



Rodolfo Costa do Prado

**Aplicação do método FrameWeb no
desenvolvimento de um sistema de informação
utilizando o framework VRaptor 4**

Vitória, ES

2015

Rodolfo Costa do Prado

**Aplicação do método FrameWeb no desenvolvimento de
um sistema de informação utilizando o framework
VRaptor 4**

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

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Departamento de Informática

Orientador: Prof. Dr. Vítor E. Silva Souza

Coorientador: Beatriz Franco Martins Souza

Vitória, ES

2015

Rodolfo Costa do Prado

Aplicação do método FrameWeb no desenvolvimento de um sistema de informação utilizando o framework VRaptor 4/ Rodolfo Costa do Prado. – Vitória, ES, 2015-

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

Orientador: Prof. Dr. Vítor E. Silva Souza

Monografia (TCC) – Universidade Federal do Espírito Santo – UFES
Centro Tecnológico
Departamento de Informática, 2015.

1. Palavra-chave1. 2. Palavra-chave2. I. Souza, Vítor Estêvão Silva. II. Universidade Federal do Espírito Santo. IV. Aplicação do método FrameWeb no desenvolvimento de um sistema de informação utilizando o framework VRaptor 4

CDU 02:141:005.7

Rodolfo Costa do Prado

Aplicação do método FrameWeb no desenvolvimento de um sistema de informação utilizando o framework VRaptor 4

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

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

Prof. Dr. Vítor E. Silva Souza
Orientador

Professor
Convidado 1

Professor
Convidado 2

Vitória, ES
2015

Agradecimentos

Agradeço à minha mãe, por ter me amado e educado com carinho a minha vida toda. Ao meu pai pelos 10 anos de sacrifício em uma terra estrangeira garantindo que nunca me faltasse nada.

Ao meu irmão pelo exemplo de como devo buscar ser sempre uma pessoa melhor. Aos meus amigos pela descontração mesmo nos piores momentos, e a meus professores pelo conhecimento adquirido nessa jornada.

E principalmente ao amor da minha vida Ellen, por ser minha luz até nas horas mais escuras.

*“You get what anybody gets - you get a lifetime.”
(Neil Gaiman, The Sandman)*

Resumo

Devido à ascensão da Internet como um dos principais meios de comunicação, houve uma demanda por software de qualidade funcionando nesta plataforma. Porém esses programas se tornaram complexos demais para serem desenvolvidos no modo *ad hoc* usado previamente. Assim, foi necessário o uso de conceitos de Engenharia de Software no desenvolvimento de sistemas e aplicações que funcionem na Web.

Foi proposto então o método FrameWeb (*Framework-based Design Method for Web Engineering*), que sugere a utilização de uma série de frameworks, juntamente com várias recomendações, que agilizam as principais fases de desenvolvimento. Em sua proposta original foram utilizados apenas um framework de cada categoria, de modo que não é possível saber se o método é aplicável quando uma coleção diferente de frameworks são usados.

Para testar a eficácia do método com diferentes frameworks foi desenvolvido uma WebApp — um Sistema de Controle de Afastamentos de Professores, ou SCAP — com um conjunto diferente de frameworks, baseados na plataforma Java EE 7. O objetivo deste trabalho é reimplementar o SCAP testando o método FrameWeb com o framework controlador frontal VRaptor 4 ao invés dos frameworks utilizados em experimentos anteriores com o método.

Palavras-chave: Engenharia Web. WebApp. FrameWeb. VRaptor. Java EE 7.

Lista de ilustrações

Figura 1 – Processo de desenvolvimento de software proposto por Conallen (2002)	18
Figura 2 – Diagrama representando a arquitetura MVC.	20
Figura 3 – Representação de um Controlador Frontal na Web (SOUZA, 2007).	21
Figura 4 – Funcionamento de um framework de Injeção de Dependências (SOUZA, 2007).	22
Figura 5 – Um processo de desenvolvimento de software simples sugerido por FrameWeb (SOUZA, 2007).	23
Figura 6 – Arquitetura padrão para WIS baseada no padrão arquitetônico Service Layer (FOWLER, 2002).	24
Figura 7 – Diagrama de Casos de Uso do subsistema Núcleo.	28
Figura 8 – Diagrama de Casos de Uso do subsistema Secretaria.	29
Figura 9 – Diagrama de Classes do SCAP.	32
Figura 10 – projeto SCAP visto no Package Explorer do Eclipse IDE.	36
Figura 11 – Modelo de Domínio do SCAP.	37
Figura 12 – Tipos enumerados do SCAP.	38
Figura 13 – Tipos enumerados do SCAP.	39
Figura 14 – Modelo de Navegação para o caso de uso Cadastrar Afastamento.	40
Figura 15 – Modelo de Navegação para o caso de uso Consultar Afastamento.	41
Figura 16 – Modelo de aplicação do subsistema Núcleo.	42
Figura 17 – Modelo de aplicação do subsistema Secretaria.	43
Figura 18 – Modelo de Persistência do subsistema Núcleo.	44
Figura 19 – Modelo de Persistência do subsistema Secretaria.	44
Figura 20 – Tela de login do SCAP.	46
Figura 21 – Tela de login incorreto do SCAP.	46
Figura 22 – Tela contendo a Lista de Afastamentos do SCAP.	47
Figura 23 – Tela que apresenta os dados de um Pedido de Afastamento no SCAP.	47
Figura 24 – Formulário de cadastro de afastamento no SCAP.	48
Figura 25 – modelo de navegação proposto para o caso de uso Consular Solicitação, com adição de um novo estereótipo ao FrameWeb.	52

Lista de tabelas

Tabela 1 – Atores do SCAP	27
Tabela 2 – Esteriótipos utilizados nos modelos de navegação do SCAP.	40

Lista de abreviaturas e siglas

UML	Unified Modeling Language
SCAP	Sistema de Controle de Afastamento e Professores

Sumário

1	INTRODUÇÃO	12
1.1	Objetivos	13
1.2	Metodologia	13
1.3	Organização da Monografia	14
2	FRAMEWEB, FRAMEWORKS E A ENGENHARIA WEB	15
2.1	Engenharia Web	16
2.2	Frameworks	18
2.2.1	Frameworks MVC	19
2.2.2	Frameworks de Mapeamento Objeto/Relacional	20
2.2.3	Frameworks de Injeção de Dependência	22
2.3	O Método FrameWeb	22
2.3.1	Arquitetura de software do FrameWeb	23
2.3.2	Linguagem de modelagem de FrameWeb	24
3	ESPECIFICAÇÃO DE REQUISITOS E ANÁLISE DO SCAP	26
3.1	Descrição do Escopo	26
3.2	Modelo de Casos de Uso	27
3.3	Análise do SCAP	29
3.3.1	Modelagem de Classes	30
3.3.2	Restrições de Integridade	30
4	PROJETO E IMPLEMENTAÇÃO	33
4.1	Arquitetura do Sistema	33
4.1.1	Tecnologia Utilizadas	33
4.2	Frameworks Utilizados	34
4.2.1	VRaptor 4	34
4.2.2	CDI	34
4.2.3	Hibernate + JPA	35
4.2.4	Pacotes do SCAP	35
4.3	Modelo de Domínio	36
4.4	Modelo de Navegação	38
4.5	Modelo de Aplicação	42
4.6	Modelo de Persistência	43
4.7	Implementação do SCAP	45

5	CONCLUSÃO	50
5.1	VRaptor e o FrameWeb	51
5.2	Trabalhos Futuros	51
	Referências	53

1 Introdução

Com a rápida ascensão da Internet como um dos principais meios de comunicação existente, através de serviços como a *World Wide Web* (Web), criou-se uma nova dimensão na utilização do computador, o que fez surgir a demanda por softwares de alta qualidade para suprir requisitos como segurança, velocidade, compatibilidade, usabilidade etc.

A fim de atender tal necessidade, houve um aumento na complexidade dos softwares criados para Internet, eis que o modo *ad hoc* de desenvolvimento antes utilizado, não seria mais suficiente. Passou-se, então, a empregar conceitos de Engenharia de Software e ferramentas para auxiliar no desenvolvimento de sistemas e aplicações que funcionem na Web (WebApps - *Web Applications*). Nesse contexto foi criada a Engenharia Web (PRESSMAN, 2011).

Recentemente, 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 WebApp, juntamente com varias recomendações para aumentar a agilidade nas principais fases de desenvolvimento (requisitos, análise, projeto e outras).

O método apresenta, então, um perfil UML (extensão da linguagem UML) com quatro tipos de modelos para a fase de projeto. Tais modelos representam conceitos utilizados por algumas categorias de frameworks, facilitando, dessa forma, a comunicação entre as equipes de projetistas e a de programadores durante o desenvolvimento da aplicação.

O FrameWeb originalmente considera algumas categorias de frameworks, dentro das quais existem várias opções a serem usadas na implementação de uma WebApp. Em sua proposta original, foram feitos experimentos com apenas um framework de cada categoria, de modo que não é possível saber se o método é adequado a um conjunto diferente de frameworks dentro das mesmas categorias. Em seu trabalho de conclusão de curso,

Duarte (2014) desenvolveu uma WebApp — um Sistema de Controle de Afastamentos de Professores, ou SCAP — com o método FrameWeb, utilizando um conjunto diferente de frameworks, baseado na plataforma Java EE 7 (DEMICHIEL; SHANNON, 2013). O trabalho serve de base para uma análise do método com vistas a uma futura adequação a uma gama maior de frameworks.

O objetivo deste trabalho é, a partir dos requisitos levantados em (DUARTE, 2014), realizar uma nova implementação do SCAP aplicando o FrameWeb, porém utilizando o framework VRaptor 4 (CAVALCANTI, 2014) ao invés das tecnologias padrão do Java EE

para controlador frontal (a saber, o JSF), propondo melhorias e modificações nos modelos, caso seja necessário, para se adequar aos novos frameworks utilizados.

1.1 Objetivos

O objetivo deste trabalho é aplicar o método FrameWeb (SOUZA, 2007) em uma nova implementação do SCAP (Sistema de Controle de Afastamentos de Professores), a partir dos requisitos levantados em (DUARTE, 2014), complementando o que foi feito e demonstrando quais evoluções podem ser feitas ao FrameWeb.

Na primeira implementação do SCAP foram utilizados os frameworks da plataforma Java EE 7 (DEMICHIEL; SHANNON, 2013), em especial seus componentes, CDI, JSF e Facelets, ao invés dos frameworks propostos originalmente pelo FrameWeb, respectivamente: Spring, Struts2 e Sitemesh.

Nesta segunda implementação foi utilizado o framework MVC VRaptor 4 (CAVALCANTI, 2014) para controlador frontal. Assim foi feita uma comparação de como o mesmo se encaixa ao FrameWeb em relação aos frameworks utilizados no desenvolvimento da primeira versão do SCAP, propondo melhorias e modificações nos modelos do FrameWeb, quando necessário.

No desenvolvimento deste projeto, foram exercitados conhecimentos aprendidos ao longo do curso de Ciência da Computação, em especial das disciplinas de Engenharia de Software, Engenharia de Requisitos, Projeto de Sistemas de Software, Programação III e Banco de Dados.

1.2 Metodologia

O trabalho descrito neste documento foi dividido nas seguintes atividades:

- Pesquisa: o método FrameWeb e os diversos frameworks nos quais ele foi inicialmente testado foram estudados, bem como a plataforma Java EE 7 e o desenvolvimento de WebApps na mesma;
- Análise do Problema Proposto: foi revisada a análise feita por Duarte (2014) do SCAP, mantendo os Professores e Secretários do Departamento de Informática da UFES como *stakeholders*, porém realizando pequenas mudanças nos requisitos levantados previamente;
- Projeto: a partir dos resultados obtidos no item acima, foi desenvolvido um novo projeto do sistema SCAP, utilizando a metodologia e os modelos de projeto propostos por FrameWeb;

- Codificação: seguindo a arquitetura de desenvolvimento proposta pelo método, foi implementada a aplicação SCAP utilizando o framework VRaptor;
- Finalização: a finalização do projeto contou com os testes finais do sistema Web além dos pequenos ajustes ou melhorias realizadas para adequação ao FrameWeb;
- Apresentação do Projeto: apresentação final, na qual o projeto e a ferramenta foram entregues e demonstrados.

1.3 Organização da Monografia

Essa monografia é dividida em 5 capítulos incluindo a introdução.

No capítulo 2 realiza-se um levantamento dos principais temas abordados ao longo deste trabalho que são: FrameWeb, frameworks e a Engenharia Web.

No capítulo 3 é feita a especificação e análise dos requisitos do SCAP (Sistema de Controle de Afastamento de Professores) sistema Web que foi desenvolvido utilizando o FrameWeb.

No capítulo 4 apresenta-se o resultado da aplicação do FrameWeb no projeto, podem ser vistos os modelos propostos e a implementação dos mesmos com as tecnologias utilizadas.

No capítulo 5 tem-se as conclusões retiradas deste trabalho juntamente com as melhorias propostas para o método FrameWeb.

2 FrameWeb, Frameworks e a Engenharia Web

Em seu início a Web era muito diferente do que vemos hoje, em um período de 20 a 15 anos ela passou de um conjunto de páginas estáticas desenvolvidas de maneira *ad hoc* para ser a principal plataforma de comunicação que já desenvolvemos. Isso se deve a um ciclo curioso que pode ser observado, quanto mais usuários existem na Web, maior é o número de funcionalidades são criadas para a mesma, e como existe mais utilidade na Web, cresce o número de usuários.

O que vemos hoje é que a Internet é parte essencial em nossas vidas, ela é usada como entretenimento, trabalho e plataforma para prestação de serviços. Para que toda essa gama de atividades fossem transferidas para a Web, foi necessária a mudança de um conjunto páginas estáticas para verdadeiros sistemas complexos desenvolvidos especialmente para esta plataforma.

Tornou-se essencial a aplicação de conceitos existentes na Engenharia de Software, adaptados para essa nova plataforma, no desenvolvimento das aplicações Web. Características do ambiente Web, como concorrência, carga imprevisível, disponibilidade, sensibilidade ao conteúdo, evolução dinâmica, imediatismo, segurança e estética (PRESSMAN, 2011) deram origem a uma nova sub-área da Engenharia de Software, a Engenharia Web.

Existem páginas atualmente na Internet que são verdadeiros sistemas de informação dentro da Web como, por exemplo, serviços do governo, sites de compras virtuais, ambientes corporativos, entre outros. Chamaremos aqui tais sistemas de WIS (*Web-based Information Systems*) e ao conjunto amplo de todas as aplicações Web como WebApps (abreviação de *Web Applications*).

Surgem então frameworks que vieram para facilitar e agilizar na produção das WebApps, fornecendo uma estrutura mais sólida para desenvolvimento. Com o passar do tempo a utilização desses frameworks se tornou o padrão na prática, devido tanto à agilidade que eles fornecem ao processo de criação quanto à garantia de qualidade, dada sua larga escala de utilização.

Neste capítulo veremos a adaptação da Engenharia de Software para satisfazer os WIS, alguns tipos de frameworks e a descrição de um método baseado nos mesmos, o FrameWeb.

2.1 Engenharia Web

A Engenharia Web (*Web Engineering* - WebE), é a Engenharia de Software aplicada ao desenvolvimento Web (PRESSMAN, 2011). A WebE propõe abordagens disciplinadas e sistemáticas para o desenvolvimento, a implantação e a manutenção de sistemas e aplicações baseados na Web (MURUGESAN et al., 2001).

De acordo com Pandolfi e Melotti (2006), os WebApps, diferentemente dos tipos convencionais de software, são geralmente provedores de informações a serem consumidas por um grande número de pessoas, por isso existe uma ênfase na apresentação, comportamento, aparência, navegação e outras qualidades estéticas.

Como os WebApps foram ficando mais robustos e complexos, o desenvolvimento dos mesmos também cresceu em dificuldade. Problemas que eram vistos no desenvolvimento de aplicações *Desktop* passaram a aparecer na criação dos WebApps. Dificuldades como: inexperiência e formação inadequada dos desenvolvedores, falta (de uso) de métricas para estimativas, falta (de uso) de modelos de processo, métodos inadequados e obsoletos, planejamento incorreto, prazos e custos excedidos, falta de documentação, dificuldades de implementação e manutenção, WebApps mal definidas e projetadas, layout inadequado (comunicação visual), mudança contínua dos requisitos, da tecnologia e da arquitetura, falta de rigor no processo de desenvolvimento, dentre outros (PERUCH, 2007).

Existe então um conjunto de atributos de qualidade a serem considerados neste caso (OLSINA et al., 2001):

- **Usabilidade:** trata-se de um requisito de qualidade como também um objetivo de aplicações Web: permitir a acessibilidade do sistema. Logo, o site deve ter uma inteligibilidade global, permitir o feedback e ajuda online, planejamento da interface/aspectos estéticos e aspectos especiais (acessibilidade por deficientes);
- **Funcionalidade:** o software Web deve ter capacidade de busca e recuperação de informações/links/funções, aspectos navegacionais e relacionados ao domínio da aplicação;
- **Eficiência:** o tempo de resposta deve ser inferior a 10s (velocidade na geração de páginas/gráficos);
- **Confiabilidade:** relativa à correção no processamento de links, recuperação de erros, validação e recuperação de entradas do usuário;
- **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.

Como a WebE visa aplicar métodos disciplinados existentes na Engenharia de Software, adaptados para criação de aplicações Web, algum processo deve ser seguido a fim de garantir esses atributos.

Devido a complexidade de se projetar e implementar um software com vários requisitos e muitos *stakeholders*, como é o caso dos WebApps, é interessante pensar-se em metodologias que usam uma abordagem incremental e iterativa. Um processo de desenvolvimento segundo esse modelo divide a concepção de software em iterações. Em cada iteração são realizadas as atividades de análise, projeto, implementação e testes para uma parte do sistema.

O modelo de processo da WebE inicia-se pela análise de requisitos, onde é levantado tudo aquilo considerado necessário para a entrega da aplicação. A análise de requisitos contém atividades de formulação do problema a ser resolvido pela WebApp. A coleta de requisitos, onde são coletadas todas as informações possíveis sobre as funções que o sistema deve executar (requisitos funcionais) e as restrições sob as quais ele deve operar (requisitos não funcionais). E por fim a modelagem de análise.

Na criação do Modelo de Análise são focados quatro tipos de análise: a de conteúdo, que identifica como deve ser apresentado o conteúdo do site ou da aplicação, a de interação, que verifica como será a interação dos usuários com a WebApp, a funcional que identifica como o conteúdo é afetado ou criado pela WebApp e a de configuração, que descreve o ambiente e a infraestrutura nos quais a WebApp reside ([PRESSMAN, 2011](#)).

Guiado pelos resultados da modelagem de análise, o projeto da WebApp enfoca seis tópicos principais: projeto de interface, projeto de estética, projeto de conteúdo, projeto de navegação, projeto arquitetural e projeto de componentes. Existem alguns métodos que podem ser aplicados no desenvolvimento de aplicações Web, como WAE ([CONALLEN, 2002](#)), OOWS ([PASTOR; FONS; PELECHANO, 2003](#)) e OOHDM ([SCHWABE; ROSSI, 1998](#)).

Na fase de implementação é feita a tradução dos modelos criados para um software de fato, deve ser escolhida uma linguagem que se adeque aos requisitos do projeto.

Como exemplo dessa abordagem temos a metodologia proposta por [Conallen \(2002\)](#), que descreve um processo de desenvolvimento de software iterativo, incremental e centrado em casos de uso. As atividades do processo são mostradas na Figura 1.

O sub-processo “Reiterar” inicia com a revisão e refinamento do plano da iteração. A seguir, as seguintes atividades são executadas em paralelo: levantamento de requisitos, análise, projeto da arquitetura, projeto detalhado, implementação, testes, gerência de alterações, desenvolvimento do ambiente e desenvolvimento do projeto. Finalizada a iteração, o progresso alcançado até o momento é apresentado às partes interessadas

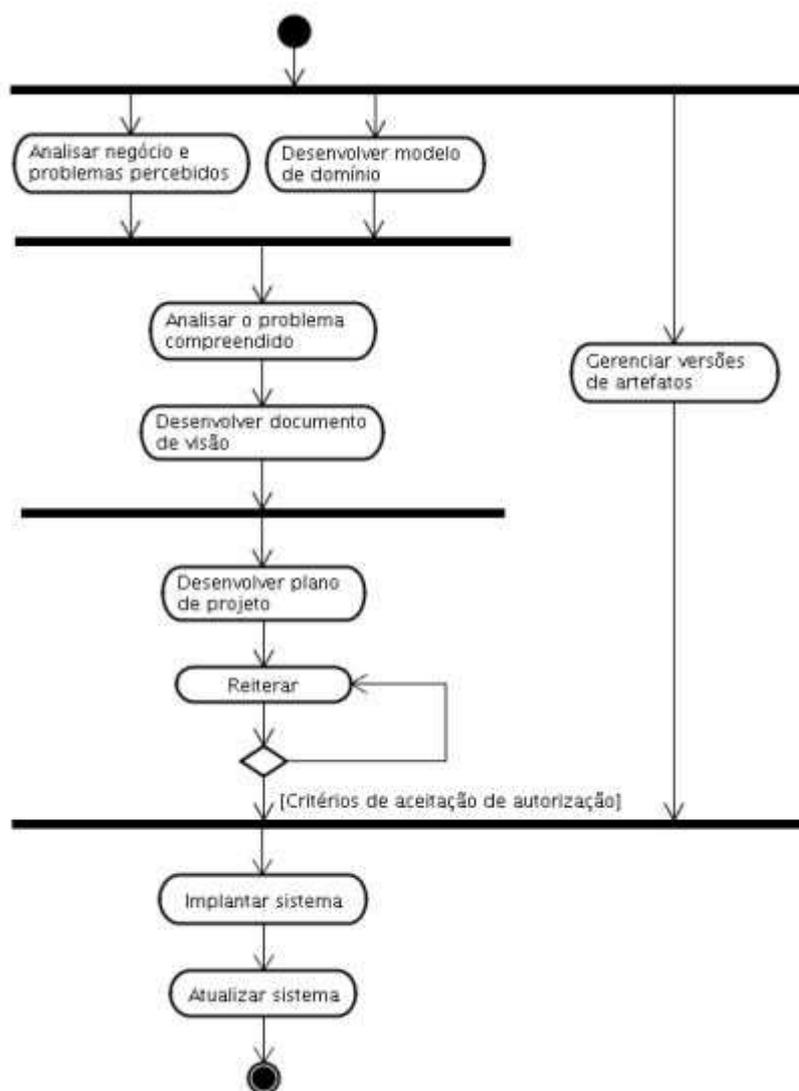


Figura 1 – Processo de desenvolvimento de software proposto por Conallen (2002)

(stakeholders) e a iteração atual é avaliada, ajustando o plano da próxima iteração, se necessário.

2.2 Frameworks

Os WIS, e a maioria dos WebApps, possuem uma infraestrutura muito similar. Por isso a partir do momento em que os primeiros projetos na área foram implementados, desenvolveu-se vários frameworks que generalizavam essa infraestrutura e poderiam ser utilizados para o desenvolvimento de novas aplicações.

Nesse contexto um framework pode ser visto como uma aplicação reutilizável e semicompleta que pode ser especializada para produzir aplicações personalizadas (HUSTED; DUMOULIN, 2004). Isso significa que o código do framework pode ser usado em conjunção

com o código específico do projeto a ser implementado. Assim esses frameworks permitem que sistemas Web de grande porte sejam construídos com arquiteturas de múltiplas camadas sem muito esforço de codificação.

Pode-se tomar como exemplo um requisito que a maioria das WIS possui, que é a autenticação dos usuários do sistema. Como em quase todos os casos a autenticação é feita a partir de um login, que consiste em um nome de usuário e senha, não há necessidade de uma interação específica entre o resto do sistema e o componente de autenticação.

[Souza \(2007\)](#), organiza a maioria dos frameworks de WebApps em 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 explicamos melhor os frameworks MVC, de Mapeamento Objeto/Relacional e de de Injeção de Dependência. que são os principais frameworks usados no projeto do SCAP.

2.2.1 Frameworks MVC

MVC é a abreviatura de Modelo-Visão-Controlador (*Model-View-Controller*), uma arquitetura de software desenvolvida pelo Centro de Pesquisas da Xerox de Palo Alto (Xerox PARC) para a linguagem Smalltalk em 1979 ([REENSKAUG, 1979](#)) ([SOUZA, 2007](#)).

Afim de manter a manutenibilidade, WebApps mais complexos precisaram realizar uma separação entre os dados (*Model*) e o layout (*View*). Assim, mudanças realizadas no layout das páginas não afetam a manipulação de dados, e estes poderão ser reorganizados sem alterar o layout. O MVC resolve este problema através da separação das tarefas de acesso aos dados e lógica de negócio, lógica de apresentação e de interação com o utilizador, introduzindo um componente entre os dois: o controlador (*Controller*).

A Figura 2 exemplifica a relação entre *Model*, *View*, *Controller* e Usuarios. Onde as linhas sólidas indicam associação direta e as tracejadas associação indireta.

