



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
COLEGIADO DO CURSO DE ENGENHARIA DE COMPUTAÇÃO

Danilo Felício Gomes

**Aplicação do método FrameWeb no
desenvolvimento do sistema SCAP utilizando o
framework Angular**

Vitória, ES

2022

Danilo Felício Gomes

Aplicação do método FrameWeb no desenvolvimento do sistema SCAP utilizando o framework Angular

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

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Colegiado do Curso de Engenharia de Computação

Orientador: Prof. Dr. Vítor E. Silva Souza

Vitória, ES

2022

Danilo Felício Gomes

Aplicação do método FrameWeb no desenvolvimento do sistema SCAP utilizando o framework Angular/ Danilo Felício Gomes. – Vitória, ES, 2022-
63 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Vítor E. Silva Souza

Monografia (PG) – Universidade Federal do Espírito Santo – UFES
Centro Tecnológico
Colegiado do Curso de Engenharia de Computação, 2022.

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

CDU 02:141:005.7

Danilo Felício Gomes

Aplicação do método FrameWeb no desenvolvimento do sistema SCAP utilizando o framework Angular

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

Trabalho aprovado. Vitória, ES, (dia) de (mês) de (ano):

Prof. Dr. Vítor E. Silva Souza
Orientador

Prof. Dr. Saulo Bortolon
Universidade Federal do Espírito Santo

Jussara Teixeira
Instituto de Tecnologia da Informação e
Comunicação do Espírito Santo

Vitória, ES
2022

Agradecimentos

Primeiramente gostaria de agradecer aos meus pais, Domingos e Dinalva e a minha tia Lia que sempre estiveram ao meu lado apoiando ao longo de toda a minha trajetória. Um agradecimento especial ao meu orientador Vitor por aceitar a me conduzir no meu trabalho de graduação e por estar sempre disponível para tirar minhas dúvidas. A professora Roberta por ter me ajudado bastante e também a todos os professores que contribuíram de alguma forma na minha formação acadêmica e agradeço também aos meus amigos e minha família por estarem sempre me motivando ao longo do curso.

Pain is temporary. It may last a minute, or an hour, or a day, or a year, but eventually it will subside and something else will take its place. If I quit, however, it lasts forever.

(Lance Armstrong)

Resumo

Atualmente a Internet é um dos mais importantes meios de comunicação, porém o seu crescimento vem acompanhado pela exigência de sistemas Web complexos e robustos. Surge então a necessidade de criar métodos e modelos para esse fim, pois os modelos tradicionais já não conseguem suprir a necessidade do mercado. Com isso, surge a Engenharia Web, que propõe métodos e ferramentas para auxiliarem no desenvolvimento, implementação e manutenção desses sistemas.

Neste contexto surge o método FrameWeb (*Framework-based Design Method for Web Engineering*), que é um método que permitir a modelagem de sistemas modernos baseado no uso de *frameworks*, com o objetivo de tornar o processo de desenvolvimento mais eficiente, pois permite que os projetistas especifiquem melhor o que deve ser implementado pelos programadores.

O FrameWeb tem como objetivo dar suporte aos mais diversos tipos de *frameworks* dentro de algumas categorias específicas (mapeamento objeto/relacional, injeção de dependências, controlador frontal, segurança) e, portanto, é interessante experimentar o método na construção de aplicações com *frameworks* diferentes.

O trabalho em questão tem como objetivo desenvolver uma nova implementação do Sistema de Controle de Afastamentos de Professores (SCAP), usando o método FrameWeb, com seus conjuntos de ferramentas, com a finalidade de avaliar sua eficiência no desenvolvimento na plataforma Angular.

Palavras-chaves Engenharia Web. FrameWeb. Angular.

