Figura 12, na qual preenche o formulário com as informações referentes à solicitação de afastamento como, por exemplo, o nome do evento e da cidade, a data de início e data de fim do afastamento, dentre outras informações.

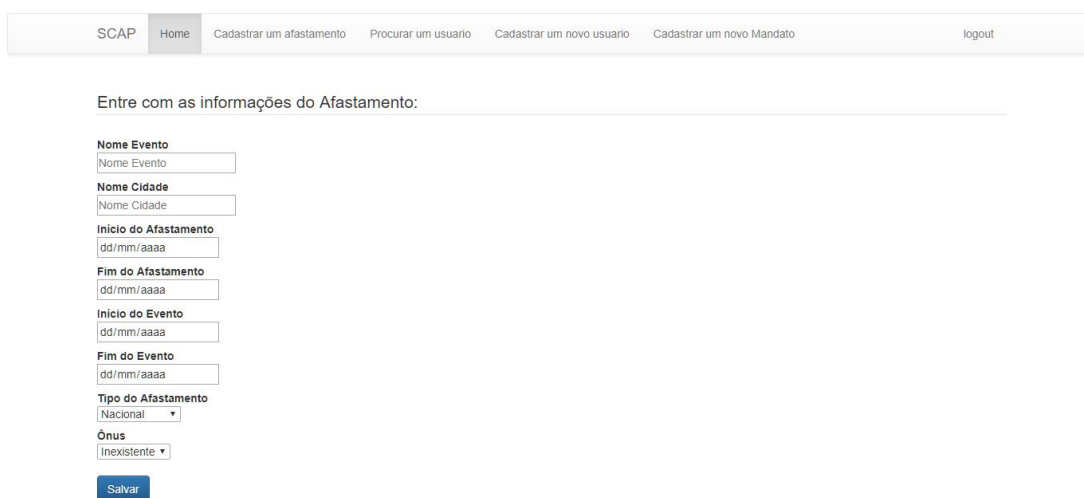
A imagem mostra a tela de cadastro de solicitação de afastamento. No topo, há uma barra de navegação com os links: "SCAP", "Home", "Cadastrar um afastamento", "Procurar um usuário", "Cadastrar um novo usuário", "Cadastrar um novo Mandato" e "logout". Abaixo, há o título "Entre com as informações do Afastamento:". O formulário contém os seguintes campos: "Nome Evento" (campo de texto), "Nome Cidade" (campo de texto), "Início do Afastamento" (campo de data dd/mm/aaaa), "Fim do Afastamento" (campo de data dd/mm/aaaa), "Início do Evento" (campo de data dd/mm/aaaa), "Fim do Evento" (campo de data dd/mm/aaaa), "Tipo do Afastamento" (menu suspenso com "Nacional" selecionado) e "Ônus" (menu suspenso com "Inexistente" selecionado). Abaixo dos campos, há um botão azul com o texto "Salvar".

Figura 12 – Tela com a formulário de cadastro de solicitação de afastamento.

A Figura 13 mostra as informações de uma determinada solicitação de afastamento. Nesta pagina ocorre a maioria das interações com a solicitação de afastamento: nela são

realizadas as ações de mudança de status do afastamento, definição do parecer, cadastro do relator, etc.

The screenshot shows a web interface for the SCAP system. At the top, there is a navigation bar with the following items: SCAP, Home, Cadastrar um afastamento, Procurar um usuário, Cadastrar um novo usuário, Cadastrar um novo Mandato, and logout. Below the navigation bar, there are three main sections:

- Solicitante:** Ramon Pereira, Matrícula: 1234, Email: ramon@ufes.br, Tel: 99999999
- Evento:** Banca de Mestrado, Cidade: Rio de Janeiro, Início: Segunda-feira, 30 de Novembro de 2015, Fim: Quinta-feira, 3 de Dezembro de 2015
- Afastamento:** Status: LIBERADO, Tipo: NACIONAL, Ônus: INEXISTENTE, Início: Sábado, 28 de Novembro de 2015, Fim: Sábado, 5 de Dezembro de 2015

Below these sections, there are five buttons: Mudar Status, Deferir um Parecer, Cadastrar um Relator, Cadastrar um Documento, and Ver pareceres. At the bottom, there is a table with the following data:

Titulo	Data Juntada	Download
listar	01/12/2015	Download
Parecer	25/05/2017	Download

Figura 13 – Tela que apresenta os dados de um pedido de afastamento.