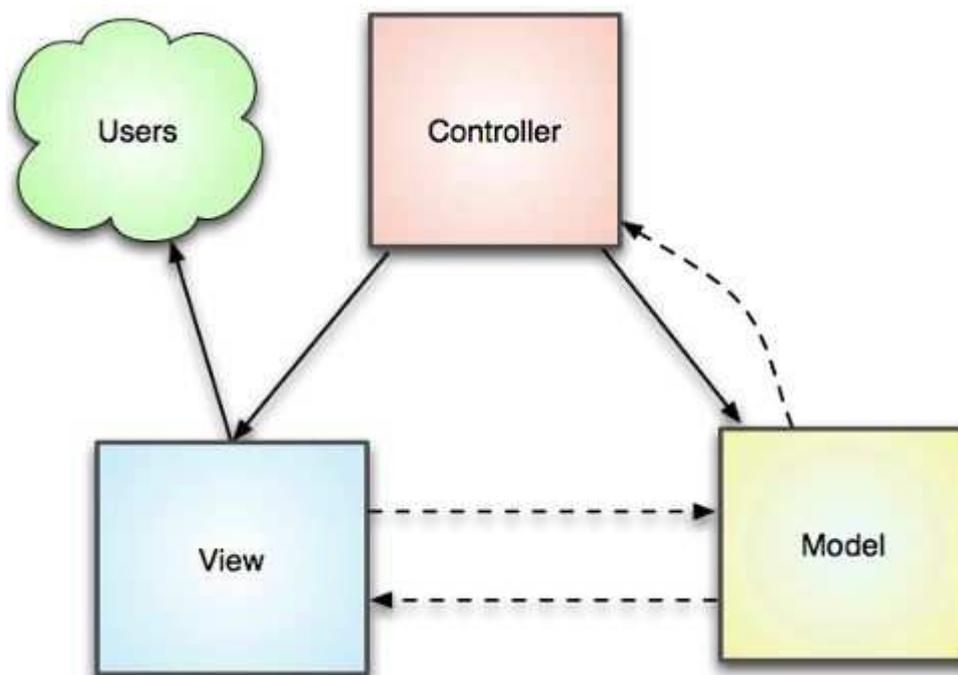


Figura 2 – Diagrama representando a arquitetura MVC.

Os elementos da *View* (Visão) fazem uma representação das informações do *Model* (Modelo) para os usuários, a partir das quais os mesmos podem interagir com a aplicação. Essa interação é tratada pelo *Controller* (Controle) que pode alterar o status dos elementos do *Model*. Este que pode notificar a *View* tais alterações, fazendo com que esta última possa atualizar a informação apresentada ao usuário.

Porém, na plataforma Web a arquitetura MVC precisa ser levemente adaptada, visto que o *Model*, situado no servidor Web, não pode notificar a *View* sobre alterações, já que esta encontra-se no navegador do lado do cliente e a comunicação é sempre iniciada pelo cliente. Portanto o nome correto para esse padrão arquitetônico, quando aplicado à Web, seria “Controlador Frontal” (*Front Controller*) (ALUR; CRUPI; MALKS, 2003, p. 166) (SOUZA, 2007).

A Figura 3 representa o funcionamento de um Controlador Frontal na Web.

Vemos que o navegador apresenta as páginas para o cliente que faz uma requisição ao servidor, este que a delega para o Controlador Frontal que, ao interagir com as classes da aplicação, executa a ação para, em seguida, delegar o resultado para a tecnologia de visão. Os frameworks MVC fornecem um controlador frontal a ser configurado pelo desenvolvedor para que se adapte ao seu projeto.

2.2.2 Frameworks de Mapeamento Objeto/Relacional

Quase todos os sistemas de informação necessitam de alguma forma de persistência de dados. O padrão da indústria atual é o uso do Sistemas de Bancos de Dados Relacionais

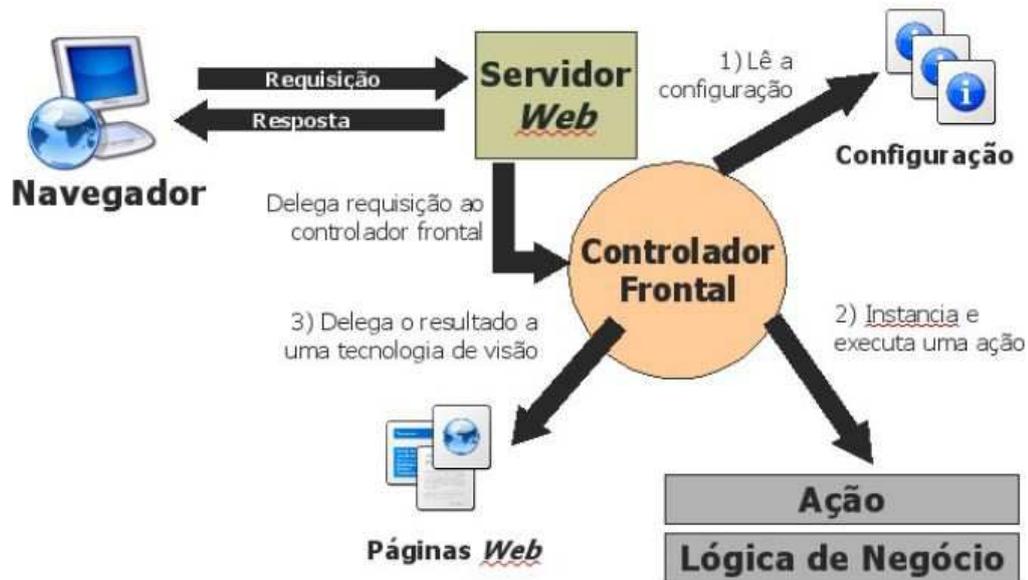


Figura 3 – Representação de um Controlador Frontal na Web (SOUZA, 2007).

(SGBDR), porém a grande maioria das aplicações são desenvolvidas no paradigma orientado a objetos (OO). Temos então um “incompatibilidade de paradigmas” (*paradigm mismatch*) (SOUZA, 2007).

Isso se dá porque “operações de SQL [linguagem estruturada de consultas para SGBDRs] como projeção sempre unem resultados em uma representação em tabelas de dados resultantes. Isso é bem diferente do grafo de objetos interconectados usados para executar uma lógica de negócio em um aplicativo Java!” (BAUER; KING, 2005).

Para solucionar esse problema surgiu na década de 80 o Mapeamento Objeto/Relacional (*Object/Relational Mapping – ORM*), que consiste na persistência automática (e transparente) de objetos de um aplicativo OO para tabelas de um banco de dados relacional, utilizando meta-dados que descrevem o mapeamento entre o mundo dos objetos e o mundo relacional (BAUER; KING, 2005).

Basicamente o desenvolvedor apenas informa ao framework de Mapeamento Objeto/Relacional como transformar os seus objetos em tabelas que serão usadas no banco de dados relacional. Isso livra o desenvolvedor do trabalho de realizar o mapeamento manualmente, o que toma muito tempo e, se realizado incorretamente, pode gerar inconsistências no banco de dados da aplicação.

Na implementação do SCAP foi usado o plugin do Hibernate juntamente com Java Persistence API (ou simplesmente JPA), que é uma API padrão da plataforma Java EE que descreve uma interface comum para frameworks de persistência de dados.

2.2.3 Frameworks de Injeção de Dependência

Na arquitetura em camadas busca-se garantir o reuso dos pacotes, e para que isso seja possível é necessário que os pacotes possuam um nível baixo de acoplamento, por exemplo: classes relacionadas ao domínio do problema não devem ser capazes de interagir somente com um tipo de persistência dos dados, e vice versa.

Assim as classes de camadas diferentes precisam interagir umas com as outras, sem precisar saber as dependências específicas das outras camadas. Assim temos o conceito de Injeção de Dependência (DI - *Dependency Injection*) (FOWLER, 2002), em que a responsabilidade de configurar dependências entre classes é assumida por um terceiro objeto. Desta forma as dependências entre os módulos não são definidas pelo programadores, elas são “injetadas” por um framework de injeção de dependências, mantendo um acoplamento baixo entre os módulos da aplicação. A Figura 4 representa o funcionamento de um framework de Injeção de Dependência.

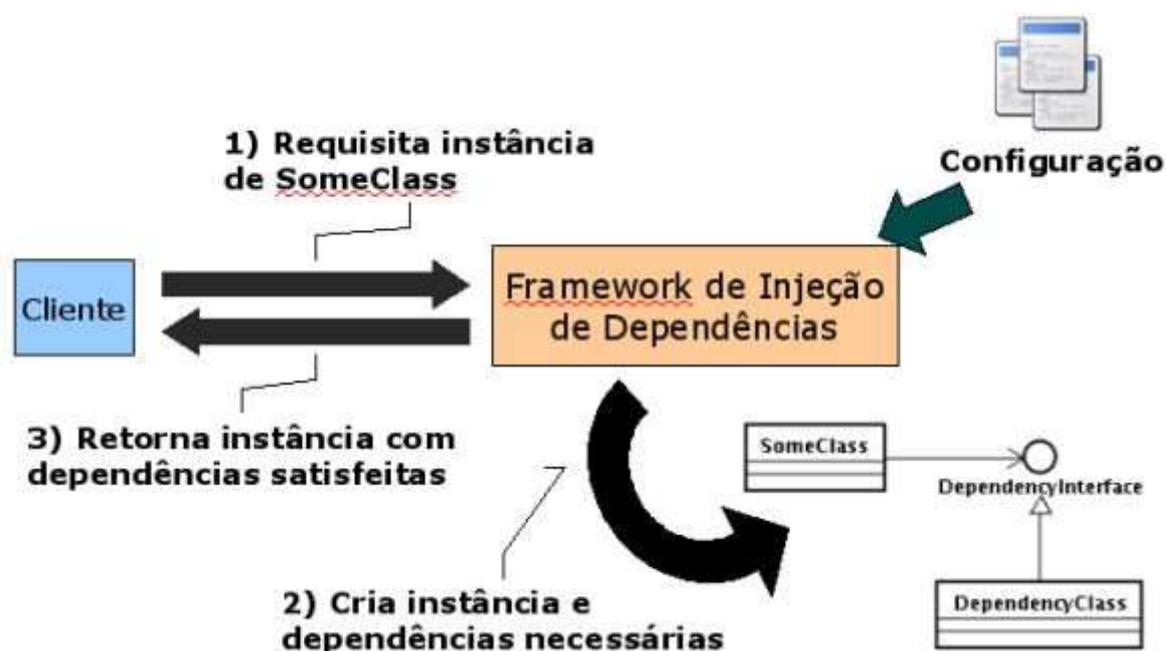


Figura 4 – Funcionamento de um framework de Injeção de Dependências (SOUZA, 2007).

2.3 O Método FrameWeb

FrameWeb (SOUZA, 2007) é um método, baseado em frameworks, para o desenvolvimento de sistemas de informação Web (*Web Information Systems – WISs*). Mesmo com o uso em larga escala dos frameworks no desenvolvimento dos aplicativos, não havia nada que abordasse diretamente aspectos característicos dos frameworks na Engenharia Web. O método FrameWeb foi criado para tratar exatamente dessa questão, assumindo

que algumas categorias de frameworks serão utilizados no processo de implementação de uma aplicação.

O FrameWeb concentra-se na fase de projeto. No entanto, espera-se que um processo de desenvolvimento completo seja conduzido de modo a produzir um produto final de qualidade. A Figura 5 ilustra o processo de desenvolvimento seguido durante a construção deste trabalho.

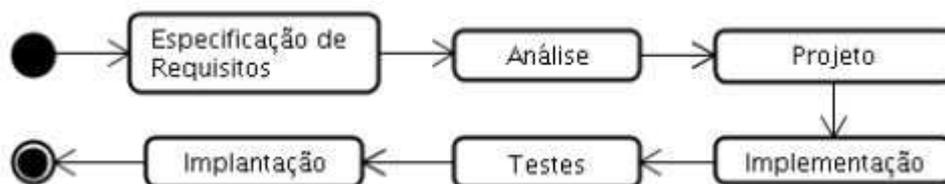


Figura 5 – Um processo de desenvolvimento de software simples sugerido por FrameWeb (SOUZA, 2007).

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 criação de um perfil UML que faz com que os diagramas fiquem mais próximos da implementação (SOUZA, 2007).

A fase de implementação é bastante agilizada devido ao uso dos frameworks e também pela fidelidade que existe entre os modelos da fase de projeto com o que será implementado.

2.3.1 Arquitetura de software do FrameWeb

Na Figura 6 vemos a arquitetura lógica padrão para WISs que é utilizada no FrameWeb. O sistema é dividido em três camadas: lógica de apresentação, lógica de negócio e lógica de acesso a dados. Essas que serão subdivididas nos pacotes: Visão, Controle, Aplicação, Domínio e Persistência.

A primeira camada, Lógica de Apresentação, lida com a interação entre o usuário e o sistema. Nela são apresentadas as informações das classes de domínio, é onde o cliente faz as suas requisições ao sistema. Esta camada contém os pacotes de Visão e Controle.

No pacote de Visão estão as páginas Web, imagens, scripts que executam ao lado do cliente e qualquer arquivo que esteja relacionado exclusivamente com a apresentação de informação ao usuário. O segundo pacote da camada é o de Controle, que contém as classes de controle juntamente com os arquivos do framework Controlador Frontal. Este possui uma relação de mútua dependência com o pacote de Visão, pois o pacote de Visão recebe

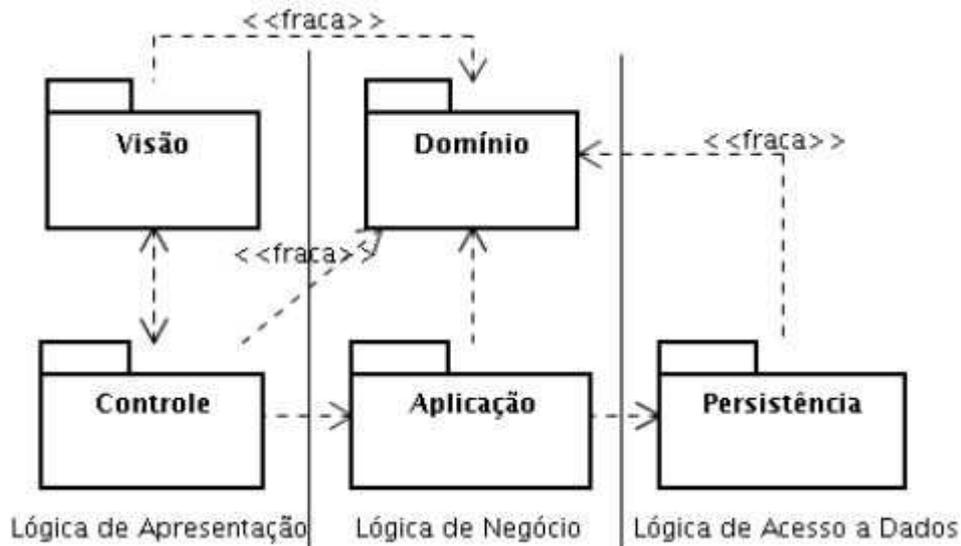


Figura 6 – Arquitetura padrão para WIS baseada no padrão arquitetônico Service Layer (FOWLER, 2002).

os estímulos que acionam as classes do pacote Controle, que ao receber tais estímulos apresenta as informações das classes do domínio a serem apresentadas ao cliente.

A Lógica de Negócio é a camada que contém os pacotes de Domínio e de Aplicação. No primeiro encontram-se as classes que representam os conceitos relacionadas ao domínio do problema, essas classes são implementadas seguindo o diagrama de classes de domínio que foi criado na fase de análise. Já no pacote de Aplicação ficam implementadas as classes que realizam os casos de usos levantados previamente, para isso ele precisa manipular objetos do domínio, assim existe uma relação de dependência entre o pacote de Aplicação e o pacote de Domínio.

Como o pacote de Aplicação executa de fato os casos de uso, ele possui uma dependência do pacote de Persistência da camada de Lógica de Acesso a Dados.

Na terceira camada, a Lógica de Acesso a Dados, fica o pacote de Persistência, que consiste em classes que gravam os atributos das classes que precisam ser persistidas pelo sistema. O frameWeb espera a utilização de algum framework para a realização do mapeamento Objeto/Relacional, juntamente com o uso do padrão de projeto Data Access Object (DAO) (ALUR; CRUPI; MALKS, 2003, p. 462). Padrão tal que separa a tecnologia de persistência de dados da Lógica de Negócio, criando uma camada de abstração, tornando a aplicação independente do framework ORM.

2.3.2 Linguagem de modelagem de FrameWeb

Seguindo a mesma abordagem de linguagens de modelagem, como WAE (CONNALLEN, 2002) e UWE (KOCH et al., 2000), FrameWeb define extensões leves para o meta-modelo da UML a fim de representar componentes típicos da Web e os componentes

relacionados aos frameworks, criando uma linguagem de modelagem, baseada em UML, com a qual são criados diagramas de quatro tipos diferentes, cujo propósito é guiar a implementação das camadas explicadas na seção anterior (SOUZA, 2007).

- **Modelo de Domínio:** é desenvolvido a partir do diagrama de classes criado na fase de análise, onde são especificados os tipos de dados usados e o mapeamento das classes de domínio que serão persistidas.
- **Modelo de Persistência:** guia a implementação das classes DAO do sistema, mostrando todas as interfaces e suas implementações juntamente com os métodos específicos de cada uma.
- **Modelo de Navegação:** é um diagrama de classes que representa as interações das páginas Web, formulários e classes de ação do Front Controller. Basicamente demonstra o funcionamento da camada de Lógica de Apresentação.
- **Modelo de Aplicação:** representa as classes de Serviço do sistema, é usado para guiar a implementação das mesmas e demonstrar suas dependências de classes de outros pacotes.

Todos esses modelos foram utilizados no projeto do SCAP. Assim, nos capítulos seguintes, exemplos práticos de como aplicá-los no desenvolvimento de um WIS facilitarão o entendimento do método. Foram usado frameworks da plataforma Java EE 7 juntamente com o controlador frontal VRaptor 4, diferente dos que foram usados na proposta inicial do FrameWeb.

3 Especificação de Requisitos e análise do SCAP

A Especificação de Requisitos é a primeira fase do processo de software, visto que nela são especificadas as necessidades que o software deve suprir. É realizada pelos desenvolvedores juntamente com os clientes, onde os mesmos explicam ao desenvolvedores o processo que deve ser realizado pela aplicação.

Os requisitos são divididos em 3 categorias: Requisitos Funcionais, Requisitos Não Funcionais e Regras de Negócio. Requisitos Funcionais são funcionalidades que o software deve ser capaz de realizar afim de atender às necessidades do cliente, por exemplo: cadastrar uma pessoa, listar documentos salvos pelo sistema, etc.

Requisitos não funcionais são relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenibilidade e tecnologias envolvidas. A maioria dos requisitos não funcionais não depende do cliente e devem existir em qualquer software desenvolvido.

Regras de Negócio são particulares para cada cliente, pois definem restrições e requerimentos do negócio que são realizados pelo próprio cliente. Assim elas podem variar entre diversos assuntos, como, por exemplo: obrigações contratuais, decisões estratégicas, leis e regulamentações entre outros.

neste capítulo é feita uma descrição de escopo do SCAP (Sistema de Controle de Afastamentos de Professores), depois tem-se os modelos de casos de uso que foram observados a partir dos requisitos previamente levantados (DUARTE, 2014), juntamente com alguns requisitos que foram observados posteriormente. A documentação completa dos requisitos do SCAP encontra-se nos apêndices deste trabalho.

3.1 Descrição do Escopo

No departamento de informática (DI) da UFES os pedidos de afastamento são solicitados para eventos no Brasil ou no exterior. Tais solicitações são avaliadas por professores do DI e, em alguns casos, também pela diretoria do Centro Tecnológico (CT) juntamente com a Pró-reitoria de Pesquisa e Pós-Graduação (PRPPG). Caso aprovado em todas as instâncias, o afastamento é aceito.

Para eventos no Brasil o pedido de afastamento precisa ser aprovado pela Câmara Departamental (formada pelos funcionários do departamento e representantes discentes). Assim, o pedido é feito por meio da lista de e-mails dos funcionários do DI, endereçado

ao chefe do departamento, cargo este que é exercido por algum professor do DI durante mandato temporário. Caso nenhum membro da Câmara Departamental se manifeste contra o afastamento em dez dias, o pedido é aprovado. Assim para eventos nacionais o processo ocorre inteiramente dentro do departamento.

Para eventos fora do Brasil é escolhido um professor (que não tenha parentesco com o solicitante) para ser o relator do pedido. Após parecer do relator, o pedido passa por aprovação no departamento como no caso acima. No entanto, é necessário ainda que o CT e a PRPPG aprovem o pedido e que o afastamento seja publicado no Diário Oficial da União. Porém o SCAP só lida com a tramitação dentro do DI, não existe uma integração com os processos do CT e da PRPPG. Desta forma, o escopo do sistema só trata do pedido na parte que cabe ao DI.

O SCAP existe para assistir os professores e secretários do departamento na tramitação das solicitações de afastamento, tornando mais simples todo o processo, tanto a criação do pedido quanto a análise e armazenamento do mesmo. O sistema atinge esse objetivo através do envio de e-mails automáticos aos envolvidos e da utilização de formulários na criação dos documentos necessários.

3.2 Modelo de Casos de Uso

Após a definição do escopo, o processo de Engenharia de Requisitos seguiu com a modelagem de casos de uso, conforme proposto por (FALBO, 2011). A Tabela 1 lista e descreve os atores identificados no levantamento de requisitos.

Tabela 1 – Atores do SCAP

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 secretários lidam com a parte administrativa do sistema, é responsabilidade deles cadastrar os professores, bem como seus parentescos e mandatos dos chefes do departamento. Outras tarefas dos secretários envolvem a realização do processo em que os pedidos de afastamento são submetidos: eles cadastram os pareceres de fora do DI e também arquivam os pedidos já finalizados.

Os professores cadastram seus pedidos de afastamento no sistema. Eles também podem se manifestar contra o afastamento de outro professor, caso ainda seja tempo hábil para tal. Se for adicionado como relator de um afastamento no exterior, é responsabilidade do professor decidir se o DI aprova ou não tal afastamento.

O chefe do departamento é um professor que foi selecionado para exercer a função administrativa de chefe do DI por um mandato. Assim é de responsabilidade dele encaminhar solicitações a relatores que deferirão pareceres sobre afastamentos no exterior.

O SCAP foi dividido em 2 subsistemas: Núcleo e Secretaria. O primeiro contempla os casos de usos dos professores e do chefe de departamento, enquanto o segundo envolve os casos de uso dos secretários.

As Figuras 7 e 8 mostram os diagramas de casos de uso destes subsistemas. Nos parágrafos que se seguem, apresentamos uma descrição sucinta dos casos de uso levantados para o SCAP. Uma versão mais detalhada dessa descrição pode ser vista nos apêndices deste trabalho.

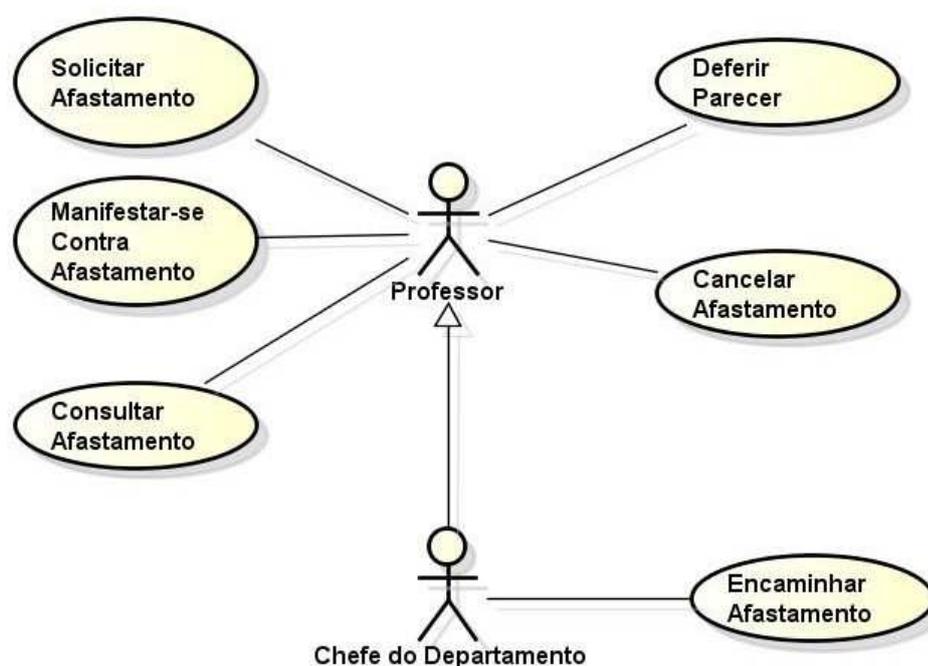


Figura 7 – Diagrama de Casos de Uso do subsistema Núcleo.

No caso de uso **Solicitar Afastamento** um professor cadastra um pedido de afastamento no sistema, informando todos os dados necessários para a tramitação do mesmo. Em **Cancelar Afastamento** o professor cancela um pedido de afastamento e o seu status é alterado para cancelado, caso ele seja o solicitante.

O caso de uso **Encaminhar Afastamento** é realizado pelo Chefe do Departamento quando o mesmo analisa um pedido de afastamento internacional recém cadastrado e aponta um professor como relator deste afastamento. Já **Deferir Parecer** é utilizado quando um professor, que foi cadastrado pelo Chefe do Departamento como relator de um afastamento internacional, cadastra seu parecer sobre o afastamento.

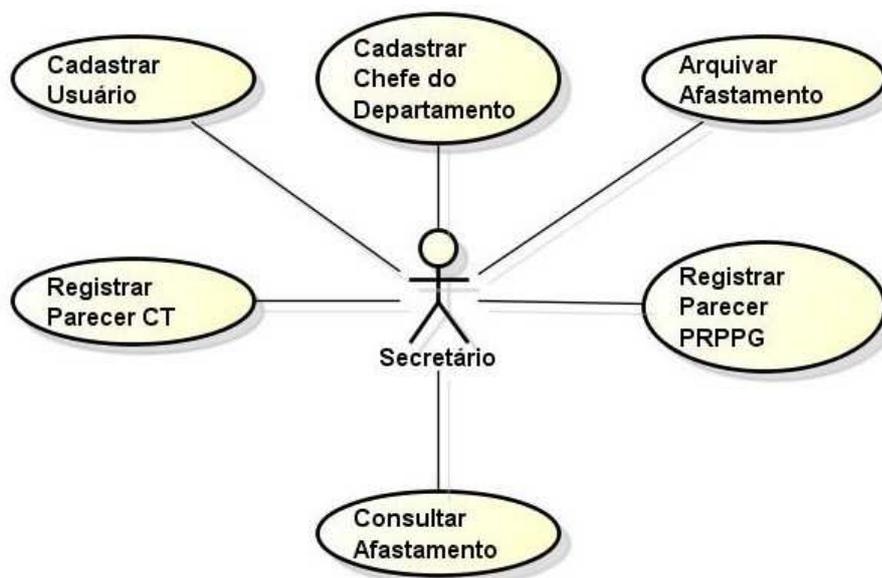


Figura 8 – Diagrama de Casos de Uso do subsistema Secretaria.