Lista de ilustrações

Figura 1 – Arquitetura padrão para WIS baseada no padrão arquitetônico <i>Service Layer</i> (FOWLER, 2002).	19
Figura 2 – Arquitetura MVC através de uma representação esquemática (PRESSMAN, 2011).	22
Figura 3 – Funcionamento do padrão arquitetônico Controlador Frontal na <i>Web</i> (SOUZA, 2007).	22
Figura 4 – Padrão MVVM	23
Figura 5 – Diagrama de Casos de Uso do subsistema Núcleo (PRADO, 2015).	26
Figura 6 – Diagrama de Casos de Uso do subsistema Secretaria (PRADO, 2015).	27
Figura 7 – Diagrama de Classes do SCAP	28
Figura 8 – Representação das camadas utilizando Angular e NodeJS.	33
Figura 9 – Estrutura de diretórios do NodeJS.	34
Figura 10 – Diretórios do Angular	35
Figura 11 – Modelo de Entidades do SCAP.	37
Figura 12 – Tipos Enumerados do SCAP.	37
Figura 13 – Modelo de Persistência do SCAP	38
Figura 14 – Modelo de Aplicação do SCAP.	39
Figura 15 – Modelo de Navegação para autenticação do usuário	40
Figura 16 – Modelo de Navegação para cadastro de professor	40
Figura 17 – Modelo de Navegação para solicitação de um afastamento	41
Figura 18 – Modelo de Navegação para buscar um afastamento	41
Figura 19 – Tela de Login do SCAP.	42
Figura 20 – Tela Inicial do Secretário.	42
Figura 21 – Tela Cadastrar Professor.	43
Figura 22 – Tela Cadastrar Parentesco.	43
Figura 23 – Tela de Cadastramento De um Mandato	43
Figura 24 – Tela Inicial do Professor no SCAP.	44
Figura 25 – Tela de Solicitação de Afastamento.	44
Figura 26 – Tela de Cadastramento de Documento.	45
Figura 27 – Tela para Deferir um Parecer.	45
Figura 28 – Tela Lista de Afastamento	46
Figura 29 – Tela de Visualizar Afastamento.	46
Figura 30 – Tela Editar Afastamento.	46

Lista de tabelas

Tabela 1 – Atores do sistema SCAP (DUARTE, 2014).	26
Tabela 2 – Implementações anteriores do SCAP.	30
Tabela 3 – Plataforma de Desenvolvimento e Tecnologias Utilizadas	31
Tabela 4 – Softwares de Apoio ao Desenvolvimento do Projeto	32

Lista de abreviaturas e siglas

FrameWeb	Framework-based Design Method for Web Engineering
SCAP	Sistema de Controle de Afastamentos de Professores
UFES	Universidade Federal do Espírito Santo
ARPANET	Advanced Research Projects Agency Network
UML	Unified Modeling Language
DAO	Domain Access Object
MVC	Model-View-Controller
DI	Dependency Injection
WIS	Web Information System
ORM	Object-Relational Mapping
SGBDR	Sistemas de Gerenciamento de Banco de Dados Relacionais
JWT	JSON Web Token
SPA	Single-Page Applications

Sumário

1	INTRODUÇÃO	12
1.1	Motivação e Justificativa	13
1.2	Objetivos	13
1.3	Método de Desenvolvimento do Trabalho	14
1.4	Organização da Monografia	14
2	REFERENCIAL TEÓRICO E TECNOLOGIAS UTILIZADAS	15
2.1	Engenharia de Software	15
2.1.1	Engenharia de Requisitos	15
2.1.2	Projeto Arquitetural	16
2.2	Engenharia Web	17
2.3	FrameWeb	18
2.4	Frameworks	21
2.4.1	Frameworks MVC	21
2.4.2	Frameworks MVVC	23
2.4.3	Angular	23
2.4.4	NodeJS	24
3	ESPECIFICAÇÃO DE REQUISITOS	25
3.1	Descrição do Escopo	25
3.2	Casos de Uso	26
3.3	Análise do SCAP	27
4	PROJETO ARQUITETURAL E IMPLEMENTAÇÃO	30
4.1	Tecnologias Utilizadas	31
4.2	Arquitetura do Sistema	31
4.2.1	Implementação do Back-end	32
4.2.2	Implementação do Front-end	34
4.3	Modelos FrameWeb	36
4.3.1	Modelo de Entidades	36
4.3.2	Modelo de Persistência	36
4.3.3	Modelo de Aplicação	37
4.3.4	Modelo de Navegação	38
4.4	Apresentação de Resultados	40
5	CONCLUSÃO	47