## 4.2 Vaadin

Apresenta-se a seguir a implementação da aplicação Web SCAP, com o controlador frontal Vaadin.

### 4.2.1 Modelo de Navegação

A Figura 14 mostra o modelo de navegação do caso de uso **Cadastrar Usuário**. A classe controladora `PessoaController` é responsável pelo cadastro de usuários no sistema. Quando seu método `formulario` é chamado ele leva para um formulário da página `PessoaView`, onde devem ser preenchidas as informações referentes ao usuário. Em seguida o método `salvar` encaminha para que seja realizada a persistência desse usuário.

O modelo de navegação da Figura 15 representa uma consulta a uma solicitação de afastamento. O usuário se encontra na página `Principal` e o método `buscar` é chamado pelo controlador, encaminhando-o para a página `AfastBusca`, na qual informa-se o identificador do afastamento. Em seguida o usuário é redirecionado para a página `AfastMostra` se o resultado da busca for bem sucedido e esta página contém todas as informações do afastamento, caso contrario e exibida um notificação ao usuário que indica que o afastamento não existe.

Quando a solicitação de afastamento é internacional essa deve ser avaliada pela diretoria do CT em conjunto com Pró-reitoria de Pesquisa e Pós-Graduação (PRPPG) essas avaliações geram documento de aprovação ou rejeição que deve ser registrado no sistema. A Figura 16 apresenta o modelo de navegação de **Cadastrar Parecer CT/PRPPG**. Para

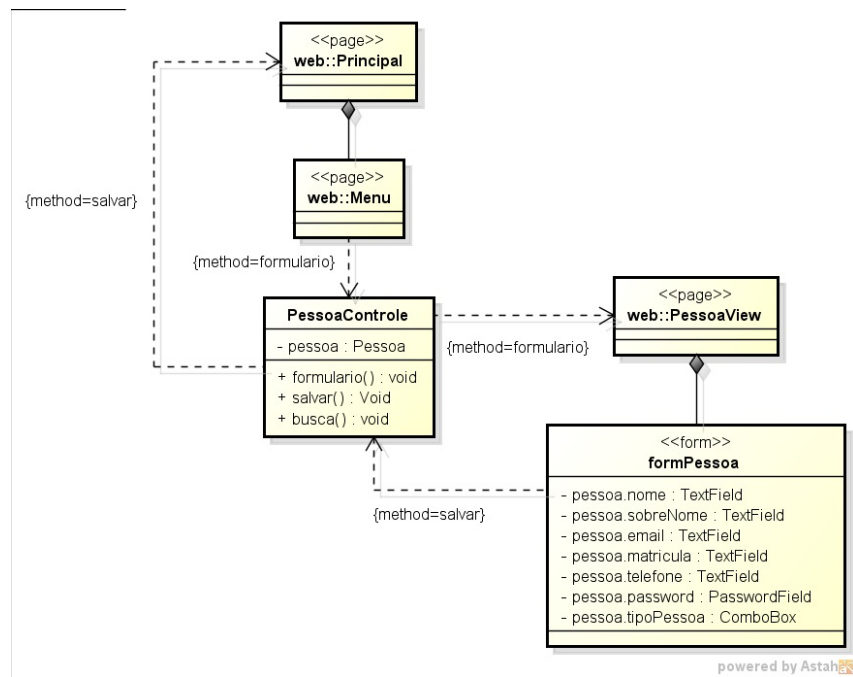


Figura 14 – Modelo de Navegação do caso de uso Cadastrar Usuário.