Consultar Afastamento, como o nome já diz, consiste em uma busca que um professor realiza para ver os dados de um pedido de afastamento. Em **Manifestar-se Contra Afastamento** um professor que é contrário a um afastamento cadastra o motivo de sua opinião contrária, assim uma reunião é marcada e os professores decidem se o afastamento é aprovado ou não.

Em **Cadastrar Usuário** um secretário cadastra um novo professor ou secretário no sistema, informando os dados pessoais necessários. Em **Cadastrar Chefe do Departamento** um secretário cadastra o mandato do Chefe de Departamento, informando as datas de início e fim do mesmo.

O casos de uso **Registrar Parecer CT** e **Registrar Parecer PRPPG** acontecem quando um secretário cadastra o parecer do Centro Tecnológico e da Pró-Reitoria de Pesquisa e Pós-Graduação, respectivamente, sobre um pedido de afastamento internacional.

Arquivar Afastamento acontece no final da tramitação de um pedido de afastamento quando o secretário muda o status do mesmo para “Arquivado”.

3.3 Análise do SCAP

Durante a fase de análise adaptamos as entidades do mundo real, para que sejam utilizadas no paradigma de Orientação a Objeto do projeto. Assim tais entidades devem ser descritas em classes, suas características serão seus atributos e suas interações as relações entre classes.

Neste capítulo é apresentada a Análise que foi feita para o SCAP (Sistema de Controle Afastamento de Professores), em seguida tem-se as classes que foram desenvolvidas a partir dessa análise.

3.3.1 Modelagem de Classes

Durante a abstração de conceitos do mundo real para classes de objetos, atributos e relações, são criados diagramas de classes, que mostram as propriedades e as associações entre as classes do sistema (PERUCH, 2007).

Na elaboração do diagrama de classes é importante o foco na multiplicidade das relações, porque a mesma é fundamental na consistência das classes do projeto com a tecnologia de persistência dos dados.

Apesar do SCAP ter sido dividido em 2 subsistemas, as classes de domínio do problema, representadas no diagrama de classes da Figura 9, todas pertencem ao subsistema Núcleo. Isso se dá porque a divisão em subsistemas ocorreu com intuito de facilitar a implementação, e não devido a natureza das classes criadas.

O Atributo “situação da solicitação” da classe Afastamento representa em qual estágio da tramitação um pedido de afastamento está, os status possíveis serão discutidos mais adiante. Já o atributo “tipo do afastamento” indica se o afastamento é necessário para atender a um evento localizado no Brasil ou no exterior.

As classes **Professor** e **Secretário** representam os professores de secretários do Departamento de Informática da UFES respectivamente, ambas herdam os atributos da classe **Pessoa**. Professores podem possuir ou não uma relação de **Parentesco** com os demais professores. O chefe do departamento é representado por um **Professor** que possui um **Mandato** vigente.

Um **Afastamento** possui todas as informações necessárias para tramitação de um pedido de afastamento de um professor. A classe **Afastamento** pode ou não possuir mais de um **Documento** e requer um **Relator** somente quando o afastamento é devido a um evento internacional.

Um **Parecer** é emitido por um professor e é referente a um único afastamento, porém um professor pode criar vários pareceres para afastamentos diferentes, assim como um professor pode ser relator de vários afastamentos.

3.3.2 Restrições de Integridade

Restrições de Integridade complementam as informações de um modelo deste tipo 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. Tais regras devem ser documentadas junto ao modelo conceitual estrutural do sistema (FALBO, 2012).

Listaremos aqui as restrições que foram observadas na primeira implementação do SCAP em (DUARTE, 2014), juntamente com novas restrições referentes a funcionalidades que não foram contempladas no projeto passado do SCAP.

- 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 a data de fim do mesmo afastamento;
- A data de início de um mandato de professor não pode ser posterior a 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.

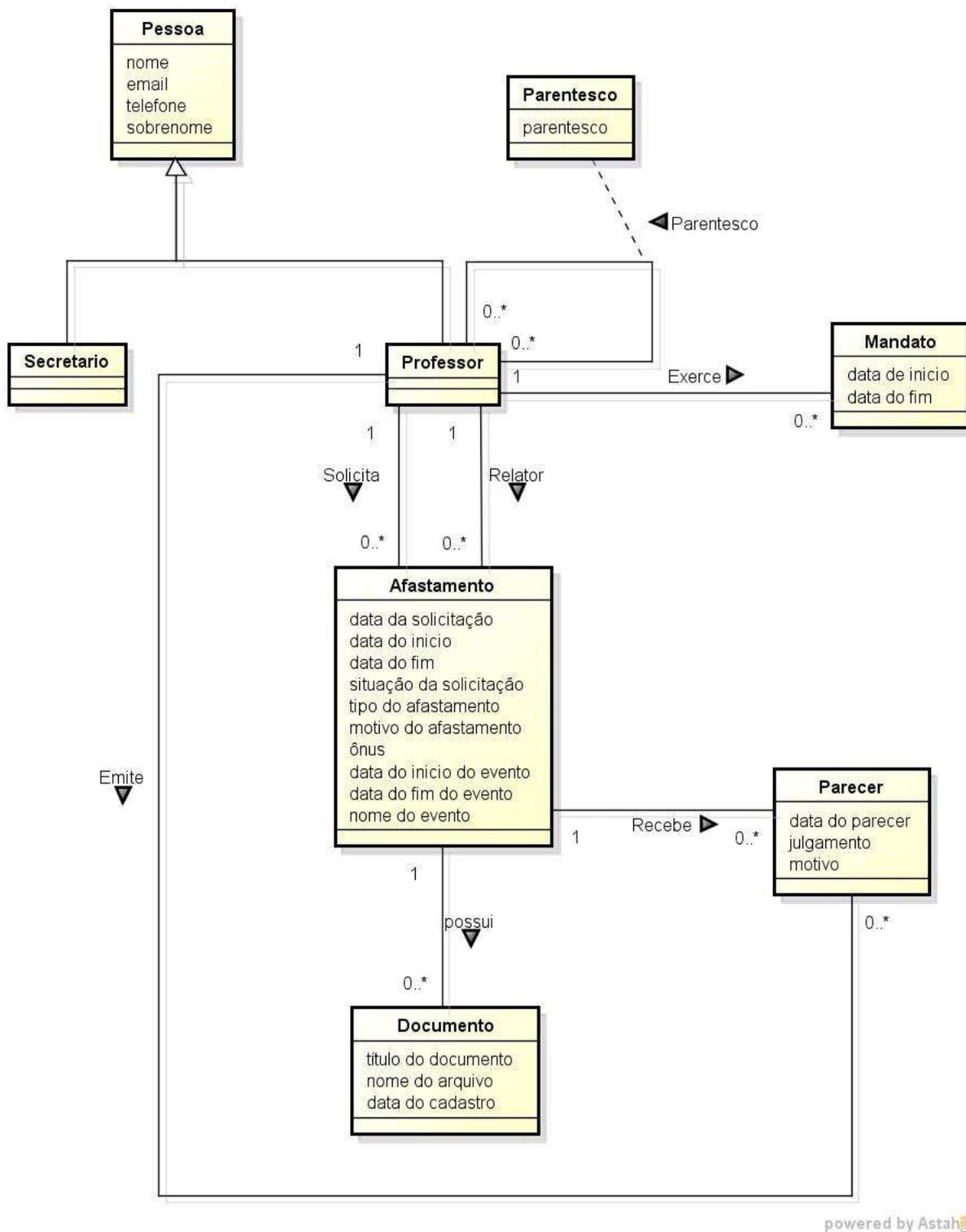


Figura 9 – Diagrama de Classes do SCAP.

4 Projeto e Implementação

Na fase de projeto deve-se produzir uma solução para o problema que foi identificado e modelado nas fases de levantamento e análise de requisitos, incorporando a tecnologia aos requisitos e projetando o que será construído na implementação. Sendo assim, é necessário conhecer a tecnologia disponível e os ambientes de hardware e software onde o sistema será desenvolvido e implantado. Durante o projeto, deve-se decidir como o problema será resolvido, começando em um alto nível de abstração, próximo da análise, e progredindo sucessivamente para níveis mais detalhados até se chegar a um nível de abstração próximo da implementação (FALBO, 2011).

Enquanto na fase de análise a tecnologia é imaginada como perfeita (capacidade ilimitada de armazenamento, custo zero e não passível de falha), a fase de projeto envolve a modelagem de como o sistema será implementado com a adição dos requisitos tecnológicos e de caráter não funcional (PRESSMAN, 2011).

Nas próximas seções é descrito como o projeto do SCAP foi realizado, na seção 4.1 é apresentada a arquitetura do sistema, com as tecnologias utilizadas e como estão dispostas as classes dentro dos pacotes do projeto. Na seção 4.2 vê-se a descrição dos frameworks previamente citados que foram aplicados no SCAP. A partir da seção 4.3 são descritos os modelos do projeto SCAP propostos no FrameWeb (Domínio, Navegação, Aplicação e Persistência). E na seção 4.7 é demonstrada o resultado da implementação do SCAP.

4.1 Arquitetura do Sistema

A definição da arquitetura do sistema inicia a fase de projeto de sistema, nela são mapeados os conceitos observados na fase de análise para a plataforma tecnológica de desenvolvimento. O nível de abstração da fase anterior é diminuído, pois busca-se alcançar maior proximidade com o nível da linguagem de programação (PERUCH, 2007).

4.1.1 Tecnologia Utilizadas

O SCAP foi implementado utilizando a linguagem de programação Java, na plataforma Java EE 7 (Java Enterprise Edition 7). Tal plataforma foi escolhida por já oferecer os frameworks CDI e de mapeamento objeto relacional. Para persistência foi utilizado o banco de dados relacional por ser a tecnologia mais aplicada no mercado.

A escolha do VRaptor como controlador frontal foi feita por ser um framework desenvolvido no Brasil e com uma larga documentação, juntamente com vários plugins

interessantes para o projeto. Dois plugins do VRaptor que valem ser mencionados são: VRaptor Tasks, que facilita o agendamento de tarefas automáticas que o SCAP deve realizar, e o VRaptor Simple Mail que ajudou no envio de e-mails automáticos.

Uma Recomendação dos criadores do VRaptor é o Apache Maven¹, que é uma ferramenta de gerência de projeto capaz de simplificar o download e instalação de bibliotecas e frameworks de um projeto. O servidor de aplicação usado foi o Wildfly 8² e o ambiente de desenvolvimento foi o Eclipse IDE³. Uma tabela com todas tecnologias utilizadas e suas descrições detalhadas pode ser lida nos apêndices deste trabalho.

4.2 Frameworks Utilizados

Nesta seção é feita uma descrição dos principais frameworks utilizados na implementação do projeto SCAP.

4.2.1 VRaptor 4

VRaptor⁴ é um framework MVC Web para desenvolvimento ágil com Java. Ele foi utilizado como framework controlador no projeto do SCAP, a descrição de controlador frontal foi feita na subseção 2.2.1 deste trabalho. Criado em 2003 no IME-USP, teve sua versão 2.0 lançada em 2005 e a versão 3.0 em 2009. Atualmente está em sua versão 4 e é mantido pela Caelum e diversos desenvolvedores de outras empresas.

O VRaptor buscar simplificar o controle de acesso do usuário através de convenções na nomenclatura dos arquivos e das classes, juntamente com uma lógica específica de diretórios que deve ser seguida na criação das páginas Web. Essa lógica será melhor explicada na seção 4.7 quando é apresentado o modelo de navegação do SCAP.

4.2.2 CDI

Contexts and Dependency Injection (CDI) para a plataforma Java EE é o framework de Injeção de Dependências (ou inversão de controle), utilizado na implementação do projeto do SCAP. É a principal dependência do VRaptor de acordo com o documentação⁵ do mesmo. Mais sobre o CDI pode ser lido na JSR-299⁶.

O CDI faz parte da especificação Java EE desde a versão 6 e substituiu o Spring que é o framework de injeção de dependências proposto no FrameWeb.

¹ Maven, <http://maven.apache.org/>

² Wildfly 8, <http://wildfly.org/>

³ Eclipse, <https://eclipse.org/downloads/>

⁴ Vraptor, <http://www.vraptor.org>

⁵ Dependências do VRaptor 4, <http://www.vraptor.org/pt/docs/dependencias-e-pre-requisitos/>

⁶ JSR-299, <https://docs.jboss.org/cdi/spec/1.0/>

4.2.3 Hibernate + JPA

Hibernate é um framework de Mapeamento Objeto/Relacional escrito na linguagem Java, que visa facilitar a transição dos objetos usados no sistema para as tabelas relacionais utilizadas no banco de dados.

Diferentemente da primeira implementação do SCAP (DUARTE, 2014), que usou o hibernate com a linguagem SQL, nesse projeto foi utilizada o JPA (Java Persistence API) com a linguagem JPQL. Assim a Lógica de Acesso a Dados ficou mais simplificada, pois em JPQL as consultas ao banco são escritas de forma mais próxima do padrão orientado ao objeto.

Para gerenciar o banco de dados usou-se o phpMyAdmin, que é um gerenciador de banco de dados gratuito implementado na linguagem PHP.

4.2.4 Pacotes do SCAP

A Figura 10 demonstra como foram divididas as classes do SCAP: primeiramente foi feita a divisão em 2 subsistemas que foram observados na fase de análise e especificação de requisitos. Depois a divisão segue o que foi proposto no FrameWeb, assim as classes foram subdivididas em 4 pacotes: Controle, Aplicação, Domínio e Persistência.

A Figura é retirada do *Package Explorer* da IDE Eclipse, que foi o ambiente de desenvolvimento utilizado no projeto. Nela vê-se toda a estrutura das pastas do sistema, juntamente com as bibliotecas e arquivos de configuração.

O pacote **br.ufes.scap.nucleo.aplicacao** contém 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.

Em sequência tem-se o pacote **br.ufes.scap.nucleo.controle** contendo as classes de Controle, responsáveis pela comunicação entre o pacote de visão e o de aplicação, tratando os estímulos enviados e recebidos pelo usuário.

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.

Finalmente, temos o pacote **br.ufes.scap.nucleo.presistencia**, no qual ficam as classes responsáveis pela lógica de acesso a dados do subsistema Núcleo. Elas são responsáveis em fazer a persistência das classes de Domínio.

Os pacotes seguintes possuem o mesmos objetivos dos supracitados, porém são referentes ao subsistema Secretaria. Vale notar que os arquivos do pacote de visão encontram-se no diretório **src\main\webapp** e são responsáveis pela interação direta com o usuário.

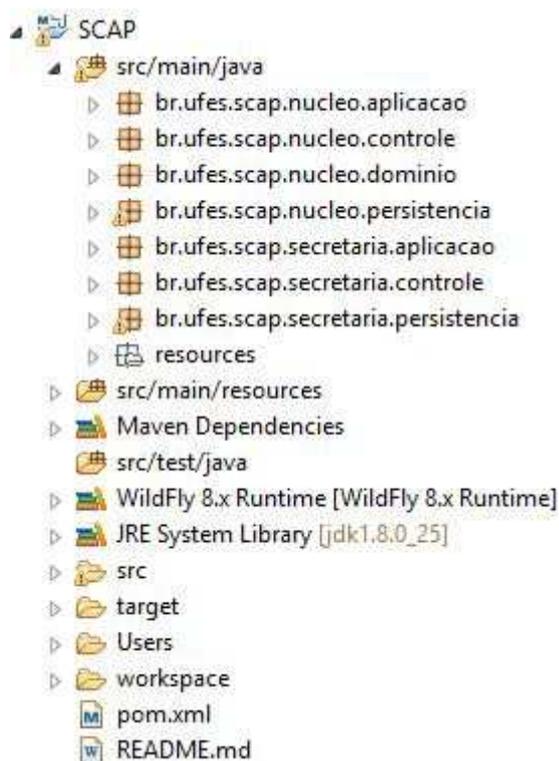


Figura 10 – projeto SCAP visto no Package Explorer do Eclipse IDE.

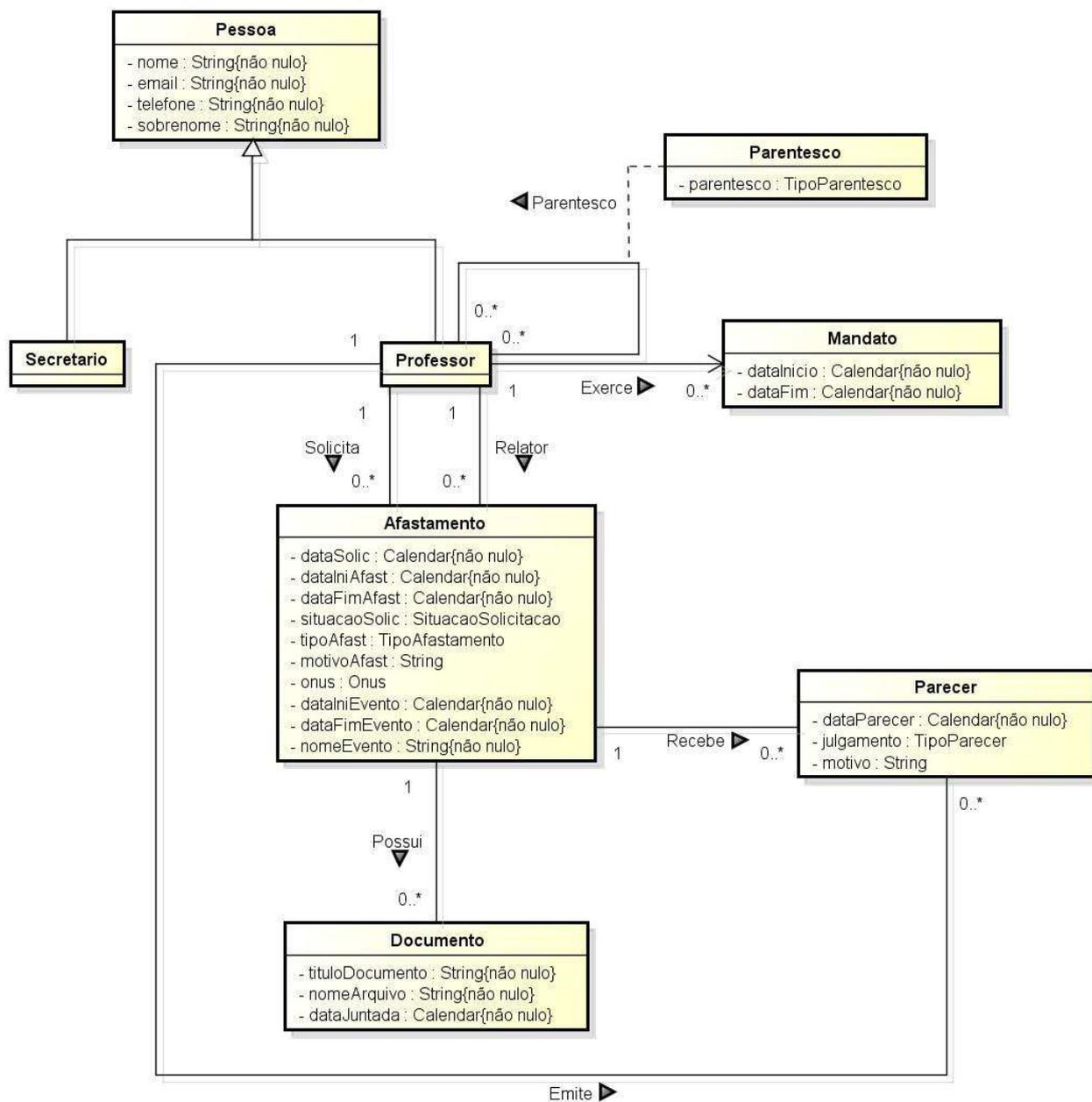
4.3 Modelo de Domínio

O modelo de domínio de FrameWeb é um diagrama de classes da UML que representa os objetos de domínio do problema e seu mapeamento para a persistência em banco de dados relacional. A partir dele são implementadas as classes da camada de domínio na atividade de implementação (SOUZA, 2007).

Ele foi construído a partir do diagrama de classes da Figura 9, que apresenta as classes de domínio que foram utilizadas na implementação. No modelo de domínio é feita uma adequação à plataforma de desenvolvimento escolhida, são indicados os tipos dos atributos, são definidas a navegabilidade das associações etc. A Figura 11 mostra o modelo de domínio do SCAP.

Pode-se observar que já são levadas em consideração a persistência das classes, pois já são salientados quais atributos podem ou não ser iniciados com valor nulo. São apresentados também os tipos de dados enumerados usados no projeto, que são: **TipoParentesco**, **SituacaoSolic**, **TipoAfast**, **Onus**, **TipoParecer**. A Figura 12 apresenta os valores desses tipos.

No tipo **Onus** apenas 3 valores são possíveis: TOTAL indica que a universidade arcará com os custos da viagem relacionada ao afastamento; PARCIAL significa que a universidade apenas manterá a remuneração normal do professor, porém não custeará a viagem; INEXISTENTE representa o caso em que mesmo a remuneração do professor é



powered by Astah

Figura 11 – Modelo de Domínio do SCAP.

suspensa durante o afastamento. **TipoAfast** apresenta valor NACIONAL para eventos realizados no brasil e INTERNACIONAL para eventos no exterior.

Em **TipoParecer** os valores FAVORAVEL E DESFAVORAVEL indicam que o parecer é a favor do afastamento ou contra respectivamente. Já em **TipoParentesco** SANGUINEO significa que o parentesco entre dois professores é sanguíneo e MATRIMONIAL quer dizer parentesco matrimonial.

O tipo **SituacaoSolicitacao** indica em qual estágio do processo o afastamento se encontra. No inicio o valor do campo é INICIADO, caso seja um evento marcado



Figura 12 – Tipos enumerados do SCAP.

como NACIONAL ele permanecerá com esse valor por 10 dias, caso nenhum professor se manifeste contra ele neste período ele é aprovado automaticamente e seu valor fica APROVADO-DI onde é terminado o processo. Caso seja um afastamento internacional o chefe de departamento cadastra um Relator para o afastamento mudando sua situação de INICIADO para LIBERADO. se o parecer do relator for FAVORAVEL o afastamento passa para APROVADO-DI, caso o parecer do CT seja também favorável ele passa para APROVADO-CT onde aguarda o parecer da PRPPG, caso este seja aprovado ele muda para APROVADO-PRPPG onde o processo é terminado. Se esses algum pareceres supracitados sejam negativos a situação passa para REPROVADO e o processo acaba. Em qualquer momento o professor pode cancelar seu pedido, assim o status é alterado para CANCELADO e o processo é terminado.

Os estados da classe Afastamento apenas descritos foram modelados em um diagrama de estados, como mostra a Figura 13. Nas transições entre os estados são indicados os casos de uso que promovem a transição de um afastamento de um estado a outro.

4.4 Modelo de Navegação

O Modelo de Navegação do FrameWeb é um diagrama de classe da UML que representa os diferentes componentes que formam a camada de Lógica de Apresentação, como páginas Web, formulários HTML e classes de ação do framework Front Controller (SOUZA, 2007).

Na Tabela 2 são descritos os estereótipos utilizados no projeto do SCAP pelos diferentes elementos que podem ser representados no Modelo de Navegação. O objetivo do

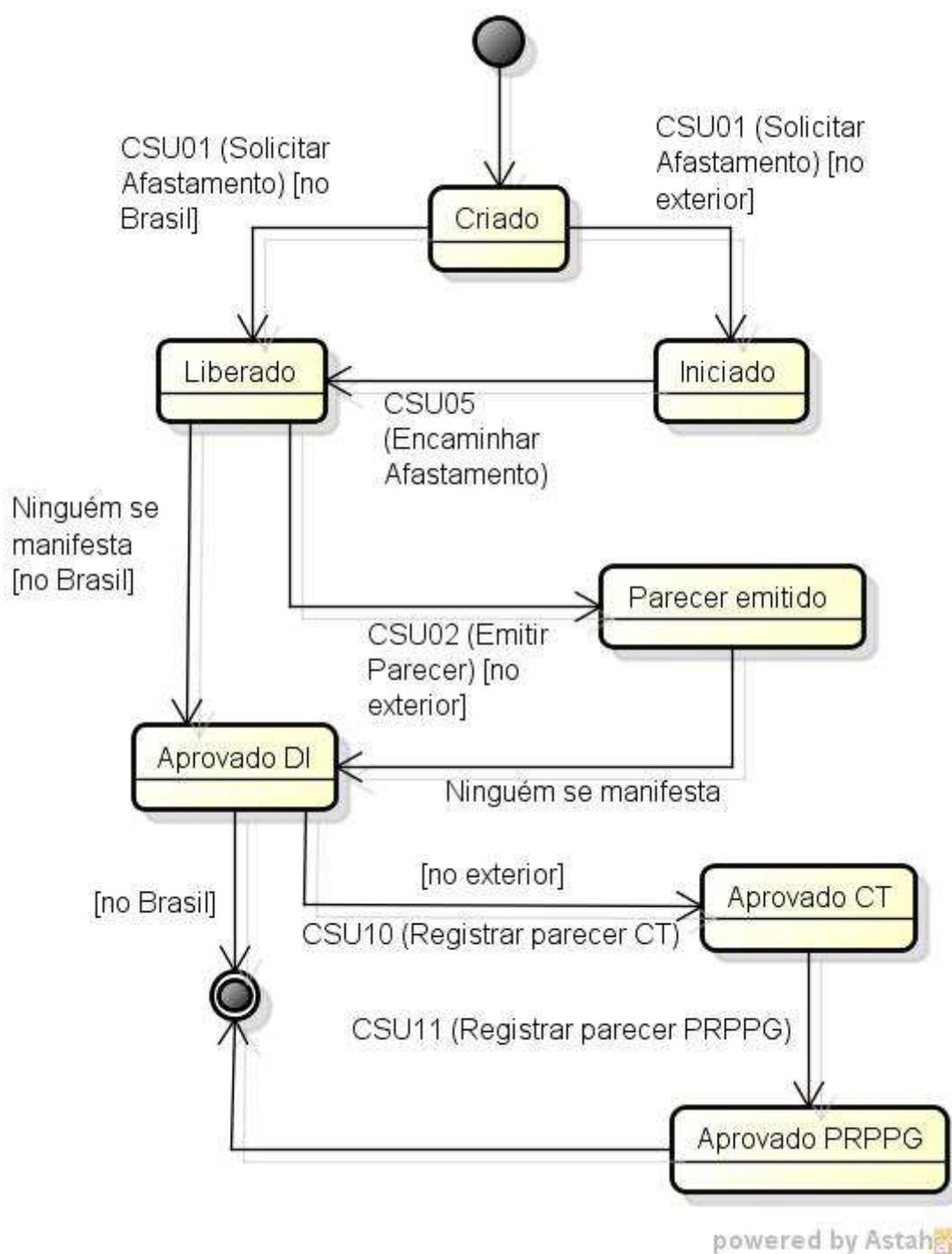


Figura 13 – Tipos enumerados do SCAP.