5.1	Considerações Finais	47
5.2	Trabalhos Futuros	48
	REFERÊNCIAS	49
	APÊNDICES	51

1 Introdução

A Internet é formada por um conjunto de redes distintas que utilizam protocolos em comum para se comunicar entre si. O início da história começou com o projeto ARPANET, no auge da Guerra Fria, onde Departamento de Defesa dos Estados Unidos tinha como objetivo desenvolver um rede de controle com capacidade de proteger as informações de um ataque nuclear, mas a grande expansão da Internet para o público se deu no início da década de 1990 com surgimento da *World Wide Web* (TANENBAUM, 2011).

Segundo Pressman (2011), nos primórdios da *World Wide Web*, entre os anos 1990 a 1995, os sites eram estruturados por conjuntos de hipertexto linkados contendo textos e gráficos limitados. Com o decorrer do tempo a Web cresceu de forma absurda e teve um grande aumento no número de usuários, gerando uma enorme demanda por sistemas Web. Com isso, os grandes sistemas comerciais começaram a explorar e desenvolver nesse ambiente, porém os modelos tradicionais já não eram capazes de suprir as necessidades do mercado (PRADO, 2015). Para auxiliar no desenvolvimento, foram aplicados os conceito da Engenharia de Software nos sistemas Web ou aplicações Web, que se caracterizam por serem altamente interativos e serem acessados por meio da Internet (SOUZA, 2007). Criou-se assim uma nova subárea da Engenharia de Software chamada Engenharia Web, que define novas abordagens para auxiliar no desenvolvimento, manutenção e implantação de aplicações Web (PRESSMAN, 2011).

Nesse cenário, surge o FrameWeb (*Framework-based Design Method for Web Engineering*) (SOUZA, 2020), que é um método para Engenharia Web que propõe a modelagem dos conceitos dos *frameworks* nos modelos de projeto de modo a facilitar a implementação. Desta maneira, a definição da arquitetura fica a cargo do projetista do sistema, facilitando a tarefa dos programadores, ou seja, isso permite que os projetistas especifiquem melhor o que deve ser implementado pelo programador, tornando assim o processo de desenvolvimento mais eficiente e produtivo (SOUZA, 2007).

Para testar o FrameWeb, diversas implementações de um mesmo sistema, chamado SCAP (Sistema de Controle de Afastamentos de Professores), foram desenvolvidas como trabalhos de conclusão de curso, por exemplo: Java EE por Duarte (2014), VRaptor 4 por Prado (2015), Spring MVC e Vaadin por Matos (2017), Wicket e Tapestry por Ferreira (2018), Ninja por Avelar (2018) e diversos outros. Ressaltando que os requisitos do SCAP foram levantados e analisados por Duarte (2014) e refinado por Prado (2015). Os outros trabalhos utilizaram estes documentos como base para gerar sua própria implementação do SCAP.

No presente trabalho será desenvolvida uma nova revisão e implementação do

SCAP, sistema Web que visa facilitar o gerenciamento de afastamento dos professores do Departamento de Informática da UFES, tornando o processo mais rápido. Serão novamente utilizados os requisitos levantados por Duarte (2014) e refinados por Prado (2015). A implementação será realizada utilizando o *framework* Angular e um conjunto de *frameworks* auxiliares, seguindo o método FrameWeb (SOUZA, 2007; SOUZA, 2020).