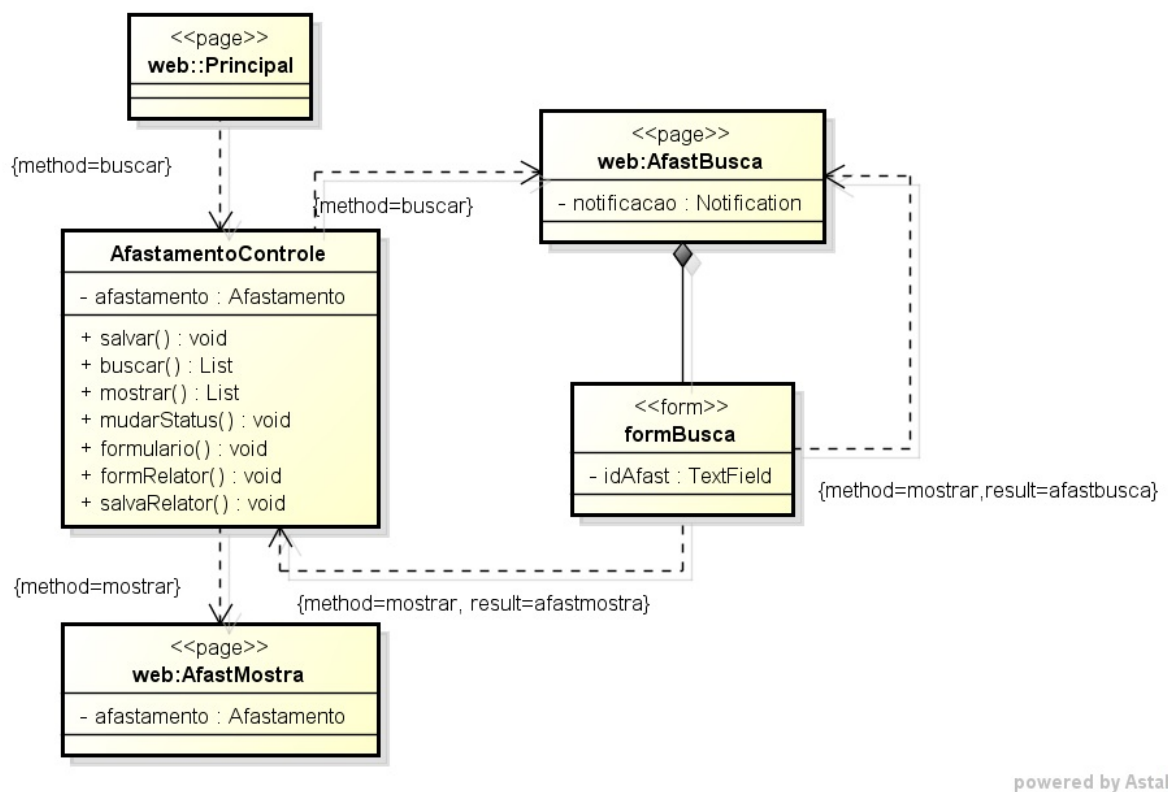


Figura 15 – Modelo de Navegação do caso de uso Consultar Afastamento.

cadastrar um documento, o secretario deve ir à tela que mostra os dados do afastamento, pagina *AfastMostra*, e acionar o botão *Cadastrar Documento*, que chama o método *formulario* da classe *DocumentoController* e o direciona para a página *DocumentoView*. Nesta página se realiza o *upload* do arquivo, que é encaminhado para aplicação para que o

documento possa ser persistido. Essa ação provoca a mudança de status do afastamento que também é realizada pelo secretário.