Modelo de Navegação é guiar a concepção das classes e componentes dos pacotes Visão e Controle.

Tabela 2 – Esteriótipos utilizados nos modelos de navegação do SCAP.

Esterótipo	O que Representa
Nenhum	Uma classe de ação, para a qual o framework Front Controller delega a execução da ação.
«page»	Uma página Web estática ou dinâmica.
«form»	Um formulário HTML.

Conforme proposto no FrameWeb, foram criados diagramas de navegação para todos os casos de uso levantados na fase de requisitos e podem ser vistos nos apêndices desse trabalho. Para ilustração, nessa seção são descritos somente os casos de uso Cadastrar Afastamento, na Figura 14, e Consultar Afastamento, na Figura 15.

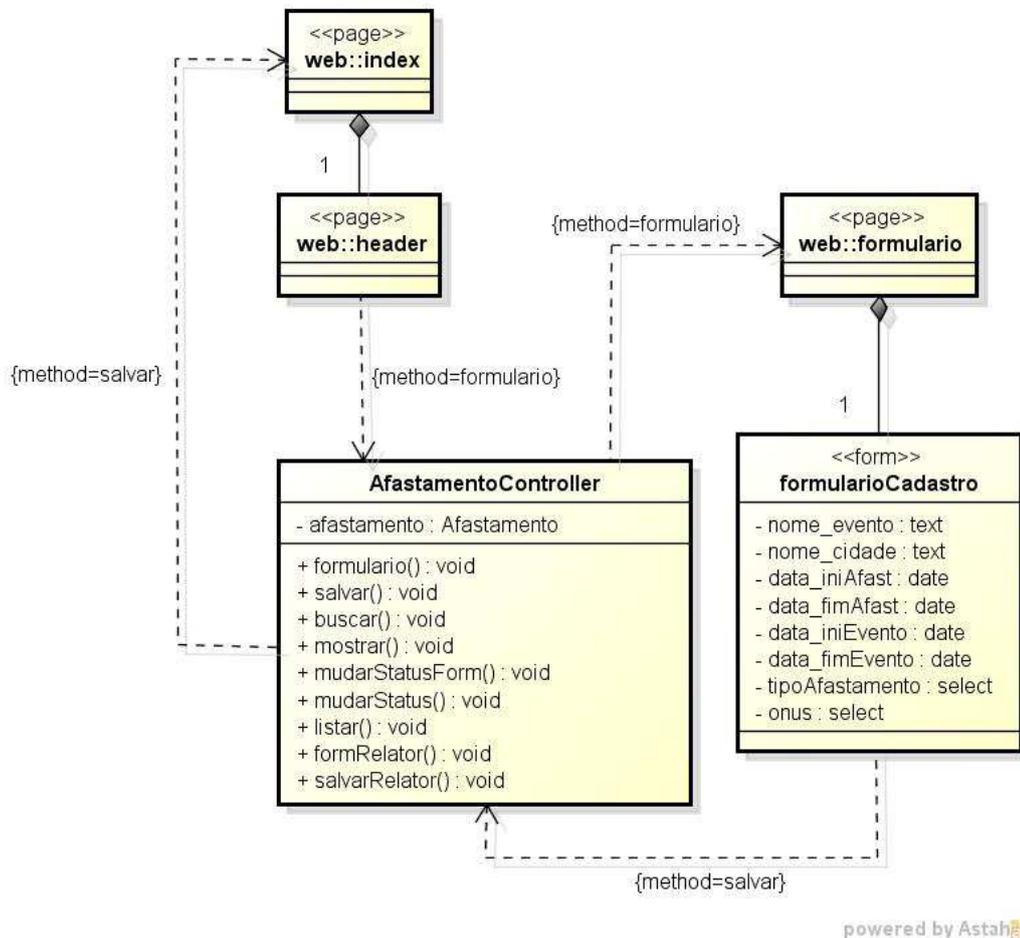


Figura 14 – Modelo de Navegação para o caso de uso Cadastrar Afastamento.

A criação do modelo de navegação leva em consideração funcionamento do framework Controlador Frontal, logicamente o modelo da Figura 14 foi feito segundo os preceitos propostos pela documentação do VRaptor.

É importante salientar que o método “salvar” não resulta no diretamente redirecionamento para a página index, na verdade a controladora em questão aciona o método

“index” da classe **IndexController** que por sua vez chama a página inicial. Isso foi omitido para que o modelo não ficasse muito complexo.

A classe **AfastamentoController** é a classe controladora que lida com a lógica de apresentação referente à classe de domínio **Afastamento**. Quando o método **formulario** da controladora é chamado as convenções de nome e diretório do VRaptor fazem com que a pagina contendo o formulário de cadastro de afastamento seja chamada. Posteriormente, quando o formulário é submetido, ele chama o método **salvar** da controladora que passa as informações do afastamento para a classes de aplicação.

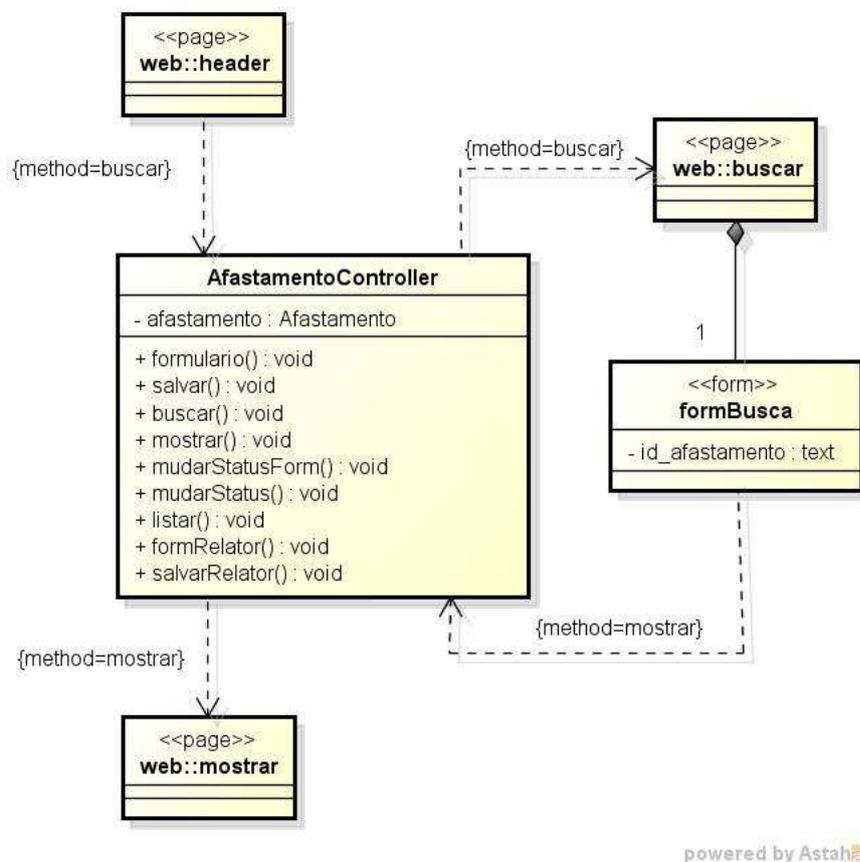


Figura 15 – Modelo de Navegação para o caso de uso Consultar Afastamento.

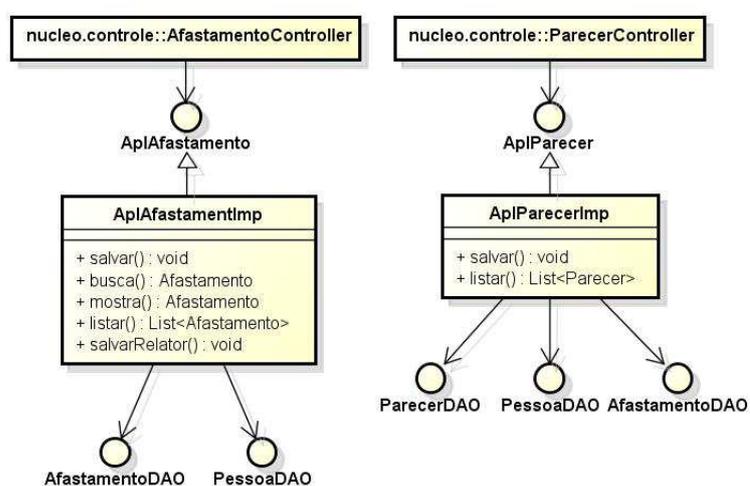
Na Figura 15 temos o modelo de navegação do caso de uso Consultar Afastamento. Mantem-se o mesmo funcionamento do diagrama anterior: quando o método **buscar** é chamado o formulário **formBusca** é apresentado ao usuário. Quando o formulário é preenchido, o método **mostrar** é chamado juntamente com a página **mostrar**, apresentando os dados recebidos da classe de aplicação **aplAfastamento** referentes ao id de afastamento recebido no formulário.

4.5 Modelo de Aplicação

O Modelo de Aplicação é um diagrama de classes da UML que representa as classes de serviço, que são responsáveis pela codificação dos casos de uso, e suas dependências (SOUZA, 2007). Ele é utilizado para demonstrar as dependências entre classes do pacote de Controle, Aplicação e Persistência. Podemos ver quais classes de ação (pacote controle da camada Lógica de Apresentação) dependem de quais classes de serviço (pacote aplicação da camada Lógica de Negócio) e quais classes DAO são necessárias (pacote de persistência da camada Lógica de acesso a Dados). A administração dessas dependências é simplificada pelo uso do framework CDI citado na subseção 4.2.2.

Assim como para as classes de controle do Modelo de Navegação, o projetista deve escolher o nível de granularidade das classes de serviço. Há também uma semelhança com o modelo de Persistência, pois no Modelo de Aplicação também não há definição de extensões UML e também valem as regras de programação por interfaces: cada classe de serviço deve ter uma interface e uma implementação (SOUZA, 2007).

As Figuras 16 e 17 são os Modelos de Aplicação criados para o subsistemas Núcleo e Secretaria respectivamente. No SCAP optou-se por manter uma classe de serviço para cada classe de controladora do pacote de controle, assim fica mais simples entender a relação entre os dois pacotes.

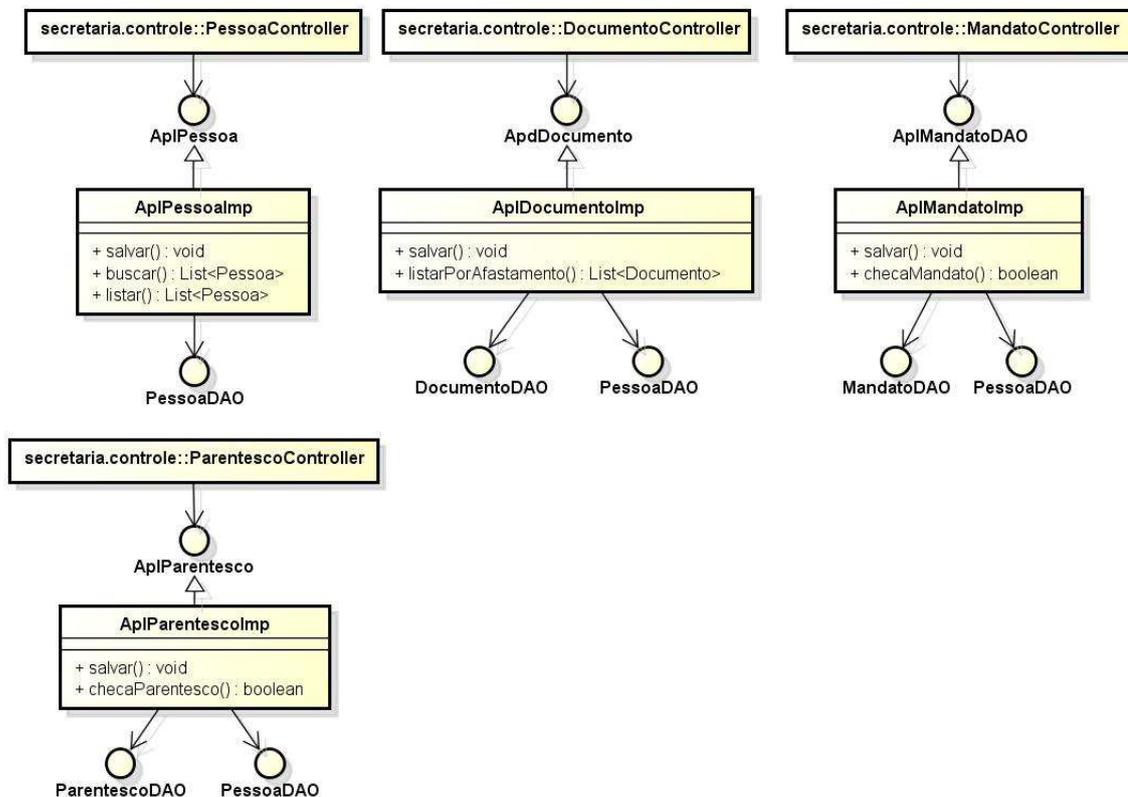


powered by Astah

Figura 16 – Modelo de aplicação do subsistema Núcleo.

Para exemplificar a lógica do diagrama podemos tomar com exemplo o caso de uso Encaminhar Afastamento, onde o Chefe do Departamento cadastra um professor como Relator de um afastamento internacional. Vemos que o RelatorController interage com a classe de serviço AplRelatorImp através da interface AplRelator. Para realizar o cadastro a classe de serviço precisa associar um professor, assim se faz necessário o PessoaDAO,

um afastamento, logo precisaremos do AfastamentoDAO e um Relator, então usa-se o RelatorDAO.



powered by Astah

Figura 17 – Modelo de aplicação do subsistema Secretaria.

4.6 Modelo de Persistência

O Modelo de Persistência é um diagrama de classes da UML que representa as classes DAO existentes, responsáveis pela persistência das instâncias das classes de domínio. Esse diagrama guia a construção das classes DAO, que pertencem ao pacote Persistência (SOUZA, 2007).

Mantendo o mesmo padrão que vimos no pacote de aplicação, cria-se uma interface para cada classe DAO do pacote. Essa interface define quais métodos devem ser implementados na classe DAO, assim o nosso sistema não fica dependente de uma tecnologia específica de persistência, pois a camada de Lógica de Negócio interage apenas com as interfaces do pacote de persistência.

De acordo com o sugerido no FrameWeb, no projeto SCAP foram criadas uma classe de persistência para cada classe de domínio que precisava ser persistida. Porém, como alguns métodos serão sempre os mesmos (Ex: `save()`, `delete()`, `merge()`, etc.), o FrameWeb propõe o uso de um DAO base, da qual as outras classes DAO herdam os

métodos mais comuns. Assim, nas classes de persistência específicas são implementados somente os métodos mais específicos de acesso a dados que a aplicação necessite.

Nas Figuras 18 e 19 são apresentados os Modelos de Persistências desenvolvidos para o projeto de SCAP, os mesmo foram usados na implementação do pacote de persistência. Vale notar que o nome das classes já indica qual tecnologia de persistência foi utilizada, esse sistema de nomenclatura é mais uma sugestão do FrameWeb para simplificar o processo de software.

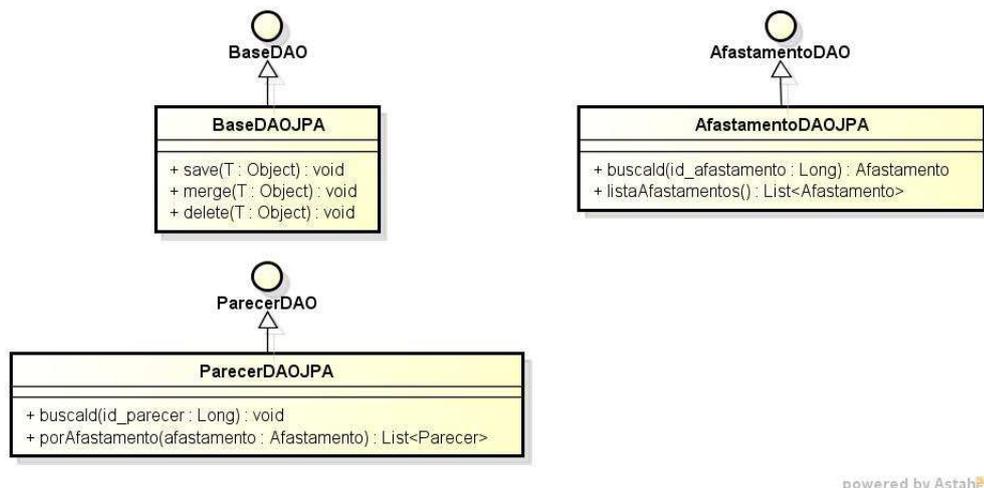


Figura 18 – Modelo de Persistência do subsistema Núcleo.

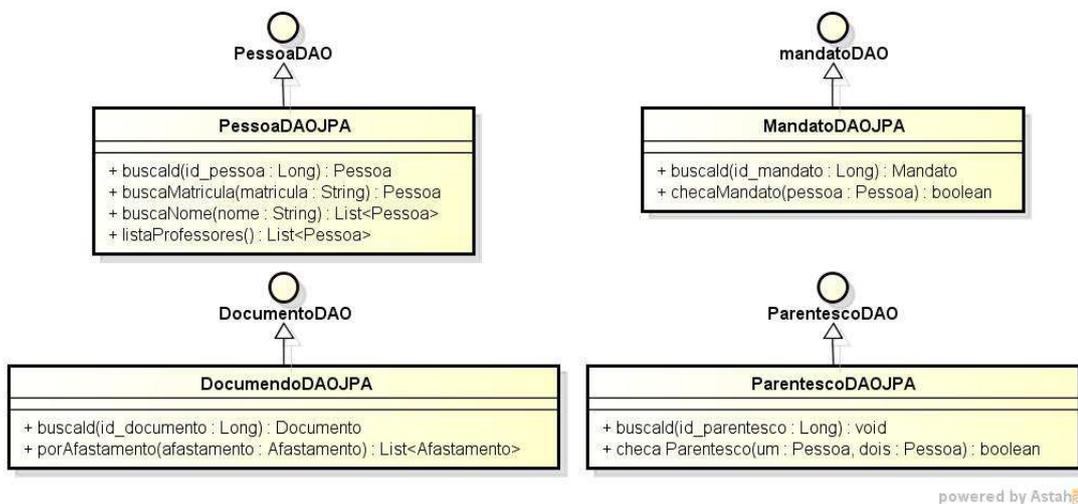


Figura 19 – Modelo de Persistência do subsistema Secretaria.

Note que a relação de herança entre os DAOs específicos e o DAO base não é representada explicitamente nos diagramas para evitar poluição visual dos mesmos. Esta também é uma recomendação de FrameWeb, ficando, portanto, o desenvolvedor incumbido de derivar essa relação implicitamente ao analisar o modelo.

4.7 Implementação do SCAP

Esta seção visa ilustrar o resultado final deste projeto, apresentando as principais funcionalidades do SCAP que foram implementadas.

Como o objetivo principal deste trabalho era a aplicação do método FrameWeb com diferentes frameworks, e não a entrega da ferramenta SCAP em si, alguns requisitos não foram contemplados por questão de tempo. A ferramenta não gera relatórios e também não é capaz de gerar as atas das reuniões que acontecem quando algum professor se manifesta contra um afastamento.

Outra funcionalidade que não foi implementada é a capacidade da ferramenta gerar os documentos atrelados aos afastamentos, ela pode apenas cadastrar um documento gerado em algum outro software e realizar a persistência do mesmo no banco de dados.

Não foi utilizado nenhum framework de decoração, pois o projeto não exige muitos recursos visuais, apenas o template HTML Bootstrap⁷ na criação das páginas, juntamente com o pacote Bootstrap table⁸ para apresentação das tabelas.

Na Figura 20 vemos a tela de login do SCAP. Sempre que o usuário acessa uma URL o interceptador **AutenticacaoInterceptor** verifica se o mesmo está autenticado no sistema, caso contrário ele é redirecionado para a tela de login. É recomendável a utilização de um framework para implementação da autenticação, porém optou-se pela criação do zero a favor de um maior aprendizado sobre interceptadores. A biblioteca usada já está inclusa no pacote do framework VRaptor 4.

Caso o usuário entre com um login inválido, inexistente ou incorreto, a classe controladora **UsuarioController** recarrega a página passando uma String com a mensagem de erro, como vemos na Figura 21.

Quando o login é realizado o usuário é direcionado para a página contendo a lista de todos os pedidos de afastamento ativos no sistema, vista na Figura 22.

Vale notar que o menu superior apresenta apenas ações referentes ao ator Secretário. Caso o usuário logado seja um professor, o menu omite os links para essas funcionalidades e apresenta apenas aquilo que o professor pode fazer.

Na Figura 23 vemos a página onde são apresentadas as informações de um afastamento, uma lista de documentos referentes ao mesmo e as opções que o usuário tem perante tal pedido de afastamento. Assim como no menu superior, as opções só são apresentadas para usuários que tenham permissão para realizá-las. Como exemplo, a opção Cadastrar Relator só aparece caso o professor autenticado seja o atual Chefe do Departamento.

⁷ Bootstrap, <http://getbootstrap.com/>

⁸ Bootstrap Table, <http://wenzhixin.net.cn/p/bootstrap-table/docs/index.html>

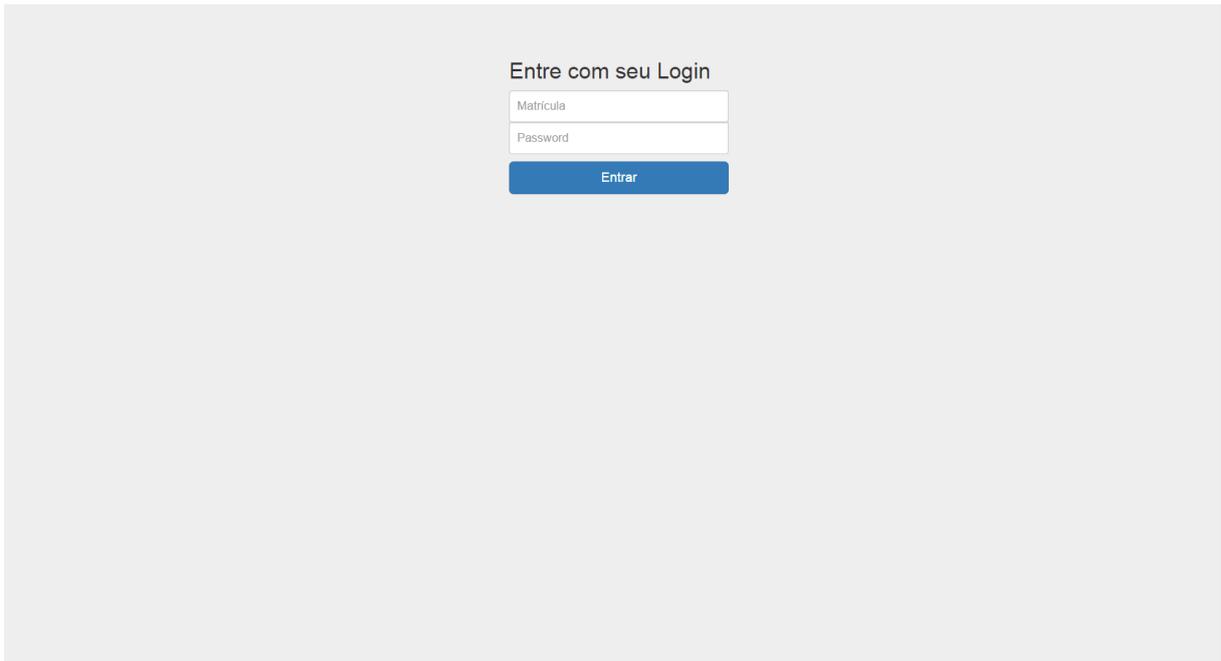


Figura 20 – Tela de login do SCAP.