1.1 Motivação e Justificativa

Conforme mencionado anteriormente, Duarte (2014) foi o primeiro a experimentar o método FrameWeb no processo de desenvolvimento de um sistema chamado SCAP. Foi utilizado Java EE como a plataforma de desenvolvimento, incluindo seu conjunto de *frameworks* padrão. Sucessivamente outros trabalhos expandiram a utilização de novos *frameworks* e contribuíram para a fortalecimento e análise do método. Por exemplo, Guterres (2019) desenvolveu o SCAP utilizando o *framework* Play, Berger (2021) com os *frameworks* Yii e Symfony, Dalapicola (2021) com os *frameworks* Quasar e AdonisJS, dentre outros.¹

Neste mesmo espírito, o presente trabalho tem como motivação utilizar o Angular, que é um *framework* que vem sendo amplamente utilizado no mercado atual, seguindo o método FrameWeb (SOUZA, 2007; SOUZA, 2020). Com a finalidade de contribuir para a avaliação do método com um novo conjunto de *frameworks*.

1.2 Objetivos

O presente trabalho de conclusão de curso tem como objetivo aplicar o método FrameWeb para a implementação do SCAP (Sistema de Controle de Afastamento de Professores) a partir dos requisitos levantados por Duarte (2014) e posteriormente melhorado por Prado (2015), utilizando o *framework* Angular e com isso contribuindo para avaliação do método.

Para alcançar o objetivo geral, os seguintes objetivos específicos devem ser satisfeitos:

- Compreensão dos requisitos do SCAP;
- Projeto do sistema SCAP utilizando o método FrameWeb (SOUZA, 2007; SOUZA, 2020);
- Implementação do Sistema SCAP utilizando o *framework* Angular.

¹ Outros trabalhos de conclusão de curso que implementaram o SCAP com diferentes *frameworks* e realizando sua modelagem com o FrameWeb podem ser encontrados em <<http://www.inf.ufes.br/~vitorsouza/orientacao/trabalhos-orientados/>>.

1.3 Método de Desenvolvimento do Trabalho

Para o desenvolvimento do trabalho foram feitas as seguintes atividades

- Atividade 1: levantamento bibliográfico, estudo e leitura de materiais que fornecem informações sobre Engenharia de Software, Engenharia de Requisitos, desenvolvimento de *softwares*, Engenharia Web, método FrameWeb, requisitos do SCAP e os *frameworks* que foram utilizados;
- Atividade 2: análise dos requisitos de software do SCAP levantadas por Duarte (2014) e refinado por Prado (2015);
- Atividade 3: utilização do método FrameWeb para projetar os modelos FrameWeb referentes ao SCAP;
- Atividade 4: codificação e implementação do SCAP utilizando os *frameworks* propostos;
- Atividade 5: elaboração da monografia e apresentação dos resultados.

1.4 Organização da Monografia

Além desta introdução, esta monografia é composta por outros quatro capítulos:

- O Capítulo 2 traz informações sobre o referencial teórico onde são reunidos conceitos e explicações de outros autores acerca dos assuntos relacionados a este trabalho;
- O Capítulo 3 apresenta o levantamento e análise dos requisitos do SCAP (Sistema de Controle de Afastamentos de Professores) feito por Duarte (2014) e refinado por Prado (2015);
- O Capítulo 4 traz o projeto e a implementação do SCAP utilizando o *framework* Angular com NodeJS, mostrando a aplicação do método FrameWeb na construção do sistema nesta nova arquitetura;
- O Capítulo 5 apresenta os resultados obtidos da aplicação do método FrameWeb nesta nova implementação do SCAP e as melhorias propostas para a utilização do mesmo.

2 Referencial Teórico e Tecnologias Utilizadas

Ao longo desse capítulo serão apresentados os referenciais teóricos usados para o desenvolvimento do projeto de graduação. A Seção 2.1 apresenta um breve resumo sobre Engenharia de Software, bem como conceitos de Engenharia de Requisitos e Projeto Arquitetural. A Seção 2.2 mostra o conceito da Engenharia Web, vertente Web da Engenharia de Software. A Seção 2.3 apresenta o método FrameWeb e suas propostas. A Seção 2.4 aborda os tipos de *frameworks* utilizados para o desenvolvimento de aplicações com o método FrameWeb.