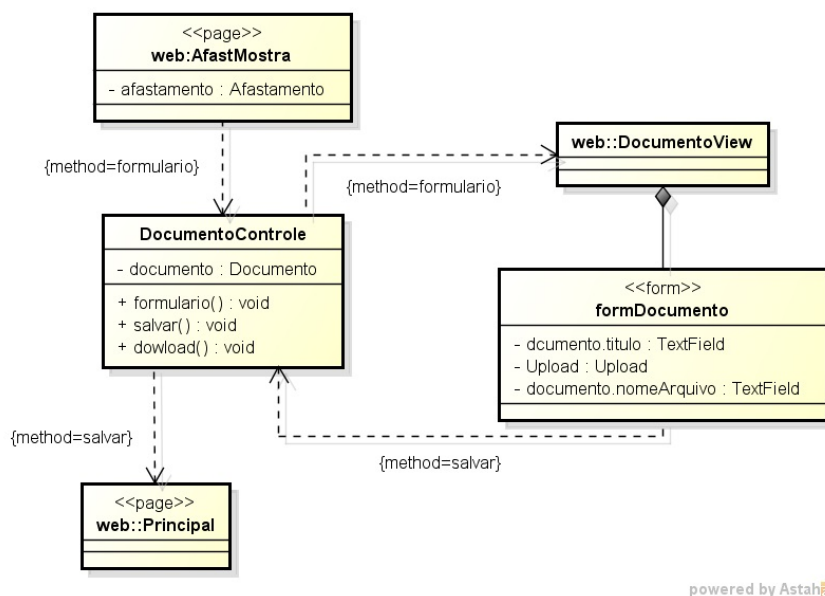


Figura 16 – Modelo de Navegação que cobre os Cadastrar Parecer CT e Cadastrar Parecer PRPPG.

## 4.2.2 Tecnologias Utilizadas

Manteve-se praticamente as mesmas tecnologias utilizadas na implementação do SCAP com o Spring MVC, sendo também implementado utilizando a plataforma Java juntamente com o ambiente de desenvolvimento Eclipse IDE, facilitando a produção do software. Utilizou-se ainda o Apache Tomcat como servidor de aplicação e a ferramenta de gerência de projetos Apache Maven.

As duas implementações compartilham o mesmo banco de dados MySQL que permite a persistência de dados em tabelas. Utilizou-se novamente o *framework* Hibernate em conjunto com a JPA para facilitar a transição dos objetos gerados no sistema para as tabelas relacionais utilizadas no banco de dados.

O *framework* controlador frontal utilizado foi o Vaadin, mantendo-se o Spring Framework para lidar com a injeção de dependências. O *framework* Vaadin é focado na interface com usuário, provendo assim um grande número de recursos gráficos para a implementação, logo não necessitando da utilização de um *framework* decorador que melhorasse a experiência do usuário.

## 4.2.3 Pacotes SCAP

A Figura 17, apresenta a distribuição das classes do SCAP na implementação do projeto utilizando o *framework* Vaadin. O sistema possui dois subsistemas, que foram



observados na fase de análise e especificação de requisitos, seguindo a arquitetura proposta pelo método FrameWeb as classes foram divididas em 5 pacotes: Visão, Controle, Aplicação, Domínio e Persistência.

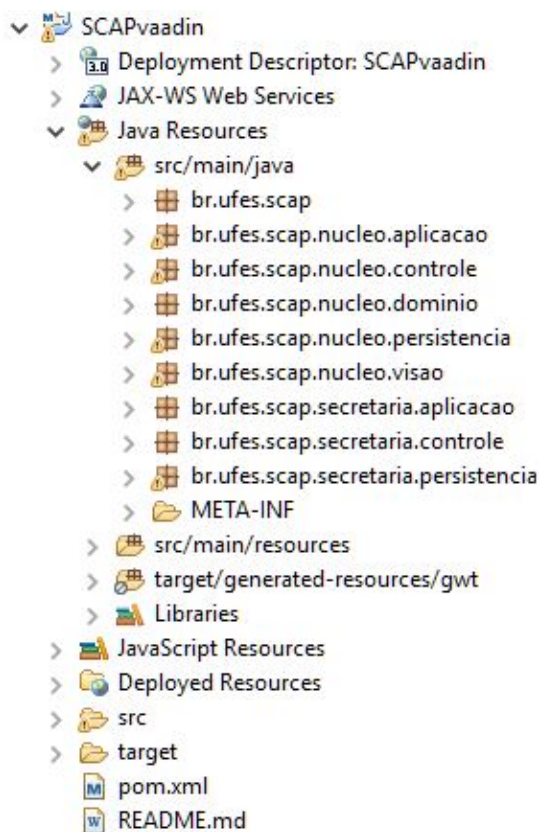


Figura 17 – Tela de login do SCAP

Podemos notar uma diferença na divisão dos pacotes da implementação com Spring MVC e Vaadin. No primeiro a visão era representada em um diretório separado dos outros pacotes, já no Vaadin as páginas são codificadas em Java facilitando para o desenvolvedor, pois é mesma linguagem escolhida para implementar o sistema. Logo, este pacote se encontra no mesmo diretório que os outros pacotes.