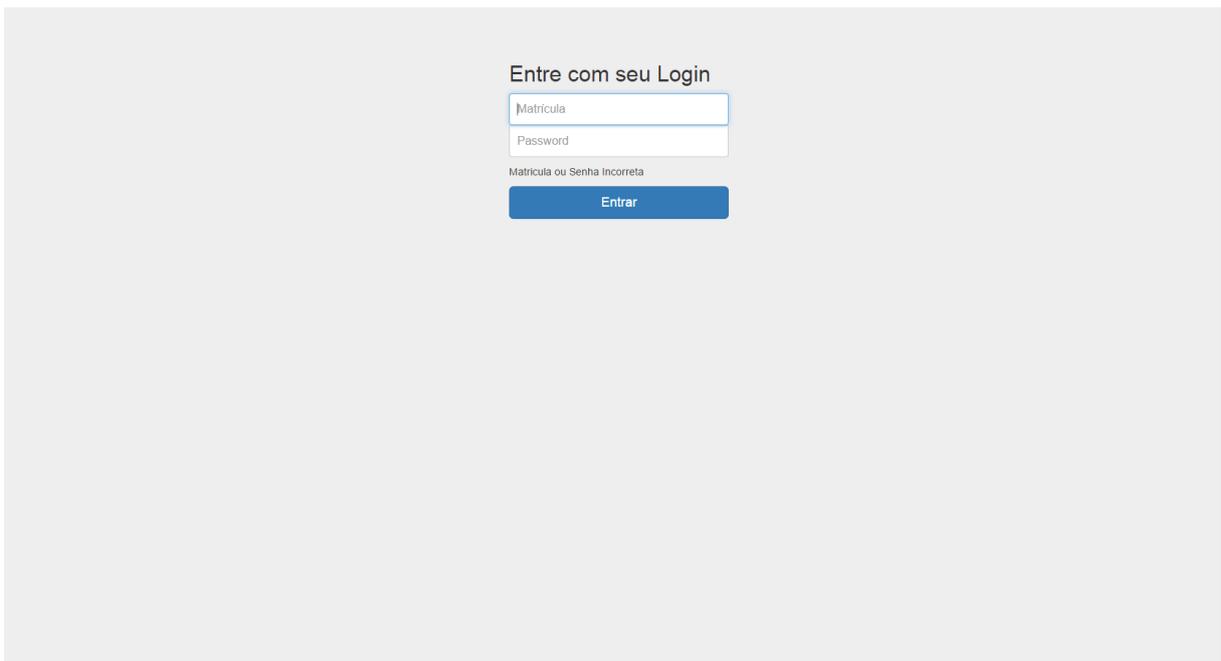


Figura 21 – Tela de login incorreto do SCAP.

Foram criadas 3 anotações na implementação do projeto: **@ProfessorRestricted**, **@SecretarioRestricted** e **@ChefeRestricted**. Essas anotações são referentes as permissões que Secretários, Professores e Chefes de Departamento, respectivamente, possuem no sistema. Quando um método que possui uma dessas anotações é disparado pelo usuário, o interceptador **AutorizacaoInterceptor** verifica se o mesmo possui autorização do sistema para realizar tal tarefa, caso contrário ele é redirecionado para uma tela de erro.

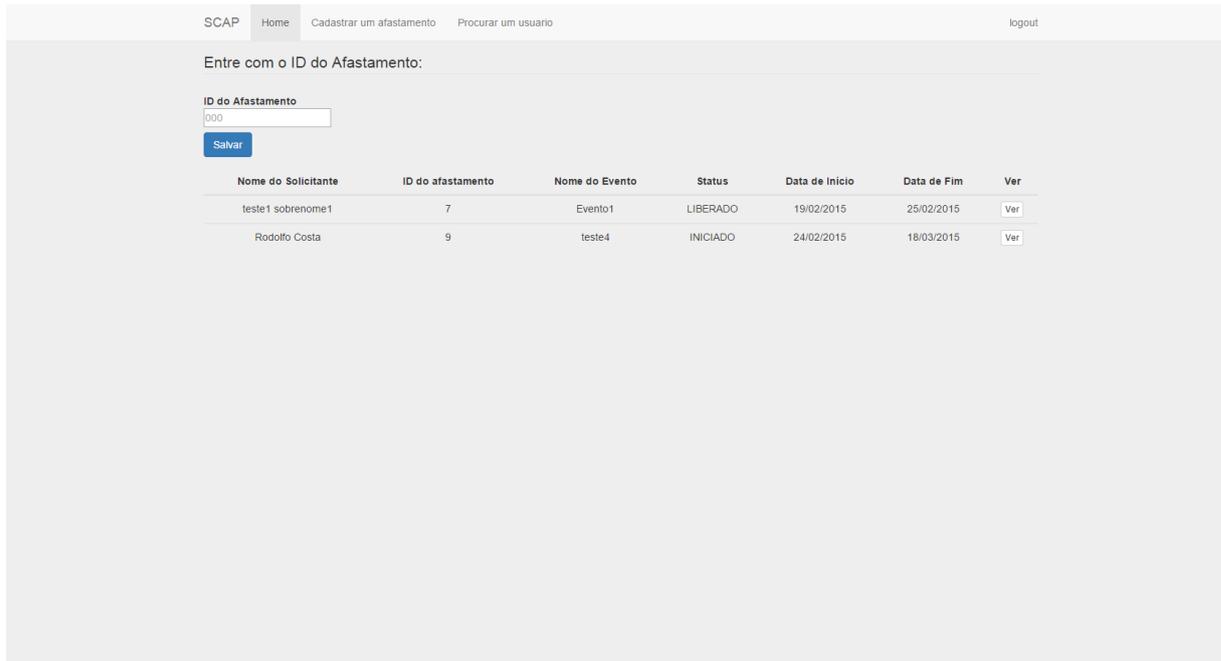


Figura 22 – Tela contendo a Lista de Afastamentos do SCAP.

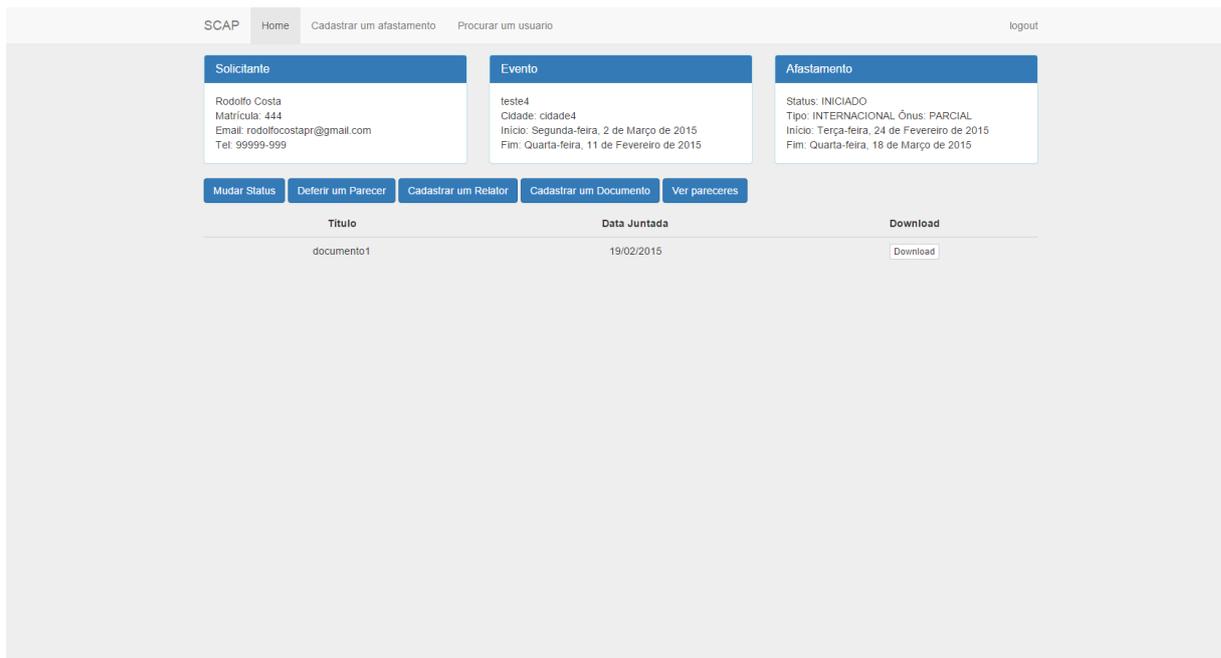


Figura 23 – Tela que apresenta os dados de um Pedido de Afastamento no SCAP.

Na Listagem 4.1 vemos como é implementada uma anotação em java, e na Listagem 4.2 vê-se como anotação é usada para especificar um método. O **AutorizacaoInterceptor** intercepta todos os métodos do sistema, quando ele percebe que o método possui uma anotação ele verifica se o usuário é autorizado a realizar um método com essa anotação.

Listagem 4.1 – código da classe responsável pela anotação @ProfessorRestricted

```

1
2 @Retention( RetentionPolicy .RUNTIME)
3 @Target({ ElementType .METHOD}) // anotação para métodos
4 public @interface ProfessorRestricted {
5
6 }

```

Listagem 4.2 – exemplo de como um método é anotado

```

1
2 @Controller
3 public class AfastamentoController {
4
5
6     @ProfessorRestricted
7     public void formulario () {
8     }
9 }

```

Todos os formulários do SCAP seguem o mesmo padrão da página apresentada na Figura 24. Para agilizar o processo de codificação dos formulários, foi usada a ferramenta *Web Form Builder for Bootstrap* do site [bootsnipp](http://bootsnipp.com)⁹.

The screenshot shows a web interface for SCAP with a navigation bar containing 'SCAP', 'Home', 'Cadastrar um afastamento', 'Procurar um usuario', and 'logout'. Below the navigation bar is a form titled 'Entre com as informações do Afastamento:'. The form contains the following fields:

- Nome Evento**: Text input field.
- Nome Cidade**: Text input field.
- Início do Afastamento**: Date input field with format 'dd/mm/aaaa'.
- Fim do Afastamento**: Date input field with format 'dd/mm/aaaa'.
- Início do Evento**: Date input field with format 'dd/mm/aaaa'.
- Fim do Evento**: Date input field with format 'dd/mm/aaaa'.
- Tipo do Afastamento**: Dropdown menu with 'Nacional' selected.
- Ônus**: Dropdown menu with 'Inexistente' selected.
- Salvar**: Blue button.

Figura 24 – Formulário de cadastro de afastamento no SCAP.

Para inclusão das datas foi usado o tipo **date** do HTML 5, pois já possui uma representação gráfica do calendário, que depois de lido e convertido para o tipo **Calendar** do Java.

⁹ bootsnipp, <http://bootsnipp.com/forms>

É importante salientar que o caso de uso “Manifestar-se contra Afastamento” foi implementado junto com o caso de uso “Registrar Parecer”. Assim quando um professor quiser se manifestar contra um afastamento basta ele cadastrar um parecer desfavorável a um afastamento. Essa junção se deve a uma questão de praticidade e pela similaridade entre os dois casos de uso.

Essa implementação do SCAP está disponível para download em: <<https://github.com/vitorsouza/tcc-rodolfo-prado>>.

5 Conclusão

Este capítulo descreve como foi utilizar o método FrameWeb, tendo o VRaptor 4 como framework controlador frontal, nesta segunda implementação do SCAP. Iremos também propor possíveis melhorias ao método baseadas na experiência adquirida no decorrer deste projeto.

O uso dos frameworks foi fundamental para simplificar a implementação do SCAP, pois além de fornecerem classes pontas para serem utilizadas, os componentes presentes nesses frameworks já foram extensivamente testados e qualquer dúvida sobre o funcionamento dos mesmos pode ser sanada com uma simples busca na Internet.

A categorização feita por Souza (2007) facilita a escolha de quais frameworks são fundamentais na implementação de um sistema Web, e os modelos propostos ajudam a traduzir muito bem o que é observado na fase de análise para o que deve ser implementado. O FrameWeb garante que as camadas do sistema mantenham um nível baixo de acoplamento, assim o SCAP pode ter seus componentes substituídos ou aperfeiçoados com muita facilidade.

Assim como observado em (DUARTE, 2014), percebeu-se também que a plataforma Java EE 7 é muito boa para a concepção de sistemas para a Web, devido a sua larga utilização no mercado ela possui um conjunto grande de frameworks e *plugins* altamente testados e documentados.

Em relação a sua primeira implementação, as seguintes mudanças foram realizadas neste projeto do SCAP:

- Foi adicionada uma regra de negócio impedindo que um professor seja relator de um afastamento requerido por um parente próximo.
- Foram desenvolvidas as classes Documento, Relator e Parentesco.
- Foram reimplementadas todas as classes, exceto as de domínio, para melhor se adequar à granularidade sugerida pelo VRaptor 4.
- Implementou-se as funcionalidades referentes aos pedidos de afastamentos internacionais.
- Às páginas Web foram todas refeitas, optou-se por uma interface mais simples.

5.1 VRaptor e o FrameWeb

O VRaptor 4 se encaixa muito bem com os modelos de navegação que são propostos no FrameWeb, algo que os programadores devem considerar ao ler os modelos de navegação. Como foi dito anteriormente o VRaptor possui certas convenções de diretório e nome que devem ser mantidas, assim varias páginas utilizadas terão exatamente o mesmo nome e são diferenciadas apenas pelas pastas onde estão localizadas, ex: uma página referente ao método “formulario” de classe controladora **AfastamentoController** deve estar localizada na pasta **WEB-INF/jsp/afastamento** do pacote de visão.

Outra consideração a ser feita é quanto as tabelas que foram implementadas no SCAP. Na Figura 22, apresentada anteriormente, vimos que existe uma tabela dos afastamento ativos no sistema, onde o usuário pode clicar no botão “ver” para ver os dados de um afastamento sem precisar preencher o formulário de busca. Porém o método FrameWeb não fornece esse nível de granularidade em relação às paginas web.

Duarte (2014) sugere a inclusão de um esteriótipo «component», que representaria um elemento da página que sofre a ação de algum outro componente, que me seu caso eram partes da página atualizadas através do Ajax. Poderíamos aproveitar tal estereótipo para representar elementos dentro da página Web que direcionam a navegação do usuário, como as tabelas presentes no SCAP. Assim fica mais especificado como deve ser implementada a navegabilidade do sistema.

A Figura 25 representa o modelo de navegação do caso de uso Consultar Solicitação usando o estereótipo *component* para representar a tabela.

5.2 Trabalhos Futuros

Além da melhoria proposta na seção anterior, dois outros trabalhos futuros podem ser sugeridos dentro deste tema. Uma nova implementação do SCAP usando um 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 ex: PHP, Ruby on Rails etc.

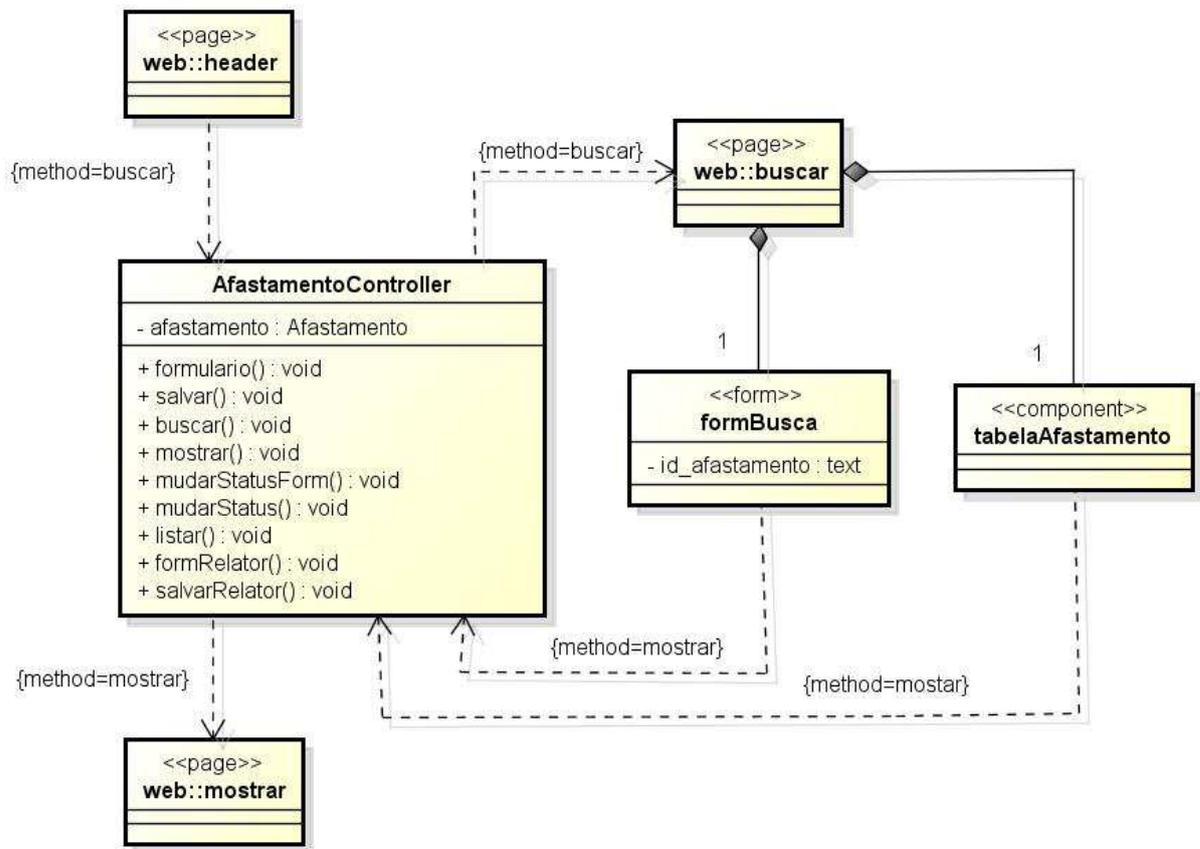


Figura 25 – modelo de navegação proposto para o caso de uso Consular Solicitação, com adição de um novo estereótipo ao FrameWeb.

Referências

- ALUR, D.; CRUPI, J.; MALKS, D. *Core J2EE Patterns*. [S.l.: s.n.], 2003. 650 p. ISBN 0131422464. Citado 2 vezes nas páginas 20 e 24.
- BAUER, C.; KING, G. *Hibernate em Ação*. 1ª edição. ed. [S.l.]: Ciência Moderna, 2005. ISBN 8573934042. Citado na página 21.
- CAVALCANTI, L. *VRaptor: Desenvolvimento ágil para web com Java*. 1ª edição. ed. [S.l.]: Casa do Código, 2014. 218 p. ISBN 9788566250268. Citado 2 vezes nas páginas 12 e 13.
- CONALLEN, J. *Building Web Applications with UML*. 2ª edição. ed. [S.l.]: Addison-Wesley, 2002. 496 p. ISBN 0785342730388. Citado 4 vezes nas páginas 7, 17, 18 e 24.
- DEMICHIEL, L.; SHANNON, B. *Java™ Platform, Enterprise Edition (Java EE) Specification, v7*. 2013. Disponível em: <<http://jcp.org/en/jsr/detail?id=342>>. Citado 2 vezes nas páginas 12 e 13.
- DUARTE, B. B. *Aplicação do Método FrameWeb no Desenvolvimento de um Sistema de Informação na Plataforma Java EE 7*. Monografia (Trabalho de Conclusão de Curso) — Universidade Federal do Espírito Santo, 2014. Citado 7 vezes nas páginas 12, 13, 26, 31, 35, 50 e 51.
- FALBO, R. *Projeto de Sistemas de Software: Notas de aula*. 2011. Disponível em: <http://www.inf.ufes.br/~falbo/files/Notas_Aula_Projeto_Sistemas.pdf>. Citado 2 vezes nas páginas 27 e 33.
- FALBO, R. *Engenharia de Requisitos: Notas de aula*. 2012. Disponível em: <http://www.inf.ufes.br/~falbo/files/Notas_Aula_Engenharia_Requisitos.pdf>. Citado na página 31.
- FOWLER, M. *Patterns of Enterprise Application Architecture*. [S.l.]: Addison-Wesley, 2002. ISBN 0321127420. Citado 3 vezes nas páginas 7, 22 e 24.
- HUSTED, T.; DUMOULIN, F. *Struts em Ação*. 1ª edição. ed. [S.l.]: Ciência Moderna, 2004. 632 p. ISBN 8573932996. Citado na página 18.
- KOCH, N. et al. Extending uml to model navigation and presentation in web applications. In: *Proceedings of Modelling Web Applications in the UML Workshop (UML'2000)*. [S.l.: s.n.], 2000. Citado na página 24.
- MURUGESAN, S. et al. Web engineering: A new discipline for development of web-based systems. *Web Engineering*, v. 2016, p. 3–13, 2001. Disponível em: <[http://www.springerlink.com/index/8ry1c125q4ujt3b9.pdf\\$%delimitador%026E30F%\\$nhttp://link.springer.com/chapter/10.1007/3-540-45144-7%2](http://www.springerlink.com/index/8ry1c125q4ujt3b9.pdf$%delimitador%026E30F%$nhttp://link.springer.com/chapter/10.1007/3-540-45144-7%2)>. Citado na página 16.
- OLSINA, L. et al. Specifying quality characteristics and attributes for websites. In: *Web Engineering*. [s.n.], 2001. v. 2016, p. 266–278. ISBN 978-3-540-42130-6. Disponível em: <<http://www.springerlink.com/content/nkwca8u2ljarttmx/>>. Citado na página 16.

- PANDOLFI, D.; MELOTTI, E. *Ciclo de Desenvolvimento de Projetos Web*. 2006. Citado na página 16.
- 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.n.], 2003. p. 65–70. Disponível em: <http://ceit.aut.ac.ir/~sa/_hashemi/MyResearch/0-SelectedPapers/2-ECommerceSystems/OOWSAMethodtoDevelopWebApplicationsfromWeb-OrientedConceptual_6.pdf>. Citado na página 17.
- PERUCH, L. A. *Aplicação e análise do Método FrameWeb com Diferentes Frameworks Web*. Monografia (Trabalho de Conclusão de Curso) — Universidade Federal do Espírito Santo, 2007. Citado 3 vezes nas páginas 16, 30 e 33.
- PRESSMAN, R. S. *Engenharia de Software - Uma Abordagem Profissional*. 7ª edição. ed. [S.l.]: McGraw-Hill, 2011. 780 p. ISBN 9788563308337. Citado 5 vezes nas páginas 12, 15, 16, 17 e 33.
- REENSKAUG, T. *THING-MODEL-VIEW-EDITOR, an Example from a planning system*. [S.l.], 1979. Citado na página 19.
- SCHWABE, D.; ROSSI, G. An Object Oriented Approach to Web-Based Application Design. *Theory and Practice of Object Systems*, v. 4, p. 207–225, 1998. ISSN 10743224. 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 (Programa de Pós-Graduação em Informática) — Universidade Federal do Espírito Santo, 2007. Citado 13 vezes nas páginas 7, 12, 13, 19, 20, 21, 22, 23, 25, 36, 38, 42 e 50.

Documento de Requisitos

Projeto: SCAP – Sistema de Controle de Afastamento de Professores

Registro de Alterações:

Versão	Responsáveis	Data	Alterações
0.1	Rodolfo Costa, Rafael dos Anjos	15/09/2014	Versão parcial inicial

1. Introdução

Este documento apresenta os requisitos de usuário da ferramenta SCAP – Sistema de Controle de Afastamento de Professores e está organizado da seguinte forma: a seção 2 contém uma descrição do propósito do sistema; a seção 3 apresenta uma descrição do minimundo apresentando o problema; e a seção 4 apresenta a lista de requisitos de usuário levantados junto ao cliente.

2. Descrição do Propósito do Sistema

O Propósito do SCAP é um sistema que visa auxiliar o controle dos afastamentos dos professores do departamento de informática da UFES, simplificando o processo de análise e armazenamento dos pedidos de afastamento. Tais afastamentos podem ocorrer devido a eventos nacionais e internacionais.

3. Descrição do Minimundo

No departamento de informática (DI) da UFES os pedidos de afastamento são solicitados para eventos no Brasil ou no exterior. Tais solicitações são avaliadas por professores do DI ou pela diretoria do Centro Tecnológico (CT) juntamente com a Pró-reitoria de Pesquisa e Pós-Graduação (PRPPG) e, caso aprovadas, o afastamento é aceito.

Para eventos no Brasil o pedido de afastamento precisa ser aprovado pelo chefe do departamento, cargo este que é exercido por algum professor do DI durante mandato temporário. Assim, para eventos nacionais o processo ocorre inteiramente dentro do departamento.

Para eventos fora do Brasil é necessário que o CT e a PRPPG aprovem o pedido e que o afastamento seja publicado no Diário Oficial da União. Porém o SCAP só lida com a tramitação dentro do DI, não existe uma integração com os processos do CT e da PRPPG. Desta forma, o escopo do sistema só trata do pedido na parte que cabe ao DI.

O SCAP existe para assistir os professores e secretários do departamento na tramitação das solicitações de afastamento, tornando mais simples todo o processo, tanto a criação do pedido quanto a análise e armazenamento do mesmo. O sistema atinge esse objetivo através do envio de e-mails automáticos aos envolvidos e da utilização de formulários na criação dos documentos necessários.