2.1 Engenharia de Software

A Engenharia de Software pode ser definida como um conjunto de métodos e ferramentas que proporcionam aos profissionais desenvolver sistemas complexos com excelente qualidade e dentro do prazo estabelecido (PRESSMAN, 2011). Ela é aplicada durante todo o processo do desenvolvimento de software, desde os períodos iniciais, como a especificações do sistema, até a etapa final, quando o sistema já está sendo utilizado. Ou seja, um processo de software é uma série de atividades que no final leva à produção de um produto de software (SOMMERVILLE, 2011). Segundo Sommerville (2011) existem quatro atividades fundamentais durante os processos de *software*. São elas:

1. **Especificação de Software:** fase onde ocorre a comunicação entre as partes interessadas, geralmente cliente e engenheiro, na qual são definidos os objetivos do *software*, os requisitos e as restrições;
2. **Desenvolvimento de Software:** etapa na qual são projetados os modelos e a codificação do *software*;
3. **Validação de Software:** etapa na qual é verificado se os requisitos solicitados foram elaborados de maneira adequada;
4. **Evolução de Software:** etapa na qual o *software* é modificado para se adequar às solicitações requeridas do cliente e do mercado.

2.1.1 Engenharia de Requisitos

A Engenharia de Requisitos é uma fase de extrema importância da Engenharia de Software, porque tem como função coletar a maior quantidade de informação relevante para a produção de um sistema. É nesta etapa que o engenheiro de requisitos fica em contato

próximo do cliente para entender o domínio do problema e dar sugestões de melhorias. Essa fase tem grande importância porque, se bem elaborada, vai satisfazer às expectativas dos *stakeholders* (DALAPICOLA, 2021).

De acordo com Sommerville (2011), os requisitos de um sistema podem ser definidos como as descrições de como o sistema deve funcionar, os serviços oferecidos e as restrições, ou seja, os requisitos refletem as necessidades dos clientes para um determinado sistema. Os requisitos têm um papel central durante o processo de software, sendo responsável pelo fracasso ou sucesso do projeto, pois os requisitos bem construídos vão garantir as necessidades dos clientes (FALBO, 2017). Os requisitos podem ser classificados como requisitos funcionais e não funcionais De acordo com Sommerville (2011):

- **Requisitos funcionais:** são descrições dos serviços que o sistema deve realizar, descreve como o sistema deve reagir a uma determinada entrada, basicamente descrevem o que o sistema deve fazer;
- **Requisito não funcionais:** são restrições aos serviços oferecidos pelo sistema, geralmente relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, etc.

2.1.2 Projeto Arquitetural

O projeto de arquitetura tem o foco de compreender como o sistema deve ser organizado e entender a estrutura geral desse sistema com a finalidade de gerar um modelo de arquitetura que descreve como o sistema está organizado. É processo criativo no qual o projeto deve ser organizado para satisfazer aos requisitos funcionais e não funcionais de um sistema. Por ser um processo criativo, as atividades no âmbito do processo dependem do tipo de sistema a ser desenvolvido (SOMMERVILLE, 2011).

A fase do projeto arquitetural tem como objetivo encontrar uma solução com base nas informações que foram levantadas nas fases de levantamento e análise de requisitos e adaptá-la para incorporar as tecnologias utilizadas na implementação, sendo assim necessário ter o conhecimento das tecnologias que estão disponíveis no mercado para, durante essa etapa, definir como o problema será resolvido (FALBO, 2018). Segundo Sommerville (2011) o projeto de arquiteturas pode ser definido em dois níveis de abstração:

1. **A arquitetura em pequena escala:** está preocupada com a arquitetura de programas individuais. O foco do problema em questão é com o modo que um programa individual é decomposto em componentes;
2. **A arquitetura em grande escala:** o foco é centrado com a arquitetura de sistemas corporativos complexos que incluem outros sistemas, programas e componentes de programas.