Os pacotes controle, aplicação e persistência do subsistema Núcleo possuem o mesmo objetivo anteriormente citados na seção 4.1.3. Visto que não abordamos anteriormente os pacotes do subsistema Secretaria, apresentamo-los nessa seção.

O pacote `br.ufes.scap.secretaria.aplicacao` contém as classes de aplicação do subsistema Secretaria, responsáveis pela execução dos casos de uso referentes a este subsistema e implementam sua lógica de negócio.

Logo depois encontra-se o pacote `br.ufes.scap.secretaria.controle` que contém as classes de controle que são responsáveis pela comunicação entre os pacotes de visão e aplicação, recebendo os estímulos do usuário pela visão e encaminhando para aplicação para execução dos casos de uso.



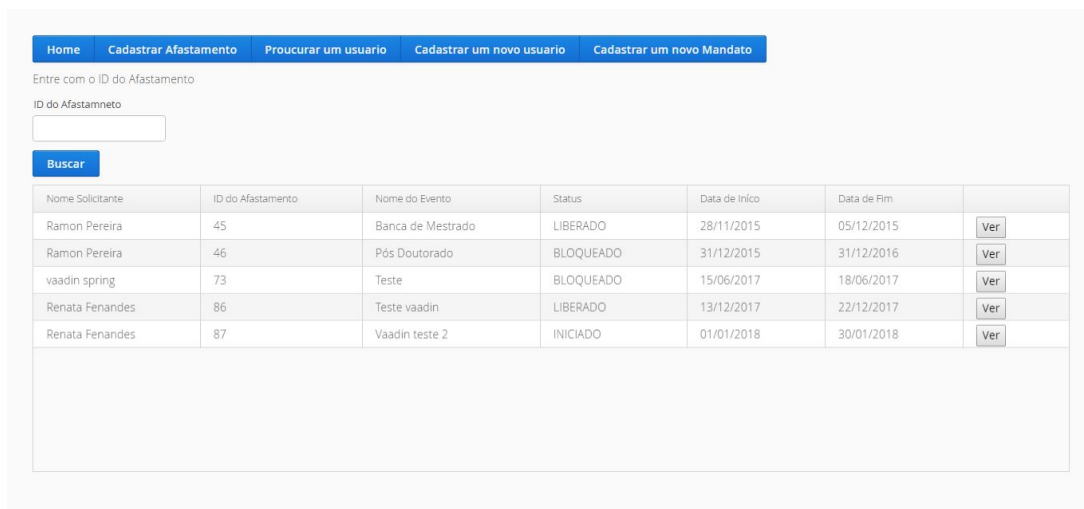
E por último temos nesse subsistema o pacote `br.ufes.scap.nucleo.persistencia`, no qual encontra-se as classes responsáveis pela lógica de acesso e armazenamento de dados. Elas são responsáveis em fazer a persistência das classes de Domínio.

Os pacotes `br.ufes.scap.nucleo.dominio` e `br.ufes.scap.nucleo.visao` possuem as todas as classes de domínio e visão respectivamente, porque são comuns a todo sistema por isso não houve divisão dessas classes entres os subsistemas.

#### 4.2.4 Resultado Implementação

Nesta seção apresentamos algumas capturas de tela de diferentes partes do sistema SCAP implementado com Vaadin.

A Figura 18 é a tela de busca da solicitação de afastamento. Nela, o usuário pode digitar o número identificador da solicitação e ela será encaminhada para a página que mostra as informações relativas àquela solicitação. Se o número identificador estiver incorreto, uma notificação é exibida nessa mesma tela. Para facilitar, a página contém uma lista de solicitações de afastamento com algumas informações sobre as mesmas, inclusive um botão que redireciona para a página onde as informações são exibidas por completo.