4. Requisitos de Usuário

Tomando por base o contexto do sistema, foram identificados os seguintes requisitos de usuário:

Requisitos Funcionais

Identificador	Descrição	Prioridade	Depende de
RF01	A ferramenta deve permitir o cadastro de usuários, armazenando informações como o nome, a matrícula, o e-mail, a senha, o tipo de usuário e (caso exista) o parentesco com outros usuários.	Alta	
RF02	A ferramenta deve permitir o cadastro de um chefe do departamento, que é um professor que já está cadastrado no sistema, registrando a data de início e fim do mandato.	Alta	RF01
RF03	A ferramenta deve permitir o cadastro de uma solicitação de afastamento, registrando o período de afastamento, período do evento, se o evento é no Brasil ou no exterior, o motivo do afastamento e o tipo de ônus da viagem.	Alta	RF01
RF04	A ferramenta deve ser capaz de exibir aos usuários cadastrados os dados de uma solicitação de afastamento que já esteja cadastrada no sistema.	Alta	RF01, RF03
RF05	A ferramenta deve ser capaz de cadastrar um parecer de um professor sobre uma solicitação da qual ele tenha sido escolhido como relator.	Alta	RF01, RF03, RF04
RF06	A ferramenta deve permitir que um professor se manifeste contra uma solicitação.	Alta	RF01, RF03
RF07	A ferramenta deve permitir que um professor seja apontado como relator de uma solicitação pelo chefe do departamento.	Alta	RF01, RF02, RF03
RF08	A ferramenta deve permitir que um professor cancele uma solicitação.	Alta	RF03
RF09	A ferramenta deve permitir que o secretário altere o status de uma solicitação.		
RF10	A ferramenta deve permitir que um secretário cadastre o parecer do CT e da PRPPG sobre uma solicitação.	Alta	RF03
RF11	A ferramenta deve ser capaz de mandar e-mails automáticos e gerar relatórios para os professores e para o chefe de departamento.	Médio	RF02, RF03

Regras de Negócio

Identificador	Descrição	Prioridade	Depende de
RN01	Um professor não pode ser escolhido como relator da própria solicitação.	Alta	RF01
RN02	Um professor não pode ser relator da solicitação de algum parente.	Média	RF01
RN03	Uma solicitação para eventos no exterior precisa do parecer do CT e da PRPPG antes de ser aprovada.	Alta	RF01, RF03, RF04
RN04	Uma solicitação só pode ser cancelada pelo professor que a criou.	Alta	RF01
RN05	Uma solicitação só pode ser arquivada quando a mesma já está com o status de concluída.	Alta	RF01, RF03, RF04, RF08

Requisitos Não Funcionais

Identificador	Descrição	Categoria	Escopo	Prioridade	Depende de
RNF01	A ferramenta deve prover controle de acesso às funcionalidades.	Segurança de Acesso	Sistema	Média	
RNF02	A ferramenta deve ser de aprendizado fácil, não sendo necessário nenhum treinamento especial para seu uso.	Facilidade de Aprendizado	Sistema	Média	
RNF03	A ferramenta deve ser de fácil operação, não sendo necessário uso contínuo para uma boa operação do sistema.	Facilidade de Operação	Sistema	Alta	
RNF04	A persistência das informações deve ser implementada usando o Sistema Gerenciador de Bancos de Dados estabelecido na plataforma de implementação do Projeto. Contudo, as ferramentas devem ser projetadas para, no futuro, ser possível utilizar outra tecnologia de bancos de dados.	Portabilidade	Sistema	Alta	
RNF05	A ferramenta deve estar disponível como uma aplicação Web, acessível a partir dos principais navegadores disponíveis no mercado.	Portabilidade	Sistema	Alta	
RNF06	O sistema deve ser fácil de manter, de modo a acomodar novas funcionalidades ou até mesmo adaptação para organizações específicas.	Manutenibilidade	Sistema	Alta	
RNF07	O desenvolvimento do sistema deve explorar o potencial de reutilização de componentes, tanto no que se refere ao desenvolvimento com reuso quanto ao desenvolvimento para reuso.	Reusabilidade	Sistema	Média	
RNF08	O sistema deve ser desenvolvido utilizando os frameworks VRaptor 4, CDI e JPA. Seguindo o método FrameWeb.	–	Sistema	Alta	

Documento de Especificação de Requisitos

Projeto: SCAP – Sistema de Controle de Afastamento de Professores

1. Introdução

Este documento apresenta o resultado da análise dos requisitos da ferramenta SCAP. A atividade de análise de requisitos foi conduzida aplicando-se técnicas de modelagem de casos de uso, modelagem de classes e modelagem de comportamento dinâmico do sistema. Os modelos apresentados foram elaborados usando a UML.

Este documento está organizado da seguinte forma: a seção 2 apresenta os subsistemas identificados, mostrando suas dependências na forma de um diagrama de pacotes; a seção 3 apresenta o modelo de casos de uso, incluindo descrições de atores, os diagramas de casos de uso e descrições de casos de uso; a seção 4 apresenta o modelo conceitual estrutural do sistema, na forma de diagramas de classes; a seção 5 apresenta o modelo comportamental dinâmico do sistema, na forma de diagramas de estado; finalmente, a seção 6 apresenta o glossário do projeto, contendo as definições das classes identificadas.

2. Identificação de Subsistemas

A Figura 1 mostra os subsistemas identificados no contexto do presente projeto, os quais são descritos na tabela abaixo.

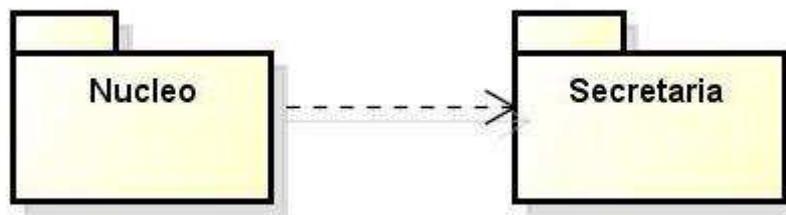


Figura 1 – Diagrama dos Subsistemas Identificados.

Tabela 1 – Subsistemas.

Subsistema	Descrição
Núcleo	É o principal subsistema do SCAP. Contém os casos de uso relacionados ao processo de solicitação de

	afastamento, com foco no papel do professor neste processo.
Secretaria	É o subsistema do SCAP criado para atender os requisitos dos secretários.

Nota: ambos os subsistemas utilizam o mesmo conjunto de classes para se referir aos elementos envolvidos nos processos de solicitação de afastamento, portanto este documento apresenta somente um diagrama de classes para todo o sistema. Se necessário, pode-se considerar que esse diagrama pertence ao subsistema Núcleo.

3. 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. Os atores identificados no contexto deste projeto estão descritos na tabela abaixo.

Tabela 2 – Atores.

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

A seguir, são apresentados os diagramas de casos de uso e descrições associadas, organizados por subsistema.

3.1 - Subsistema Núcleo

A Figura 2 apresenta o diagrama de casos de uso do subsistema Núcleo.

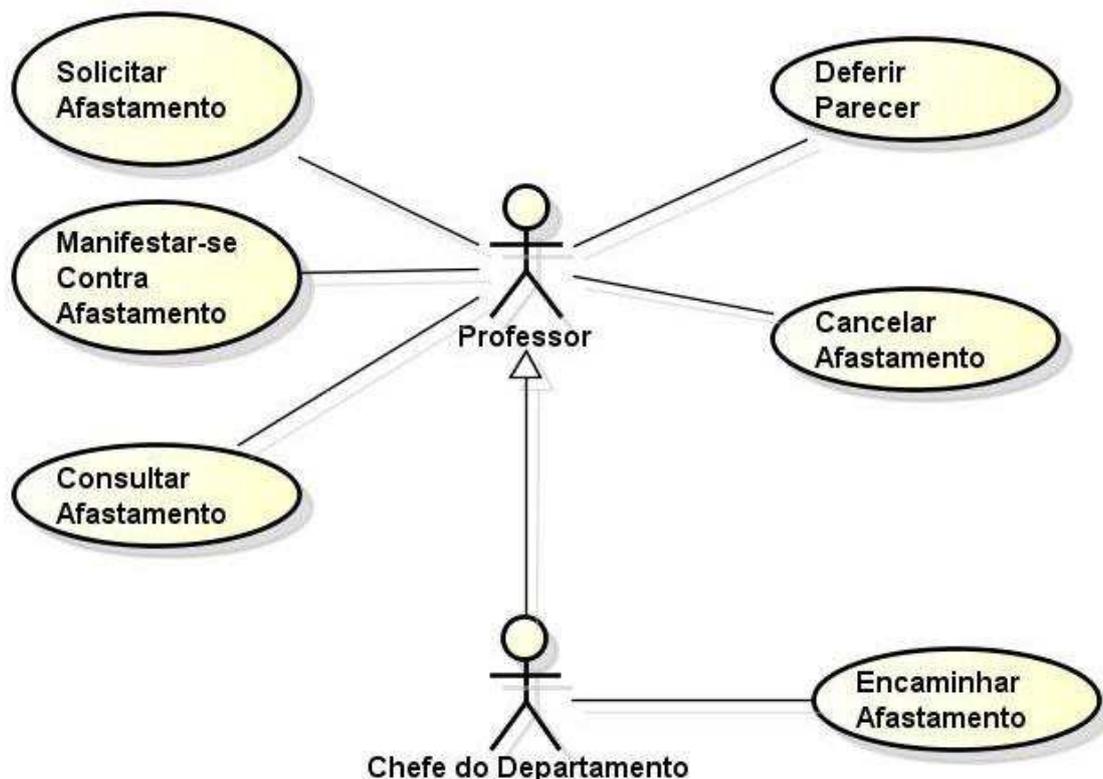


Figura 2 – Diagrama de Casos de Uso do Subsistema *Núcleo*.

A seguir, são apresentadas as descrições de cada um dos casos de uso identificados. Os casos de uso cadastrais de baixa complexidade, envolvendo inclusão, alteração, consulta e exclusão, são descritos na tabela abaixo, segundo o padrão da organização.

Tabela 3 – Casos de Uso Cadastrais do Subsistema *Núcleo*.

Subsistema	Núcleo				
Identificador	Caso de Uso	Ações Possíveis	Observações	Requisitos	Classes
CSU03	Manifestar-se Contra Afastamento	I	[I] Algum professor do DI se manifesta contra uma solicitação de afastamento. Informando sua justificativa.	RF01, RF03, RF04, RF05, RF06	Professor, Afastamento, Parecer.
CSU04	Cancelar Afastamento	E	[E] O Professor do DI que criou um afastamento pode cancelar o mesmo. Um e-mail é enviado ao chefe do departamento.	RF01, RF03, RF08	Professor, Afastamento.
CSU05	Encaminhar Afastamento	A	[A] Chefe do departamento encaminha a solicitação de afastamento para outro professor do DI que será o relator.	RF01, RF02, RF03, RF05	Professor, Afastamento, Relator.
CSU06	Consultar Afastamento	C	[C] Algum professor do DI busca através da lista ou do código de um afastamento específico uma solicitação de afastamento.	RF01, RF03	Professor, Afastamento.

Descrição de Caso de Uso

Projeto: SCAP – Sistema de Controle de Afastamento de Professores

Subsistema: Núcleo

Identificador do Caso de Uso: CSU01

Caso de Uso: Solicitar Afastamento

Descrição Sucinta: Neste Caso de Uso um Professor do DI cadastra um afastamento no sistema.

Fluxos de Eventos Normais

Nome do Fluxo de Eventos Normal	Precondição	Descrição
Aprovação de Afastamento Automática	1. Afastamento deve ser realizado para atender algum evento realizado no Brasil.	<ol style="list-style-type: none">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.4. 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 às 17 horas) a solicitação é automaticamente aprovada.5. Um e-mail automático é enviado ao professor solicitante informando o resultado final da solicitação.
Encaminhamento de Afastamento	1. Afastamento deve ser realizado para atender algum evento realizado no exterior.	<ol style="list-style-type: none">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 ao chefe do departamento avisando da nova solicitação.4. O status da solicitação é alterado para: “Iniciada”.5. Chefe do departamento encaminha a solicitação de afastamento para outro professor do DI que será o relator da solicitação de afastamento.6. O status da solicitação é alterado para: “Liberada”.7. Um e-mail automático é enviado ao professor escolhido como relator informando que o parecer é necessário.

Gerar Ata de Aprovação de Afastamento	<ol style="list-style-type: none"> 1. Afastamento deve ser realizado para atender algum evento realizado fora do Brasil. 2. Parecer do Afastamento deve estar com o status: "Aprovado-DI". 	<ol style="list-style-type: none"> 1. É gerado um trecho de ata contendo o parecer do relator e as informações do afastamento do professor para dar encaminhamento ao restante do processo que ocorre fora do DI.
--	--	--

Fluxos de Eventos Variantes

Nome do Fluxo de Eventos Normal Relacionado	Variante	Descrição
Análise de Afastamento	Passo: 2 - Encaminhamento de Solicitação de Afastamento.	<ol style="list-style-type: none"> 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.
Professor se Manifesta Contra Afastamento	Passo: 4 - Aprovação de Afastamento Automática.	<ol style="list-style-type: none"> 1. Algum professor do DI se manifesta contra o Afastamento. 2. Após uma reunião o secretário altera o status do afastamento para "aprovado-DI" ou "Reprovado-DI".

Requisitos Relacionados: RF01, RF02, RF03, RF04, RF05, RF06, RF07, RF11.

Classes Relacionadas: Professor, Afastamento, Parecer.

Descrição de Caso de Uso

Projeto: SCAP – Sistema de Controle de Afastamento de Professores

Subsistema: Núcleo

Identificador do Caso de Uso: CSU02

Caso de Uso: Deferir Parecer

Descrição Sucinta: Neste Caso de Uso um Professor do DI apontado como relator de um afastamento cadastra seu parecer.

Fluxos de Eventos Normais

Nome do Fluxo de Eventos Normal	Precondição	Descrição
Deferir Parecer Positivo	<ol style="list-style-type: none">1. Afastamento deve ser realizado para atender algum evento realizado no exterior.2. Afastamento deve estar com status: "Liberado".	<ol style="list-style-type: none">1. O professor apontado como relator preenche um formulário onde informa seu parecer positivo.2. O Parecer é cadastrado e um documento é gerado pelo sistema contendo o parecer.3. O status do afastamento é alterado para: "aprovado-DI".4. Um e-mail é enviado ao professor solicitante informando a alteração de status do afastamento.5. O Afastamento ficara com o status: "aprovado-DI" até que o parecer do CT e do PRPPG seja cadastrado pelo secretário.
Deferir Parecer Negativo	<ol style="list-style-type: none">1. Afastamento deve ser realizado para atender algum evento realizado no exterior.2. Afastamento deve estar com status: "Liberado"	<ol style="list-style-type: none">1. O professor apontado como relator preenche um formulário onde informa seu parecer negativo.2. O Parecer é cadastrado e um documento é gerado pelo sistema contendo o parecer.3. Um e-mail é enviado ao professor solicitante informando o parecer, e uma reunião com os professores decide se o parecer é valido ou não..

Requisitos Relacionados: RF01, RF02, RF03, RF04, RF05, RF06, RF07, RF11.

Classes Relacionadas: Professor, Afastamento, Parecer.

3.2 - Subsistema Secretaria

A Figura 3 apresenta o diagrama de casos de uso do subsistema Núcleo.

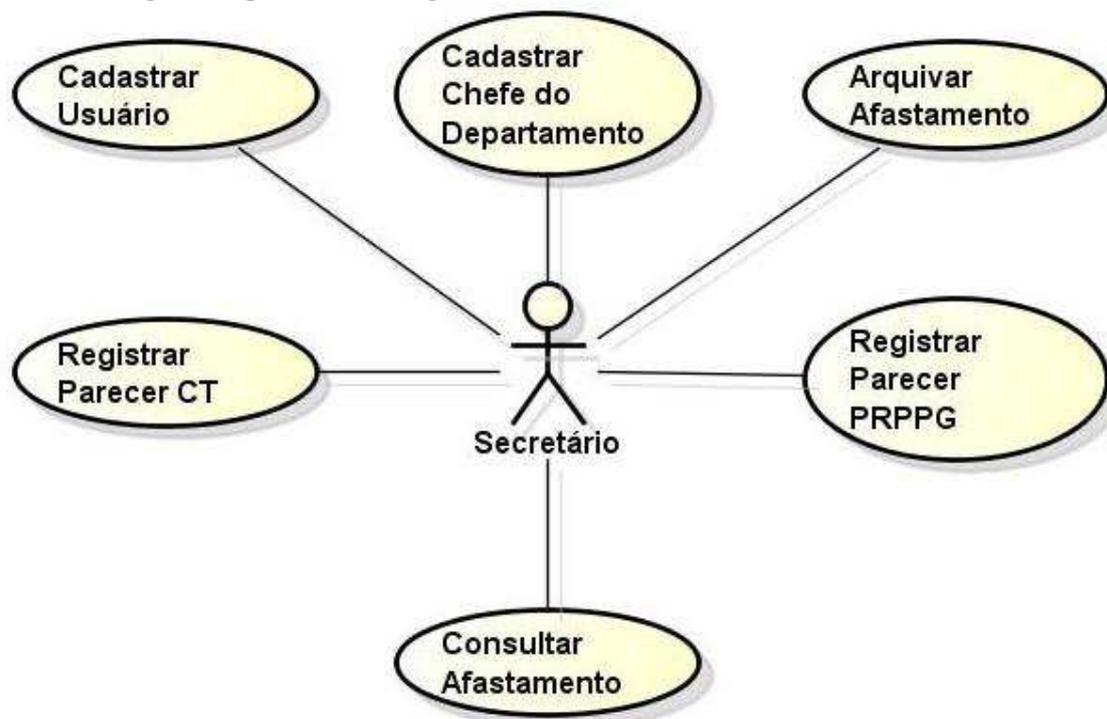


Figura 3 – Diagrama de Casos de Uso do Subsistema *Secretaria*.

Tabela 5 – Casos de Uso Cadastrais do Subsistema *Secretaria*.

Subsistema	Núcleo				
Identificador	Caso de Uso	Ações Possíveis	Observações	Requisitos	Classes
CSU07	Cadastrar Usuário	I.	[I] Informa-se: nome, matrícula, e-mail, senha e tipo de usuário.	RF01	Professor, Secretário.
CSU08	Cadastrar Chefe do Departamento	I.	[I] O secretário busca no sistema o professor que será o chefe do departamento, e cadastra seu mandato informando as datas de início e fim do mesmo.	RF01, RF02	Professor, Secretário, mandato.
CSU09	Consultar Afastamento	C.	[C] Buscar na lista ou por código um afastamento cadastrado e consultar seus dados.	RF01, RF03, RF04, RF05, RF06	Professor, Afastamento, 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.

Descrição de Caso de Uso

Projeto: SCAP – Sistema de Controle de Afastamento de Professores

Subsistema: Secretaria

Identificador do Caso de Uso: CSU10

Caso de Uso: Registrar Parecer CT

Descrição Sucinta: Neste caso de uso o Secretário cadastra o parecer do CT no sistema.

Fluxos de Eventos Normais

Nome do Fluxo de Eventos Normal	Precondição	Descrição
Cadastrar Parecer CT	<ol style="list-style-type: none">1. Afastamento deve ser realizado para atender algum evento realizado no exterior.2. Afastamento deve estar com status: "Aprovado-DI".	<ol style="list-style-type: none">1. O secretário cadastra o parecer do Centro Tecnológico.2. O secretário faz upload do documento enviado pelo CT contendo o parecer.3. Se o parecer for positivo o status do afastamento é alterado para: "Aprovado-CT".4. Se o parecer for negativo o status do afastamento é alterado para: "Reprovado-CT".

Requisitos Relacionados: RF01, RF03.

Classes Relacionadas: Secretário, Afastamento.

Descrição de Caso de Uso

Projeto: SCAP – Sistema de Controle de Afastamento de Professores

Subsistema: Secretaria

Identificador do Caso de Uso: CSU11

Caso de Uso: Registrar Parecer PRPPG

Descrição Sucinta: Neste caso de uso o Secretário cadastra o parecer da PRPPG no sistema.

Fluxos de Eventos Normais

Nome do Fluxo de Eventos Normal	Precondição	Descrição
Cadastrar Parecer PRPPG	<ol style="list-style-type: none">1. Afastamento deve ser realizado para atender algum evento realizado no exterior.2. Afastamento deve estar com status: "Aprovado-CT".	<ol style="list-style-type: none">1. O secretário cadastra o parecer da PRPPG.2. O secretário faz upload do documento enviado pela PRPPG contendo o parecer.3. Se o parecer for positivo o status do afastamento é alterado para: "Aprovado-PRPPG".4. Se o parecer for negativo o status do afastamento é alterado para: "Reprovado-PRPPG".

Requisitos Relacionados: RF01, RF03.

Classes Relacionadas: Secretário, Afastamento.

4. Modelo Estrutural

O modelo conceitual estrutural visa capturar e descrever as informações (classes, associações e atributos) que o sistema deve representar para prover as funcionalidades descritas na seção anterior. A seguir, são apresentados o diagrama de classe do subsistema Núcleo, que contém todas as classes do domínio do sistema. Na seção 6 – Glossário de Projeto – são apresentadas as descrições das classes presentes nos diagramas apresentados nesta seção.

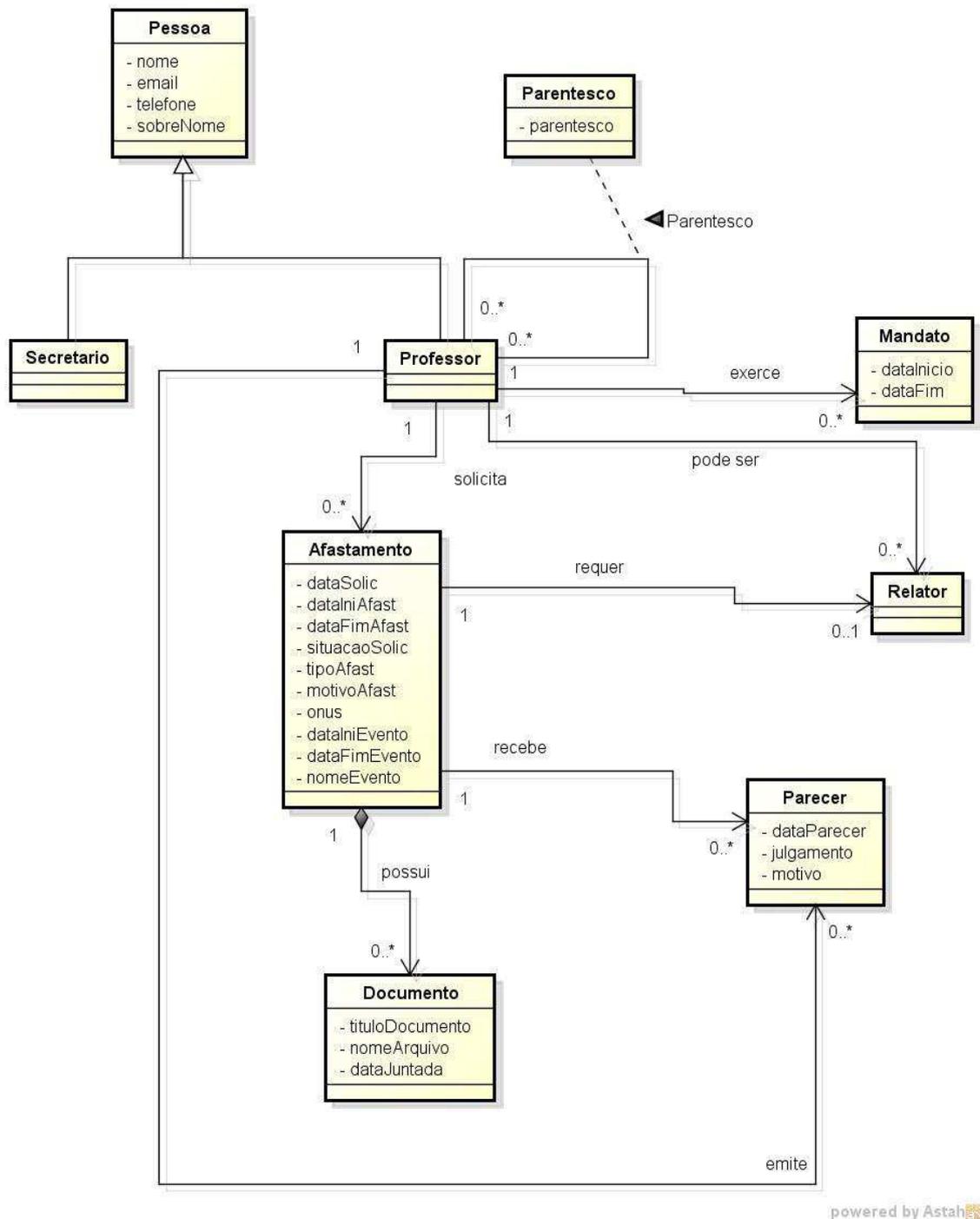


Figura 4 – Diagrama de Classes do Sistema.

5. Modelo Dinâmico

O modelo dinâmico visa capturar o comportamento dinâmico do sistema. A seguir, são apresentados os diagramas de estados elaborados no contexto deste projeto.

A Figura 5 apresenta o diagrama de estados da classe Afastamento do subsistema Núcleo.

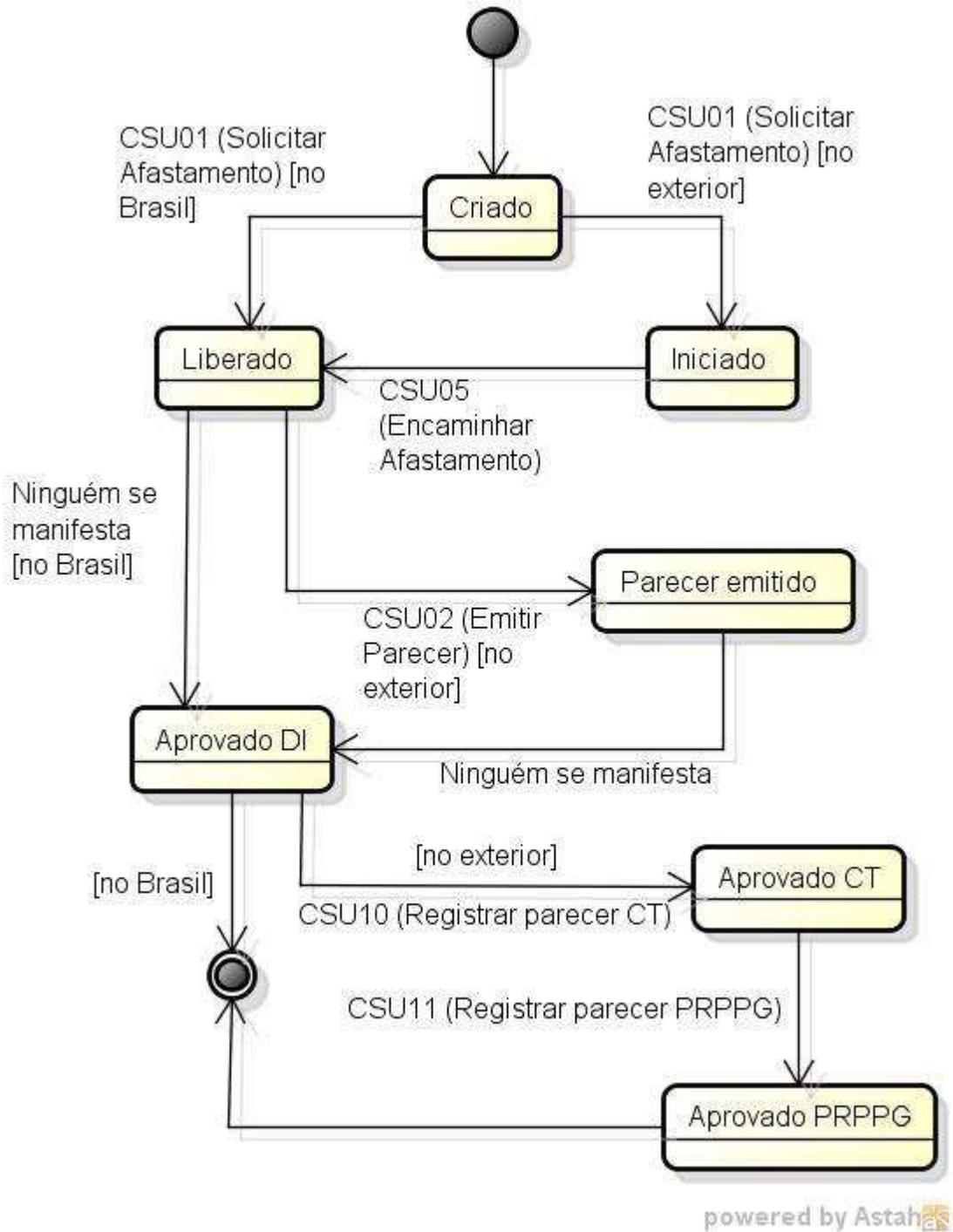


Figura 5 – Diagrama de Estados da Classe Afastamento.