2.2 Engenharia Web

A evolução da Internet teve um grande impacto em nossas vidas. No começo, a Internet era simplesmente usada para armazenamento de informações de forma acessível e universal e não tinha grande efeito nos sistemas de *software*. Eram sistemas executados em computadores locais. Em torno do ano 2000, a Internet começou a se expandir e diversos recursos e ferramentas passaram a serem incluídos nos navegadores. Isso gerou uma grande demanda para o desenvolvimento de novos produtos de *software* (SOMMERVILLE, 2011).

Atualmente a tecnologia Web é essencial para diversas áreas na vida, como negócios, comércio, educação, engenharia, entretenimento, finanças, governo, indústria, etc. Porém nem sempre foi assim, os primeiros sites eram bem básicos e com poucos impactos no meio social. Desta forma, é primordial que sistemas passem por um processo de engenharia chamado Engenharia Web, com o objetivo de construir e implantar uma solução eficiente e eficaz que cumpra com as expectativas de seus usuários e as estratégias de negócio pois, assim como nos demais tipos de software, é necessária a total compreensão do problema para se modelar e projetar uma solução adequada que seja implementada e testada corretamente (BEDER, 2012).

A Engenharia Web pode ser definida como a utilização de princípios, conceitos e métodos da Engenharia de Software, de uma maneira que se adaptem às características inerentes de aplicações Web (BEDER, 2012). As características de uma aplicação Web são diferentes das de um software convencional (PRESSMAN; LOWE, 2009). De acordo com Pressman e Lowe (2009) as aplicações Web devem apresentar algumas dessas características:

- Concentração na rede: as aplicações Web devem servir às necessidades diversificadas de clientes. No contexto de redes de computadores, temos: a Internet, Intranet (comunicação restrita ao escopo de uma organização) ou a Extranet (comunicação entre redes);
- Concorrência: as aplicações Web devem suportar um enorme volume de usuários acessando ao mesmo tempo;
- Carga imprevisível: a quantidade de usuários pode variar de uma hora para outra, dependendo do momento;
- Desempenho: apresentar o conteúdo no lado do cliente de forma rápida, para o usuário não desistir de esperar;
- Disponibilidade: as aplicações devem estar disponível 24h por dia, embora 100% disponível seja pouco provável;
- Orientada a dados: as aplicações Web tem por objetivo apresentar dados em forma de conteúdo de texto, gráficos, áudio e vídeo ao usuário final;

- Sensível ao conteúdo: as aplicações Web devem conter qualidade na exibição do conteúdo;
- Evolução contínua: as aplicações convencionais evoluem cronologicamente por uma série de versões, que são lançadas em determinado tempo, as aplicações Web por serem dinâmicas, podem ser atualizadas a cada instante;
- Urgência: as aplicações Web devem ser colocadas no mercado o mais rápido possível. Os engenheiros Web devem utilizar métodos para planejamento, análise, projeto, implantação e testes;
- Segurança: como as aplicações Web estão disponíveis para um grande número de usuário, devem proteger conteúdo de forma confidencial e oferecer modos seguros de transmissão de dados;
- Estética: a aparência de uma aplicação Web é algo fundamental. Pois facilita na venda dos produtos ou ideias.

De acordo com [Pressman e Lowe \(2009\)](#), há cinco atividades principais durante desenvolvimento de uma aplicação Web. É recomendado o uso de uma abordagem incremental, ou seja, atividades de arcabouço ocorrerão repetidamente à medida que cada incremento de aplicação Web for desenvolvido e entregue:

- **Comunicação:** início do processo, momento em que ocorre interação entre as partes interessadas com objetivo de compreender as restrições e necessidades do software e fazer o levantamento de requisitos;
- **Planejamento:** etapa onde se estabelece um plano incremental para o trabalho de desenvolvimento;
- **Modelagem:** abrange a criação de modelos (por exemplo, diagramas de classes) que auxiliam o processo de desenvolvimento;
- **Construção:** utilizando uma linguagem de programação, transforma a lógica em geração de código e o teste necessário para revelar erros no código;
- **Implantação:** consiste na entrega de um incremento ao cliente, que o avalia e oferece *feedback* com base nessa avaliação.