Nome Solicitante	ID do Afastamento	Nome do Evento	Status	Data de Início	Data de Fim	Ver
Ramon Pereira	45	Banca de Mestrado	LIBERADO	28/11/2015	05/12/2015	Ver
Ramon Pereira	46	Pós Doutorado	BLOQUEADO	31/12/2015	31/12/2016	Ver
vaadin spring	73	Teste	BLOQUEADO	15/06/2017	18/06/2017	Ver
Renata Fenandes	86	Teste vaadin	LIBERADO	13/12/2017	22/12/2017	Ver
Renata Fenandes	87	Vaadin teste 2	INICIADO	01/01/2018	30/01/2018	Ver

Figura 18 – Tela formulário de busca e lista de solicitações de afastamento.

O formulário exibido na Figura 19 é referente ao cadastro de usuários no sistema e o secretário do departamento é o responsável pelo preenchimento dessas informações. Este usuários podem ser classificados em dois tipos: professor ou secretário.

A Figura 20 mostra todas as informações da solicitação de afastamento. Nessa tela, podem ser executadas as ações de cadastro de documento. Ela também contém a lista de documentos cadastrados para a solicitação. O chefe do departamento pode cadastrar um relator para o afastamento por meio desta tela. Já o professor, além de consultar a situação da sua solicitação, pode cancelá-la por meio desta página, além de ter acesso a outras funções.

Home   Cadastrar Afastamento   Procurar um usuario   Cadastrar um novo usuario   Cadastrar um novo Mandato

Insira os dados da Pessoa

Nome:

Sobre Nome:

Email:

Matricula:

Telefone:

Password:

Professor

Salvar

Figura 19 – Tela com o formulário de cadastro de usuário.

Home   Cadastrar Afastamento   Procurar um usuario   Cadastrar um novo usuario   Cadastrar um novo Mandato

Solicitante	Evento	afastamento
vaadin.spring Matricula: 2222 Email: vaadin@spring Tel: 9988776655	Teste Cidade: Vitória Inicio: Sexta-feira, 16 de Junho de 2017 Fim: Sábado, 17 de Junho de 2017	Status: BLOQUEADO Tipo: NACIONAL Ônus: INEXISTENTE Inicio: Quinta-feira, 15 de Junho de 2017 Fim: Domingo, 18 de Junho de 2017

Mudar Status   Definir um Parecer   Cadastrar um Relator   Cadastrar um Documento   Ver pareceres

Titulo	Data Juntada	Download arquivo

Figura 20 – Tela que apresenta os dados de um pedido de afastamento.

## 5 Considerações Finais

Este último capítulo descreve a experiência da utilização do FrameWeb com os *frameworks* controlador frontal Spring MVC e Vaadin, para a implementação do Sistema de Controle de Afastamento e Professores (SCAP), além de discutir também a aplicabilidade do método com a mudança de *frameworks* proposta.

Deve-se ressaltar que o uso de *frameworks* provê uma infraestrutura base na construção de Sistemas de Informação Web (*Web-based Information Systems* ou WISs), o que auxilia na construção mais rápida do software. Com isso o método FrameWeb atinge um ponto específico, assumindo a utilização de *frameworks* durante a construção da aplicação, define uma arquitetura básica para WIS e propõe modelos de projeto que se aproximam da implementação do sistema usando *frameworks*.

Para atender o objetivo proposto, o autor deste trabalho tratou o desenvolvimento como uma tarefa de mudança de tecnologia, como por exemplo se uma empresa resolvesse adotar um outro *framework* no desenvolvimento de seus projetos, no caso a mudança do *framework* controlador frontal. Sendo assim, considerou que o software já tinha sido projetado e implementado, era necessário realizar um estudo de todo o processo de desenvolvimento e executar alteração da tecnologia.

Depois de todo o estudo realizado, gradualmente foram atingindo-se os objetivos propostos, realizando a criação dos modelos de navegação para os *frameworks* adotados para a construção do sistema. A partir destes, juntamente com todos os modelos anteriormente levantados, partiu-se para a parte de aplicação prática na fase de implementação e, assim, atingiu-se o principal objetivo, que era a implementação de duas novas versões do SCAP com os controladores frontais Spring MVC e Vaadin.

Ao final do processo, concluímos que o método FrameWeb agilizou e modularizou a construção da aplicação, tornando mais fácil a sua implementação, pois como a mudança proposta impacta apenas no modelo de navegação e somente as classes referentes a esse modelo precisaram ser recriadas, todas as outras partes da estrutura do projeto não precisaram ser alteradas.