6. Glossário de Projeto

Esta seção apresenta as definições dos principais conceitos envolvidos no projeto.

- **Professor:** professor do Departamento de Informática da UFES.
- **Secretário:** funcionário que trabalha na secretária do Departamento de Informática da UFES.
- **Afastamento:** pedido de afastamento de suas funções que um professor realiza para que possa atender a algum evento.
- **Chefe do Departamento:** professor do departamento que exerce a função administrativa de Chefe durante um determinado mandato.
- **Centro Tecnológico (CT):** localizado no campus de Goiabeiras, em Vitória, é composto pelos departamentos de Engenharia Ambiental, Engenharia Civil, Engenharia de Produção, Engenharia Elétrica, Engenharia Mecânica e Informática.
- **Pró-Reitoria de Pesquisa e Pós-Graduação:** a Pró-Reitoria de pesquisa e Pós-Graduação (PRPPG) conduz a política institucional da UFES de pós-graduação *strictu sensu* (cursos de mestrado e doutorado) e *lato sensu* (cursos de especialização e de aperfeiçoamento), de pesquisa (incluindo Iniciação Científica), e de inovação tecnológica.
- **Relator:** professor do DI que fica responsável em proferir um parecer em relação a um afastamento, sugerindo sua aprovação ou reprovação.
- **Parecer:** documento que registra a aprovação ou reprovação do relator, CT ou PRPPG em relação a um afastamento.
- **Parentesco:** relação de parentes sanguínea ou matrimonial entre dois professores que impede que um seja relator do afastamento do outro.

Documento de Projeto de Sistema

Projeto: SCAP – Sistema de Controle de Afastamento de Professores

1. Introdução

Este documento apresenta o documento de projeto (*design*) da ferramenta SCAP – Sistema de Controle de Afastamento de Professores. Este documento está organizado da seguinte forma: a seção 2 apresenta a plataforma de software utilizada na implementação da ferramenta; a seção 3 trata de táticas utilizadas para tratar requisitos não funcionais (atributos de qualidade); a seção 4 apresenta o projeto da arquitetura de software.

2. Plataforma de Desenvolvimento

Na Tabela 1 são listadas as tecnologias utilizadas no desenvolvimento da ferramenta, bem como o propósito de sua utilização.

Tabela 1 – Plataforma de Desenvolvimento e Tecnologias Utilizadas.

Tecnologia	Versão	Descrição	Propósito
Java	1.7.5	Linguagem de programação orientada a objetos e independente de plataforma.	Desenvolvimento de aplicativos em linguagem de programação orientada a objetos e independente de plataforma.
Java EE	7	Plataforma de Programação para servidores na linguagem Java.	Suporte na implementação do Projeto.
Java Persistence API (JPA)	2.0	API para persistência de dados por meio de mapeamento objeto-relacional.	Padronizar o mapeamento objeto-relacional por meio de uma interface de persistência comum aos principais <i>frameworks</i> de mapeamento objeto-relacional.
VRaptor	4	Framework MVC Web para desenvolvimento ágil com Java.	Framework Controlador Frontal.
Contexts and Dependency Injection for Java EE (CDI)	1.1	<i>Framework</i> gratuito para Injeção de dependências.	Integrar as diferentes camadas da arquitetura e prover serviços de transação.

Tecnologia	Versão	Descrição	Propósito
Wildfly 8	8.2	Servidor de Aplicação baseado na plataforma JEE e implementado na linguagem Java. Desenvolvido pela empresa JBoss inc.	Prover um servidor para as páginas <i>web</i> e ser compatível com a especificação Java EE 7.
Eclipse IDE for Java EE Developers	4.4	Ambiente de desenvolvimento (IDE) para a linguagem Java.	Facilitar a atividade de implementação de software.
phpMyAdmin	4.3	Gerenciador de banco de dados gratuito implementado na linguagem PHP.	Facilitar a gerencia do banco de dados da aplicação.
Vraptor Simple Mail	4.0	Plugin para envio de e-mails do VRaptor	Facilitar a criação e o envio de e-mails automáticos para os usuários do sistema.
Apache Maven	3.2.5	Ferramenta de gerência de projeto baseada em project object model (POM).	Simplificar o download das depências do projeto.
Vraptor Tasks	4.1	Pluguin para execução de tarefas agendadas.	Execução das tarefas agendadas do SCAP.

3. Atributos de Qualidade e Táticas

Na Tabela 2 são listados os atributos de qualidade considerados neste projeto, com uma indicação se os mesmos são condutores da arquitetura ou não e as táticas a serem utilizadas para tratá-los.

Tabela 2 – Atributos de Qualidade e Táticas Utilizadas.

Categoria	Requisitos Não Funcionais Considerados	Condutor da Arquitetura	Tática
Facilidade de Aprendizado, Facilidade de Operação	RNF02, RNF03	Sim	<ul style="list-style-type: none"> Prover ao usuário a capacidade de entrar com comandos que permitam operar o sistema de modo mais eficiente. Para tal, as interfaces do sistema devem permitir, sempre que possível, a entrada por meio de seleção ao invés da digitação de campos.
Segurança de Acesso	RNF01	Sim	<ul style="list-style-type: none"> Identificar usuários usando <i>login</i> e autenticá-los por meio de senha. Autorizar usuários, usando as classes de usuários definidas no Projeto ODE.

Categoria	Requisitos Não Funcionais Considerados	Condutor da Arquitetura	Tática
Manutenibilidade, Portabilidade	RNF04, RNF05, RNF06, RNF07	Sim	<ul style="list-style-type: none"> Organizar a arquitetura da ferramenta segundo uma combinação de camadas e partições. A camada de lógica de negócio deve ser organizada segundo o padrão Camada de Serviço. A camada de gerência de dados deve ser organizada segundo o padrão DAO. Separar a interface com o usuário do restante da aplicação, segundo o padrão MVC.
Reusabilidade	RNF07	Não	<ul style="list-style-type: none"> Reutilizar componentes e <i>frameworks</i> existentes. Quando não houver componentes disponíveis e houver potencial para reuso, devem-se desenvolver novos componentes para reuso.

4. Arquitetura de Software

A arquitetura de software da ferramenta SCAP, segue o que é proposto no FrameWeb onde é definida uma arquitetura lógica padrão para WISs baseada no padrão arquitetônico *Service Layer* (Camada de Serviço), proposto por Randy Stafford em (FOWLER, 2002, p. 133). O sistema é dividido em três camadas: Lógica de Apresentação, de Negócio e de Acesso a Dados.

A primeira camada contém os pacotes de Visão e Controle, a segunda contém o de Domínio e o de Aplicação e a terceira somente o pacote de Persistência. Cada pacote será explicado melhor nas próximas seções onde serão descritos os dois subsistemas do SCAP (Núcleo e Secretaria). A Figura 1 representa a visão geral das camadas e seus pacotes juntamente com o relacionamento que existe entre eles.

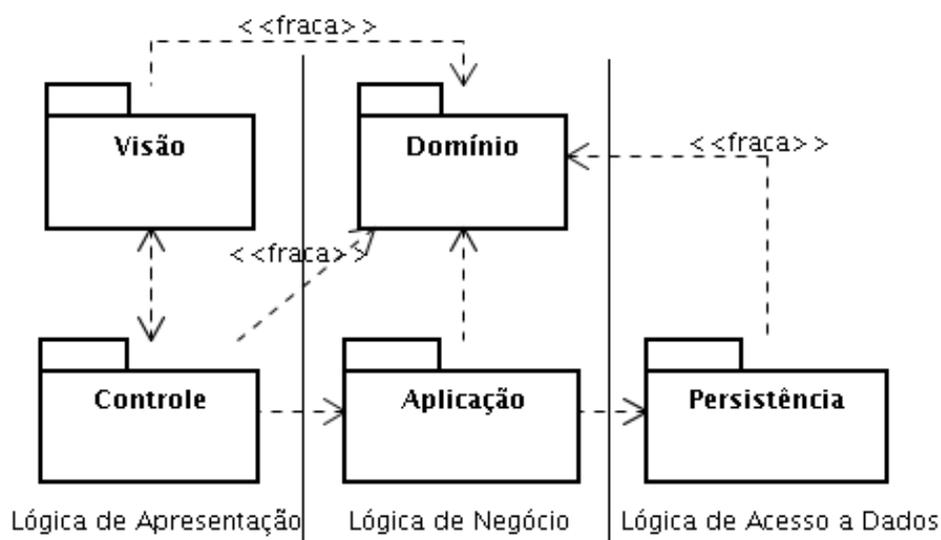


Figura 1 – Arquitetura padrão para WIS baseada no padrão arquitetônico Service Layer (FOWLER, 2002).

4.1 – Subsistema *Núcleo*

O *Núcleo* é o principal subsistema do SCAP, é nele que estão localizadas todas as classes do domínio, e é onde existe a interface de interação com os professores, que são os principais usuários do sistema.

4.1.1 – Lógica de apresentação

Tem como objetivo prover interfaces gráficas com o usuário. Isso é realizado através de dois pacotes, o de Visão, que contém páginas *web*, folhas de estilo, imagens e outros arquivos relacionados exclusivamente com a exibição de informações ao usuário. Já o pacote de Controle envolve classes de ação e outros arquivos relacionados ao *framework* Controlador Frontal. Esses pacotes são realizados a partir do Modelos de Navegação (Figura 2 a 8). Vale notar que alguns casos de uso serão implementados pelas mesmas classes, realizando as mesmas funções, e a diferenciação existe apenas a nível de usuário. Assim tais casos de uso possuem o mesmo modelo de navegação.

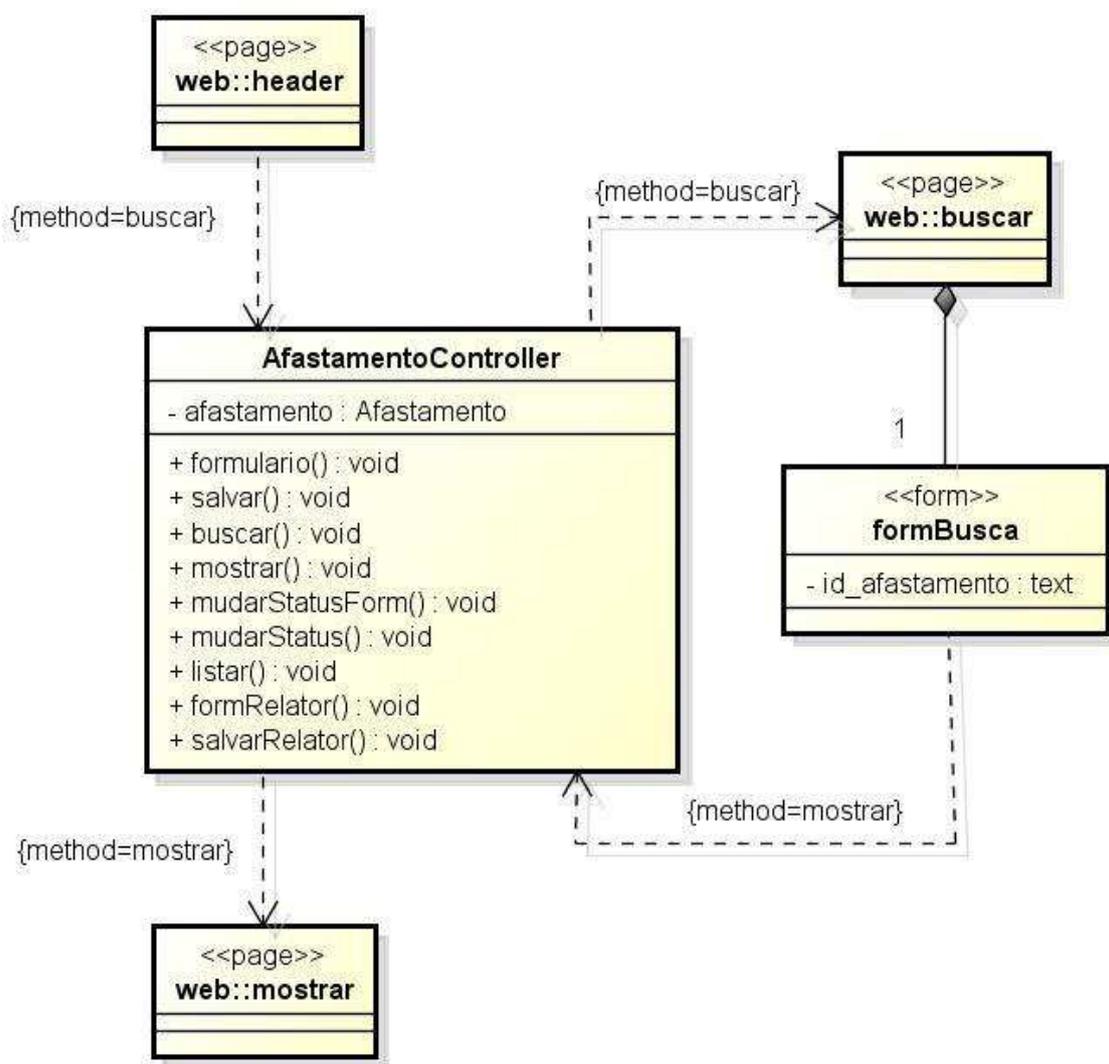


Figura 2 – Modelo de Navegação do caso de uso Cadastrar Solicitação

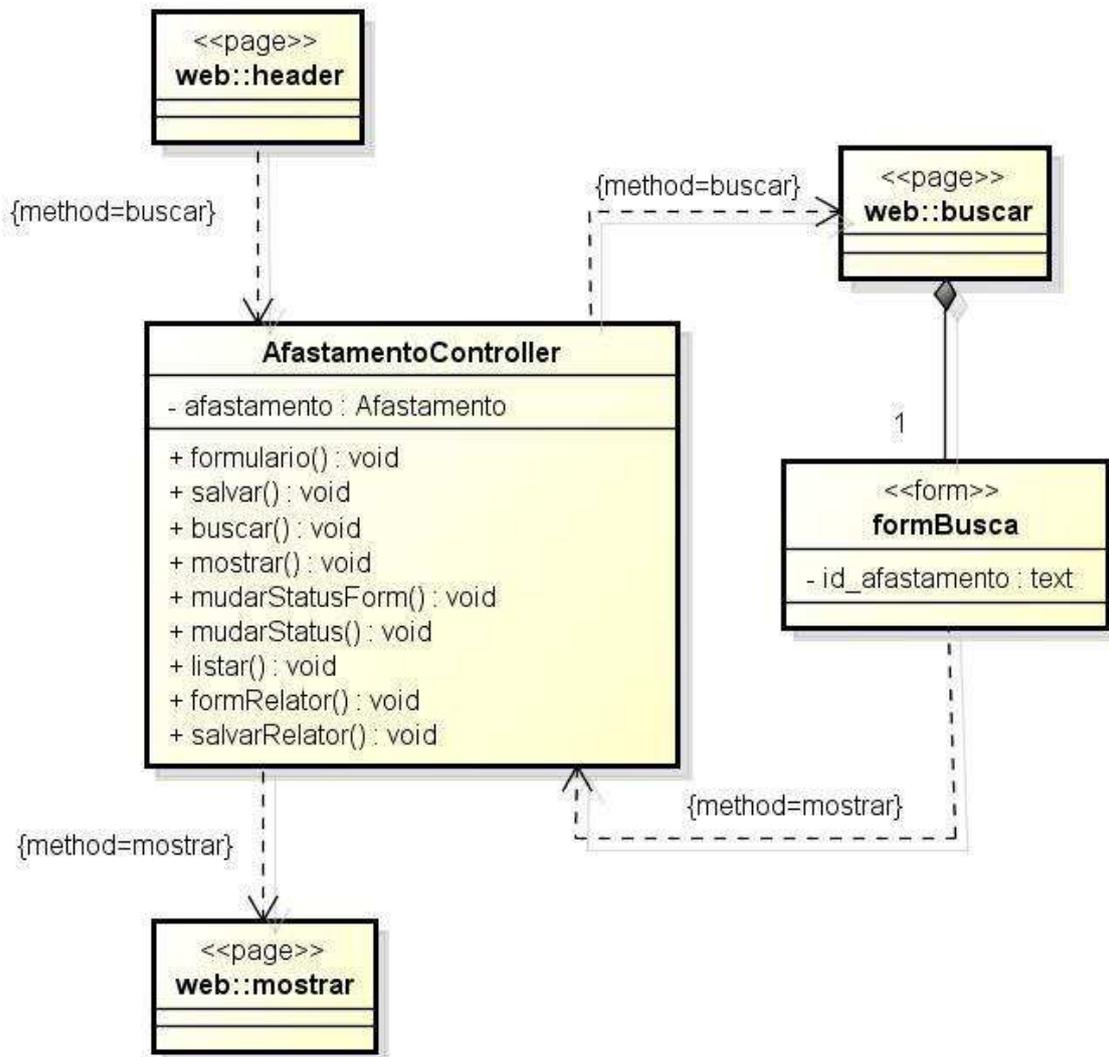
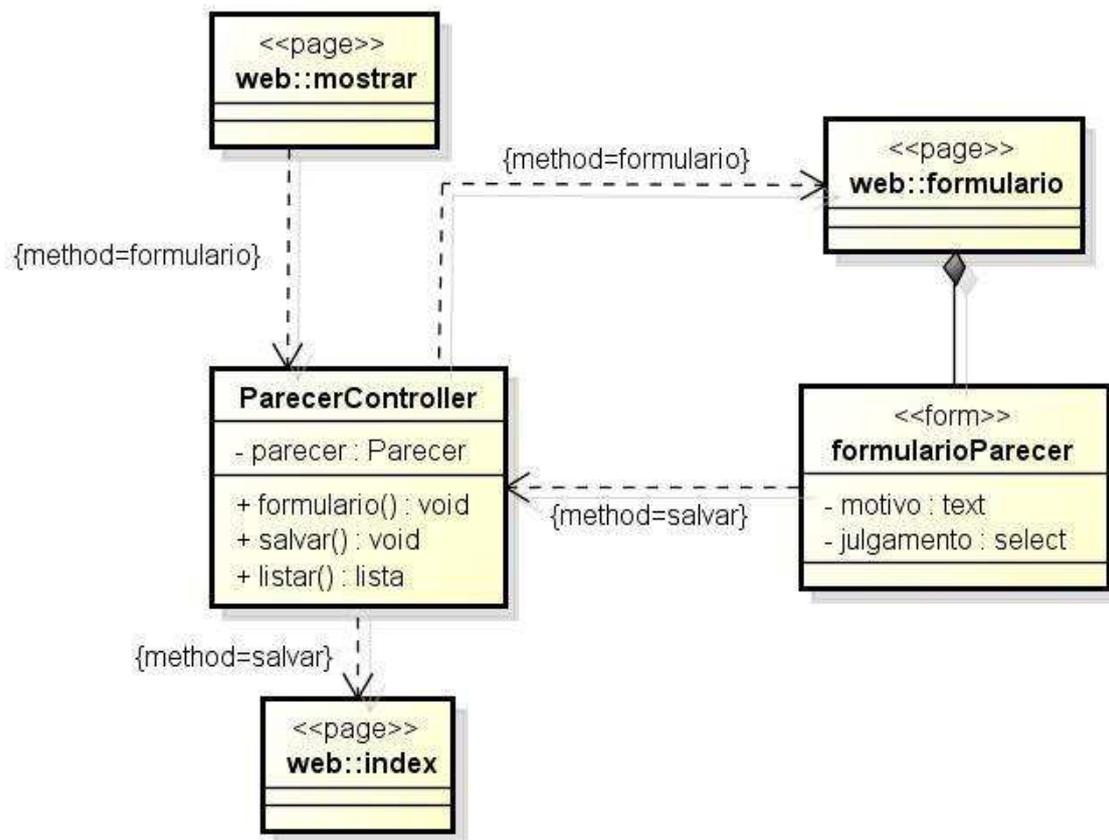


Figura 3 – Modelo de Navegação do caso de uso Consultar Afastamento



powered by Astah

Figura 4 – Modelo de Navegação que cobre os casos de uso Manifestar-se Contra Afastamento e Deferir Parecer

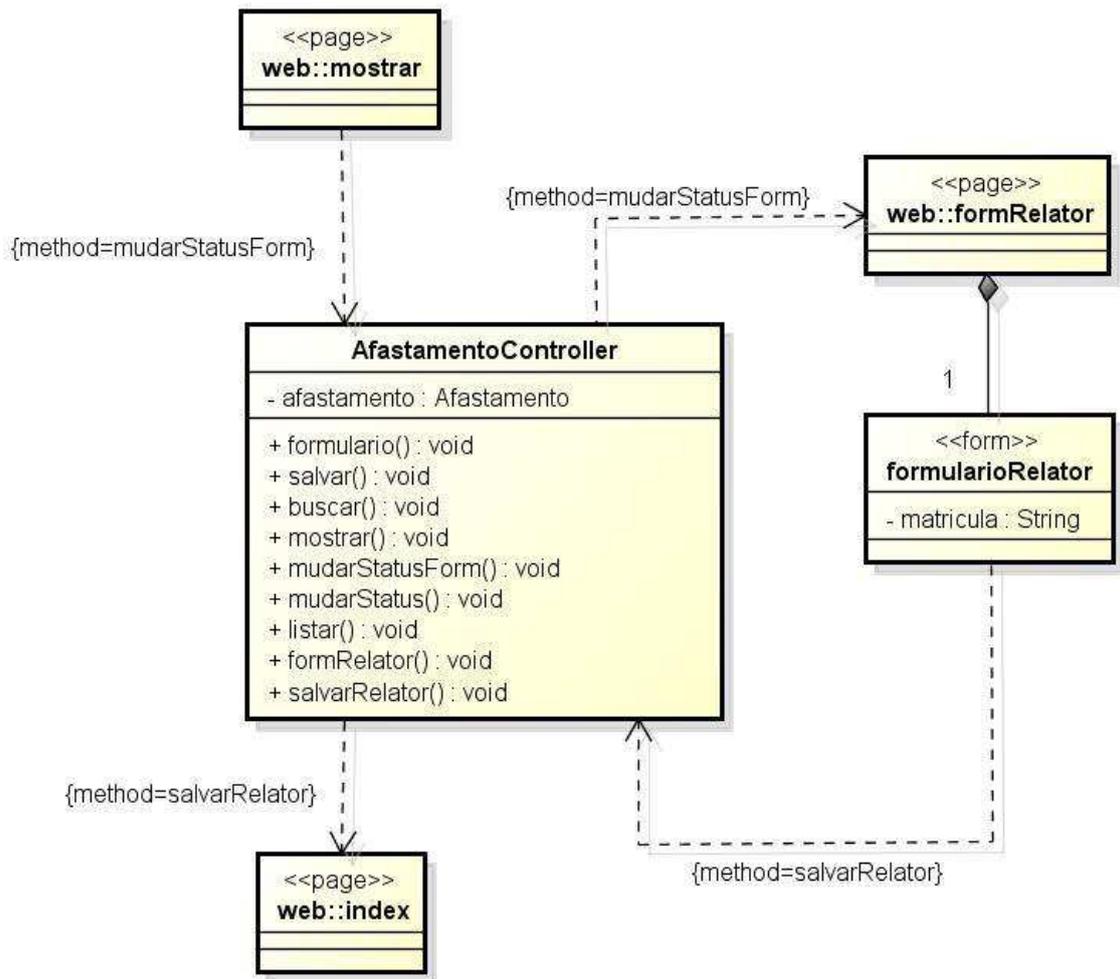
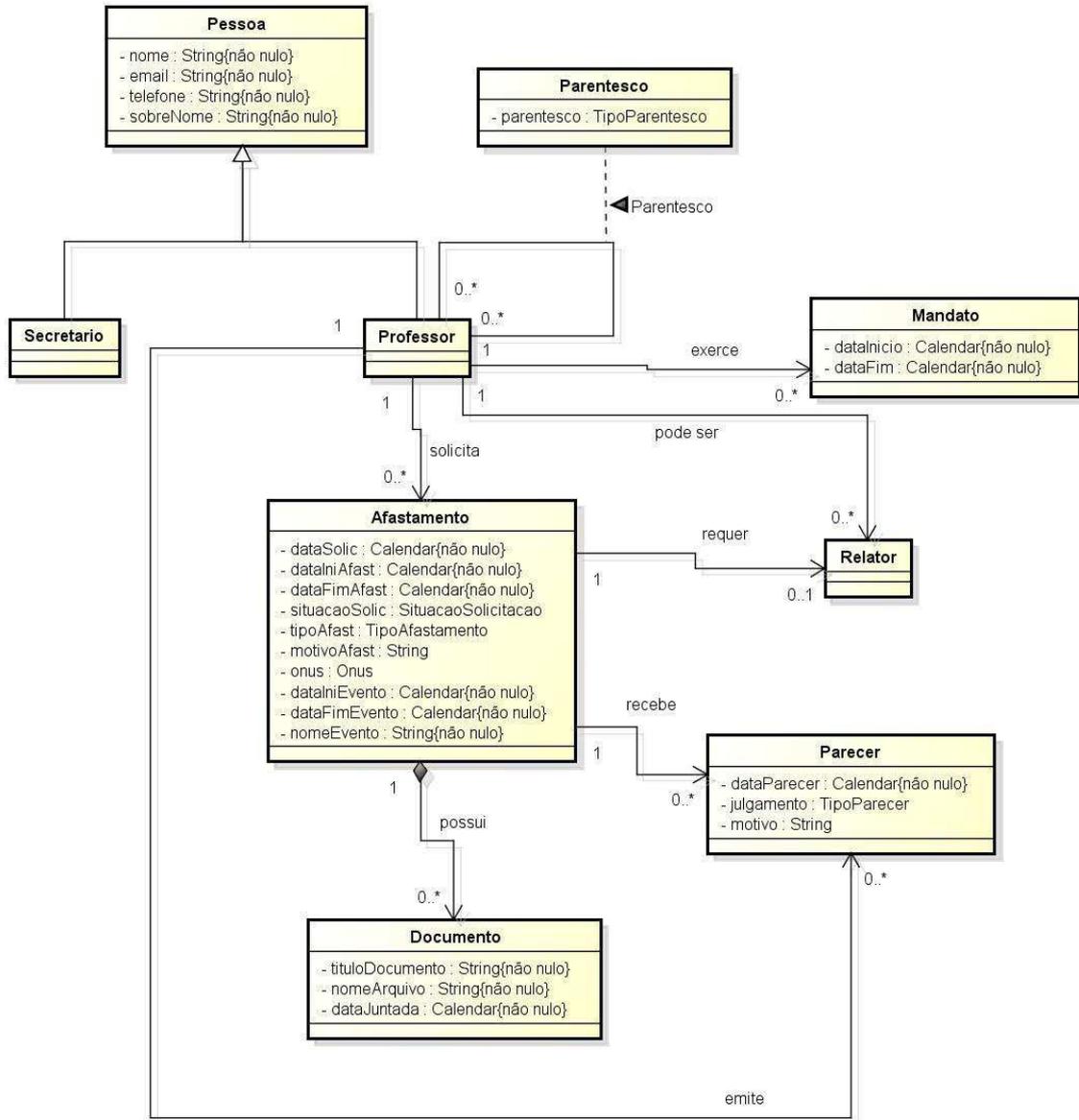