2.3 FrameWeb

O FrameWeb ([SOUZA, 2007](#); [SOUZA, 2020](#)) é basicamente um método de projeto utilizado para desenvolvimento de sistemas de Informação Web (*Web Information Systems*

- WISs) com base na utilização de *frameworks*. Mesmo existindo vários *frameworks* no mercado, não havia metodologia que juntasse as características dos *frameworks* para auxiliar no desenvolvimento de WISs durante a fase de projeto. O método FrameWeb surgiu com o objetivo de projetar modelos que se aproximam da implementação do sistema. O método está centrado em auxiliar nas fases do projeto arquitetural deixando a cargo dos projetistas e facilitando a vida dos programadores.

O FrameWeb tem foco na fase de Projeto e propõe o uso de uma arquitetura lógica que divide o sistema em distintas camadas, denominada padrão arquitetônico *Service Layer* (FOWLER, 2002), mostrada na Figura 1.

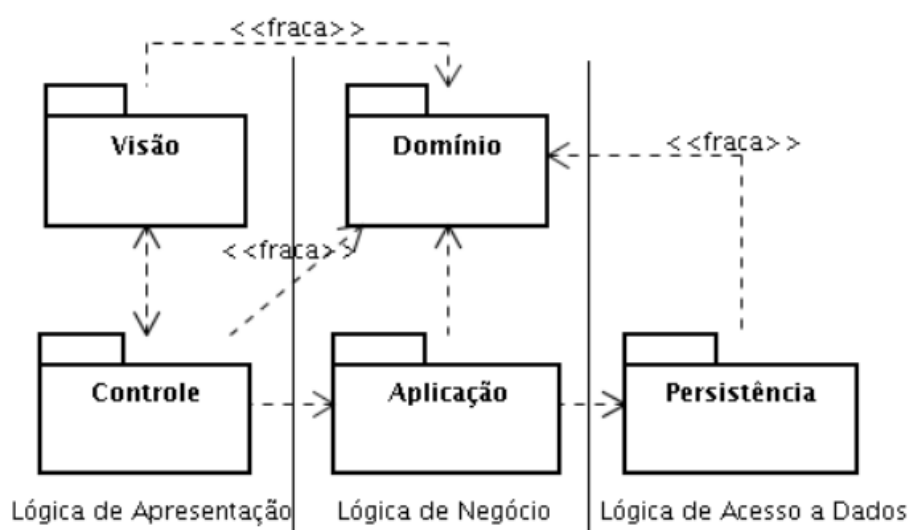


Figura 1 – Arquitetura padrão para WIS baseada no padrão arquitetônico *Service Layer* (FOWLER, 2002).

- **Lógica de Apresentação:** a lógica de apresentação visa prover a interação entre o usuário e o sistema, nessa camada ocorre uma interação direta entre o usuário e o sistema. É dividida em dois pacotes: Controle e Visão:
 - **Visão:** contém as páginas Web, folhas de estilo, conteúdo multimídia e *scripts*, sendo responsável por realizar a interação do usuário com o sistema. Para cada ação realizada pelo usuário será criado um evento que será tratado e enviado para o controle;
 - **Controle:** contém as classes que tem objetivo de controlar as interações enviando requisições para os objetos da Lógica de Negócio.
- **Lógica de Negócio:** compreende as funcionalidades da lógica de negócio, como as suas regras, conceitos do domínio. Dividida em dois pacotes: Domínio e Aplicação:
 - **Domínio:** contém as classes UML do domínio do problema, classes que foram modeladas durante a fase de análise;