Uma das dificuldades encontradas na realização do trabalho foi o desconhecimento que o autor possuía em relação a utilização dos *frameworks*, acreditando assim que o método FrameWeb o auxiliou nesse quesito pois a integração proposta pelo método com os *frameworks* facilitou o aprendizado e agregação de conhecimento.

A alteração no modelo de navegação afeta diretamente as classes de controle e visão, a implementação da aplicação com Spring MVC alterou apenas as classes de controle

pois este conseguiu utilizar toda a parte da visão utilizada na construção anterior com VRaptor, entretanto com o Vaadin todas as duas tiveram que ser alteradas pois esse caracteriza-se pela construção da interface da aplicação utilizando a linguagem Java para produção dos artefatos de visão, o que facilitou, já que foi a mesma linguagem escolhida para implementação de todo o sistema. Assim, pôde-se perceber que alguns *frameworks* possuem características em comum e outros não.

Por fim, ressalta-se que esse trabalho mostra que o FrameWeb pode ser aplicado aos *frameworks* Spring MVC e Vaadin, o que pode ajudar na disseminação da utilização do método.

Como trabalhos futuros, imagina-se uma nova implementação do SCAP usando uma coleção diferente de *frameworks* e tecnologias na plataforma Java EE 7 e uma aplicação de método FrameWeb usando uma plataforma e linguagem de programação diferente do Java, como por exemplo PHP.

# Referências

- ALUR, D.; CRUPI, J.; MALKS, D. *Core J2EE Patterns: Best Practices and Design Strategies*. [S.l.]: Prentice Hall PTR, 2003. (Core Series). ISBN 9780131422469. Citado 2 vezes nas páginas [19](#) e [22](#).
- BLAHA, M.; RUMBAUGH, J. *Modelagem e projetos baseados em objetos com UML 2*. [S.l.]: Elsevier, 2006. ISBN 9788535217537. Citado na página [13](#).
- CAVALCANTI, L. *VRaptor: Desenvolvimento ágil para web com Java*. [S.l.]: Casa do Código, 2014. ISBN 9788566250800. Citado na página [12](#).
- CONALLEN, J. *Building Web Applications with UML*. 2ª. ed. [S.l.]: Addison-Wesley, 2002. (Addison-Wesley object technology series). ISBN 9780201730388. Citado 2 vezes nas páginas [17](#) e [22](#).
- DEITEL, H.; DEITEL, M. *Java: como programar*. 8ª. ed. [S.l.]: PRENTICE HALL BRASIL, 2010. ISBN 9788576050193. Citado na página [13](#).
- DEMICHIEL, L.; SHANNON, B. *Java™ Platform, Enterprise Edition (Java EE) Specification, v7*. [s.n.], 2013. Disponível em: <http://jcp.org/en/jsr/detail?id=342>. Citado na página [12](#).
- DUARTE, B. B. *Aplicação do Método FrameWeb no Desenvolvimento de um Sistema de Informação na Plataforma Java EE 7*. Monografia (Projeto de Graduação) — Departamento de Informática, Universidade Federal do Espírito Santo, Vitória (ES), Brasil, 2014. Citado 6 vezes nas páginas [12](#), [13](#), [14](#), [24](#), [28](#) e [30](#).
- ELMASRI, R.; NAVATHE, S. *Sistemas de banco de dados*. [S.l.]: PEARSON BRASIL, 2011. ISBN 9788579360855. Citado na página [13](#).
- FALBO, R. A. *Engenharia de Requisitos*. [s.n.], 2017. 178 p. Disponível em: [https://inf.ufes.br/~falbo/files/ER/Notas\\_Aula\\_Engenharia\\_Requisitos.pdf](https://inf.ufes.br/~falbo/files/ER/Notas_Aula_Engenharia_Requisitos.pdf). Citado 4 vezes nas páginas [13](#), [25](#), [27](#) e [28](#).
- FALBO, R. A. *Projeto de Sistemas de Software*. [s.n.], 2017. 135 p. Disponível em: [https://inf.ufes.br/~falbo/files/PSS/Notas\\_Aula\\_Projeto\\_Sistemas\\_2017.pdf](https://inf.ufes.br/~falbo/files/PSS/Notas_Aula_Projeto_Sistemas_2017.pdf). Nenhuma citação no texto.
- FOWLER, M. *Patterns of Enterprise Application Architecture*. [S.l.: s.n.], 2002. (Addison-Wesley). ISBN 0321127420. Citado 2 vezes nas páginas [7](#) e [21](#).
- GAMMA, E. et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. [S.l.]: Pearson Education, 1994. (Addison-Wesley Professional Computing Series). ISBN 9780321700698. Citado na página [18](#).
- GINIGE, A.; MURUGESAN, S. Web engineering: An introduction. *IEEE multimedia*, IEEE, v. 8, n. 1, p. 14–18, 2001. Citado na página [16](#).
- GRÖNROOS, M. Book of vaadin: Vaadin framework 8. *Vaadin Ltd*, 2017. Disponível em: <https://vaadin.com/book>. Citado na página [20](#).