Figura 5 – Modelo de Navegação do caso de uso Encaminhar Afastamento

4.1.2 – Lógica de Negócio

A Lógica de Negócio está dividida em 2 pacotes: Domínio e Aplicação. O primeiro contém classes que representam conceitos de domínio do problema e é implementado a partir do Modelo de Domínio (Figura 4). O último tem por responsabilidade implementar os casos de uso definidos na especificação de requisitos. E é implementado a partir do Modelo de Aplicação (Figura 3).



powered by Astah

Figura 6 – Modelo de Domínio do SCAP para o subsistema Núcleo

<<enum>> SituacaoSolicitacao
INICIADO BLOQUEADO LIBERADO APROVADO-DI APROVADO-CT APROVADO-PRPPG ARQUIVADO CANCELADO REPROVADO

<<enum>> TipoAfast
NACIONAL INTERNACIONAL

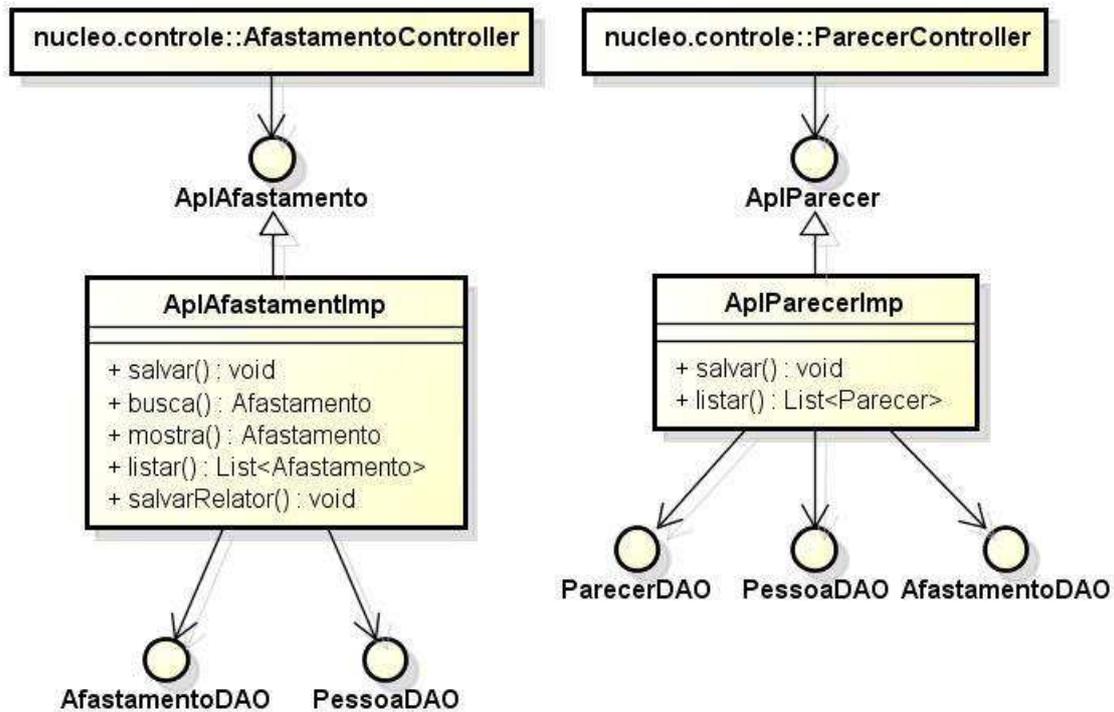
<<enum>> TipoParentesco
SANGUINEO MATRIMONIAL

<<enum>> TipoParecer
FAVORAVEL DESFAVORAVEL

<<enum>> Onus
TOTAL PARCIAL INEXISTENTE

powered by Astah 

Figura 7 –Tipos enumerados do SCAP

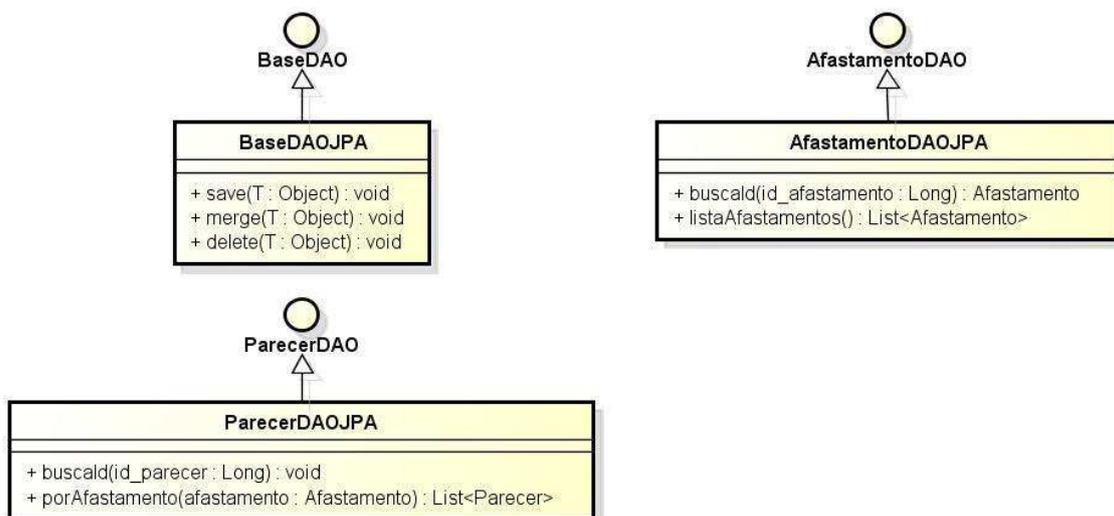


powered by Astah

Figura 8 – Modelo de Aplicação do SCAP para o subsistema Núcleo

4.1.3 – Lógica de Acesso a Dados

A camada de Acesso a Dados contém somente o pacote de persistência, que é responsável pelo armazenamento de objetos em mídia de longa duração, como bancos de dados, arquivos, serviços de nome etc. No Framework espera-se o uso de algum framework de mapeamento objeto/relacional por meio do padrão *Data Access Object* (DAO). Esse pacote é realizado a partir do modelo de persistência (Figura 3).



powered by Astah

Figura 9 – Diagrama do Modelo de Persistência do Subsistema *Núcleo*

4.2 – Subsistema *Secretaria*

É um subsistema do SCAP criado para atender os usuários da categoria Secretário. Ele não inclui nenhuma classe de domínio ou persistência, tendo somente a camada de Lógica de Apresentação. Na Figura 6 está o modelo de navegação do caso de uso Cadastrar Pessoa e na Figura 7 o Modelo de Aplicação do subsistema.

4.2.1 – Lógica de Apresentação

Segue abaixo os diagramas de navegação do subsistema Secretaria, usados na implementação dos pacotes de controle e de visão.

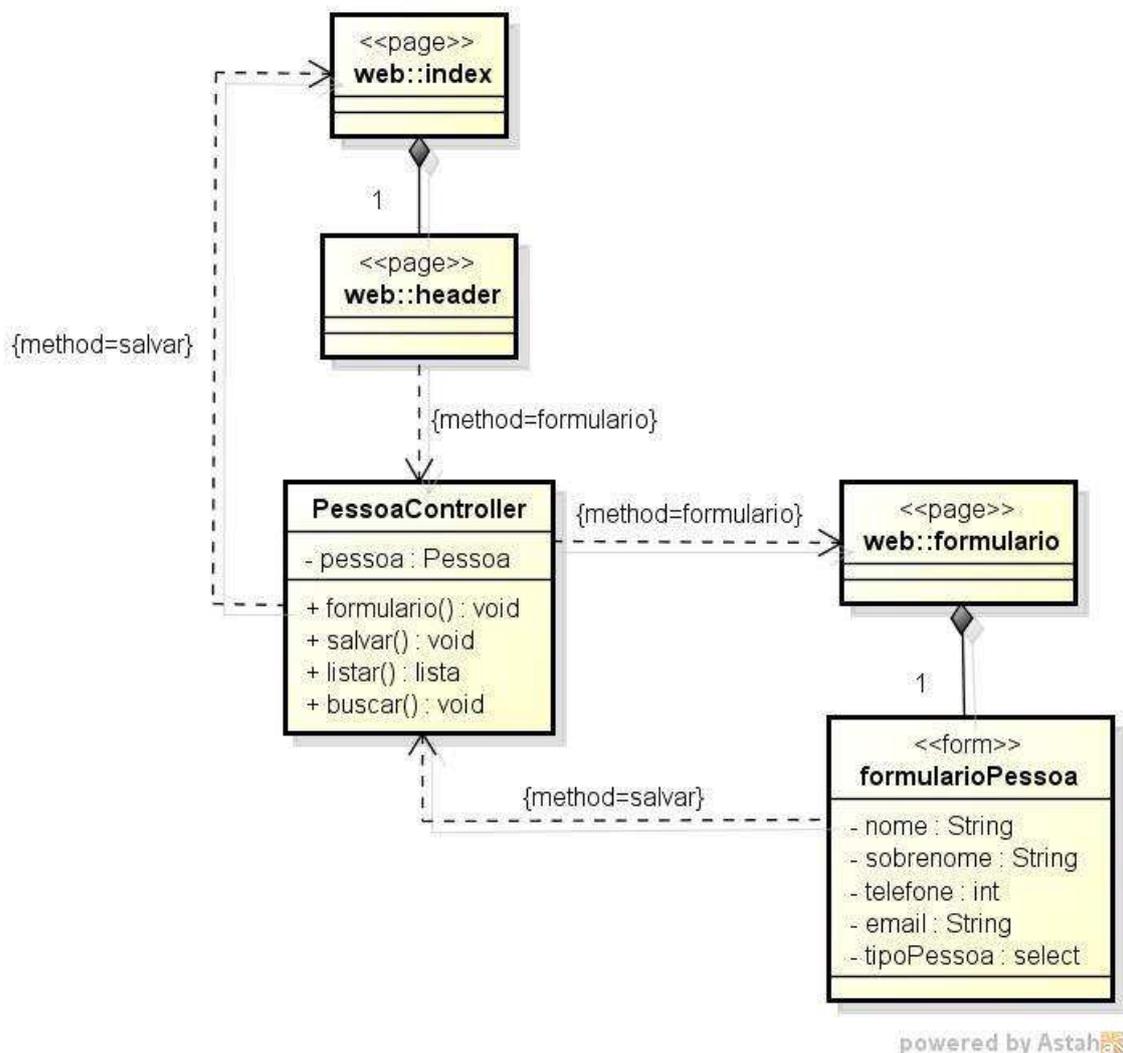


Figura 10 – Modelo de Navegação do caso de uso Cadastrar Usuário

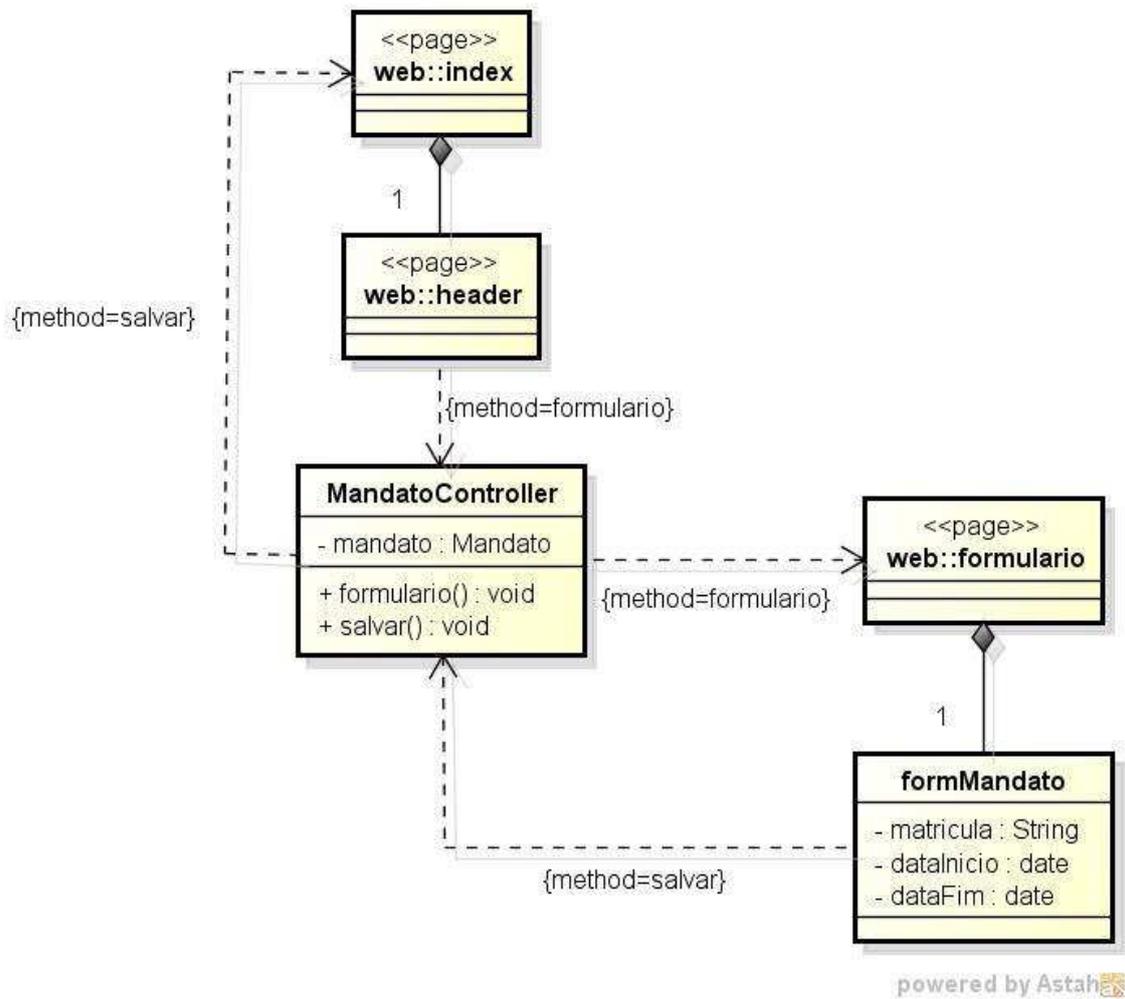
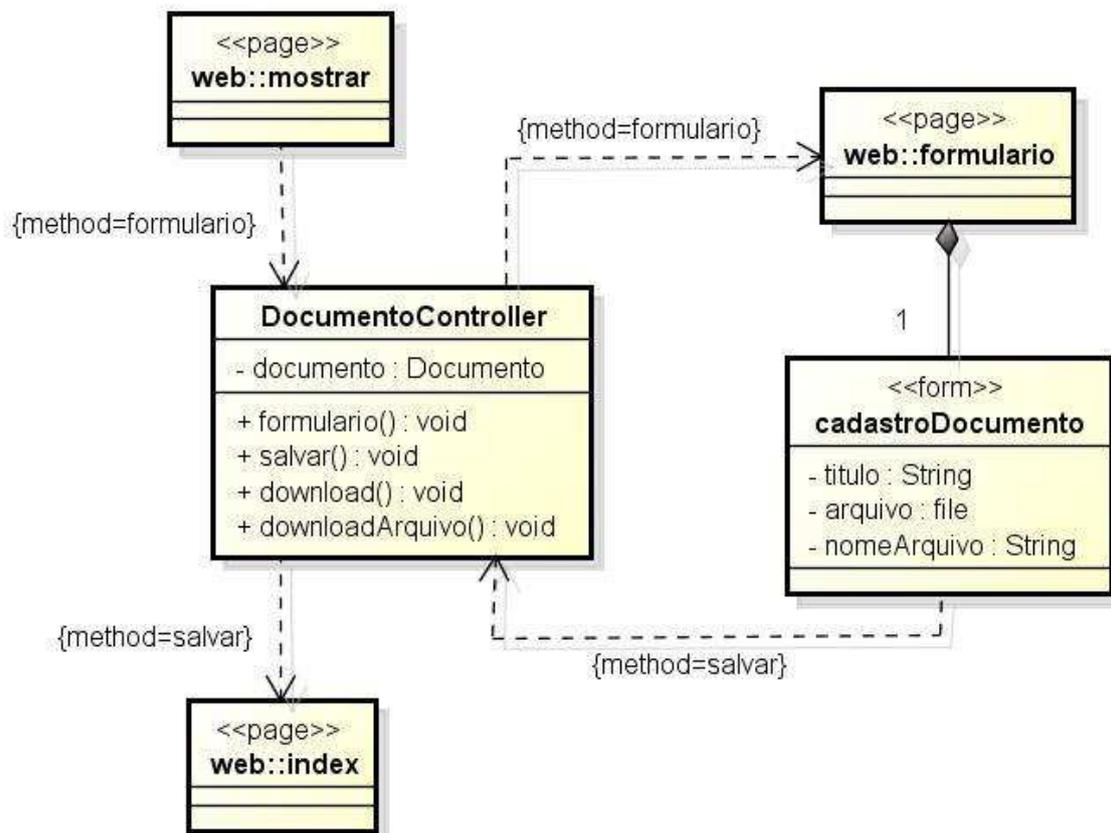


Figura 11 – Modelo de Navegação do caso de uso Cadastrar Chefe do Departamento

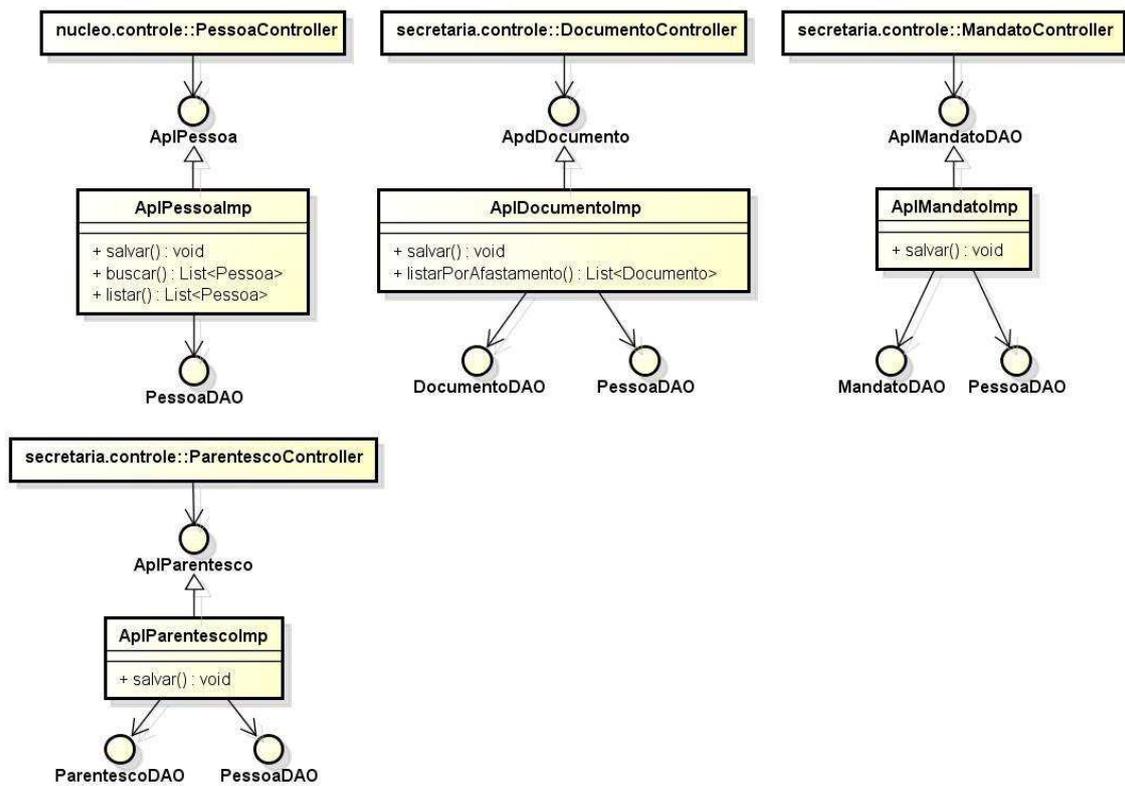


powered by Astah

Figura 12 – Modelo de Navegação que cobre os Cadastrar Parecer CT e Cadastrar Parecer PRPPG

4.2.2 – Lógica de Negócio

Segue abaixo somente o modelo de Aplicação do subsistema Secretária, já que o modelo de domínio é compartilhado entre os dois subsistemas.

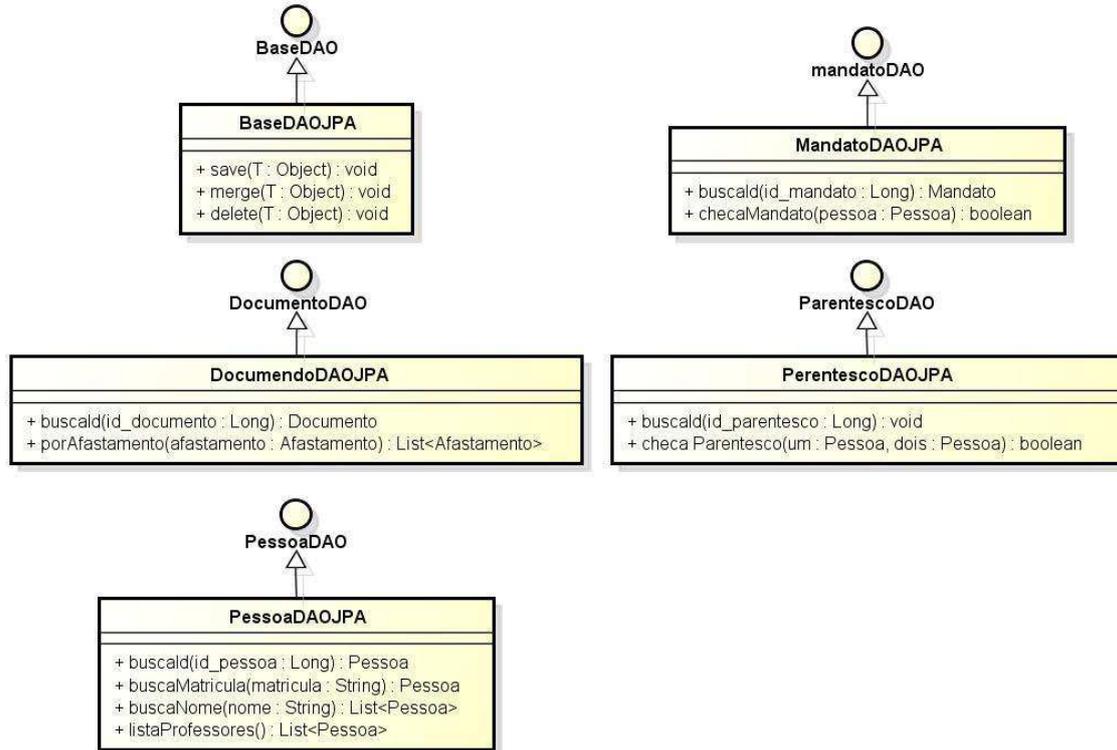


powered by Astah

Figura 13 – Modelo de Aplicação do SCAP para o subsistema Secretária

4.2.3 – Lógica de Acesso a Dados

Segue abaixo o modelo de persistência do subsistema secretaria usado na implementação do pacote de persistência do subsistema secretaria.



powered by Astah

Figura 14 – Modelo de Persistência do Subsistema Secretária

5. Referências

FOWLER, M. **Patterns of Enterprise Application Architecture**. Addison-Wesley, ISBN 0321127420, novembro 2002.