- JOHNSON, R. et al. The spring framework reference documentation. *Interface*, v. 21, 2004. Disponível em: <<https://docs.spring.io/spring-framework/docs/4.3.7.RELEASE/spring-framework-reference/html/>>. Citado na página 20.
- KOCH, N. et al. Extending uml to model navigation and presentation in web applications. In: YORK, ENGLAND. *Proceedings of Modelling Web Applications in the UML Workshop*. [S.l.], 2000. Citado na página 22.
- MALDONADO, J. C. et al. Padrões e frameworks de software. *Notas Didáticas, Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, ICMC/USP, São Paulo, SP, Brasil*, 2002. Citado na página 18.
- MARTINS, B. F. S. *Uma abordagem dirigida a modelos para o projeto de Sistemas de Informação Web com base no método FrameWeb*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2016. Citado 2 vezes nas páginas 21 e 22.
- MURUGESAN, S. et al. Web engineering: A new discipline for development of web-based systems. In: *Web Engineering*. [S.l.]: Springer, 2001. p. 3–13. Citado na página 15.
- OLSINA, L.; LAFUENTE, G.; ROSSI, G. Specifying quality characteristics and attributes for websites. *Web Engineering*, Springer, p. 266–278, 2001. Citado na página 17.
- PASTOR, O.; FONS, J.; PELECHANO, V. Oows: A method to develop web applications from web-oriented conceptual models. In: *International Workshop on Web Oriented Software Technology (IWWOST)*. [S.l.: s.n.], 2003. p. 65–70. Citado na página 17.
- PERUCH, L. A. *Aplicação e análise do Método FrameWeb com Diferentes Frameworks Web*. Monografia (Trabalho de Conclusão de Curso) — Departamento de Informática, Universidade Federal do Espírito Santo, Vitória (ES), Brasil, 2007. Citado na página 27.
- PRADO, R. C. d. *Aplicação do método FrameWeb no desenvolvimento de um sistema de informação utilizando o framework VRaptor 4*. Monografia (Projeto de Graduação) — Departamento de Informática, Universidade Federal do Espírito Santo, Vitória (ES), Brasil, 2015. Citado 12 vezes nas páginas 7, 8, 12, 13, 14, 19, 24, 25, 26, 27, 28 e 30.
- PRESSMAN, R. S. *Engenharia de Software - Uma abordagem profissional*. 7ª edição. ed. [S.l.]: McGraw-Hill, 2011. ISBN 9788563308337. Citado 4 vezes nas páginas 12, 13, 15 e 17.
- SCHWABE, D.; ROSSI, G. An object oriented approach to web-based application design. In: . Wiley and Sons, 1998. p. 207–225. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.57>>. Citado na página 17.
- SOUZA, V. E. S. *FrameWeb: um Método baseado em Frameworks para o Projeto de Sistemas de Informação Web*. Dissertação (Mestrado em Informática) — Universidade Federal do Espírito Santo, Vitória, ES, Brasil, 2007. Citado 8 vezes nas páginas 7, 12, 13, 18, 19, 20, 21 e 22.