- **Aplicação:** está relacionada aos modelos de Casos de Uso criados durante a etapa de Especificação de Requisitos, provê uma camada de serviço que é independente da interface de usuário.
- **Lógica de Acesso de Dados:** é a camada que faz interface com banco de dados, estabelece acesso a dados, gerencia requisições e sincroniza os dados. Contém o pacote de persistência.
 - **Persistência:** apresenta as classes que têm a função de gravar os objetos de domínio que precisam ser persistidos pelo sistema em um banco de dados. O pacote é encarregado por criar, atualizar, recuperar e remover dados que precisam ser persistidos no sistema. FrameWeb sugere o padrão DAO (*Data Access Object*) (ALUR; CRUPI; MALKS, 2003), que adiciona uma camada de abstração a mais, permitindo a separação da lógica de acesso a dados da tecnologia de persistência, ou seja, torna a aplicação independente do *framework* de persistência, podendo trocá-lo caso necessário.

Com o intuito de representar os conceitos nos *frameworks*, o uso linguagem de modelagem se tornou necessário. Então devemos modelar os artefatos que serão codificados e implementados pelos desenvolvedores. O método propõe quatro distintos tipos de modelos, que são baseados no diagrama de classes da UML:

- **Modelo de Entidades:** contém diagrama de classes UML que foi desenvolvido na etapa de análise, representado o domínio do problema. Cada classe recebe então configurações sobre seu mapeamento para a persistência em banco de dados (SOUZA, 2007);
- **Modelo de Persistência:** contém a representação de um diagrama de classes que representa as classes DAO, encarregadas de persistir instâncias das classes de entidade no banco de dados (SOUZA, 2007);
- **Modelo de Navegação:** contém o diagrama de classe da UML que representa os distintos componentes que constituem a camada lógica de apresentação, como páginas Web, formulários HTML e classes de ação do *framework Front Controller* (SOUZA, 2007);
- **Modelo de Aplicação:** contém a representação do diagrama de classes da UML encarregada por mostrar as classes de serviço, que codificam os casos de uso, e suas dependências (SOUZA, 2007).

2.4 Frameworks

Framework pode ser definido como um conjunto de código reutilizável, com a finalidade de compor uma infraestrutura de uma aplicação semi-completa. Os *frameworks* agilizam o desenvolvimento de aplicações Web (SOUZA, 2007). Sua utilização reduz o tempo de desenvolvimento, pois reduz a manutenção e maximiza o reuso de código (GAMMA et al., 2000). Souza (2007) descreveu seis categorias específicas de *frameworks*:

- *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 serão mostrados os detalhes dos *frameworks* de MVC e MVVM (*Model, View, ViewModel*, uma variação do MVC) que foram utilizados neste trabalho. Discutiremos também como o Angular e o NodeJS se encaixa nestas categorias.

2.4.1 Frameworks MVC

MVC é a abreviatura de Modelo-Visão-Controlador (*Model-View-Controller*). É um padrão arquitetural no qual a aplicação é dividida em 3 camadas, cada uma com diferentes funções. O (**modelo**) representa as informações sobre o domínio do problema. A (**visão**) tem a função de exibir as informações de forma visual ao usuário. O (**controlador**) é responsável por gerenciar as requisições dos usuários, ou seja, é a camada intermediária entre o (**modelo**) e a (**visão**) (FOWLER, 2002). A Figura 2 demonstra o funcionamento desse padrão arquitetural.

De acordo com Souza (2007) a camada de (**visão**) encontra-se do lado do cliente, é o conjunto das páginas Web exibidas pelos navegadores, por outro lado, a camada de (**modelo**) encontra-se do lado do servidor. A comunicação entre as duas é intermediada pela camada de (**controle**). Ou seja, a camada de (**visão**) é separada da camada de (**modelo**) isso garante que as modificações na camada de apresentação não tenham efeitos diretos na camada de persistência da aplicação.

Com observado por Souza (2007), a arquitetura MVC precisa de modificações para atender as necessidades dos *Webapps*. Pois o (**modelo**) localizado no lado do servidor, não é capaz de enviar as notificações das modificações para a (**visão**), já que se encontra

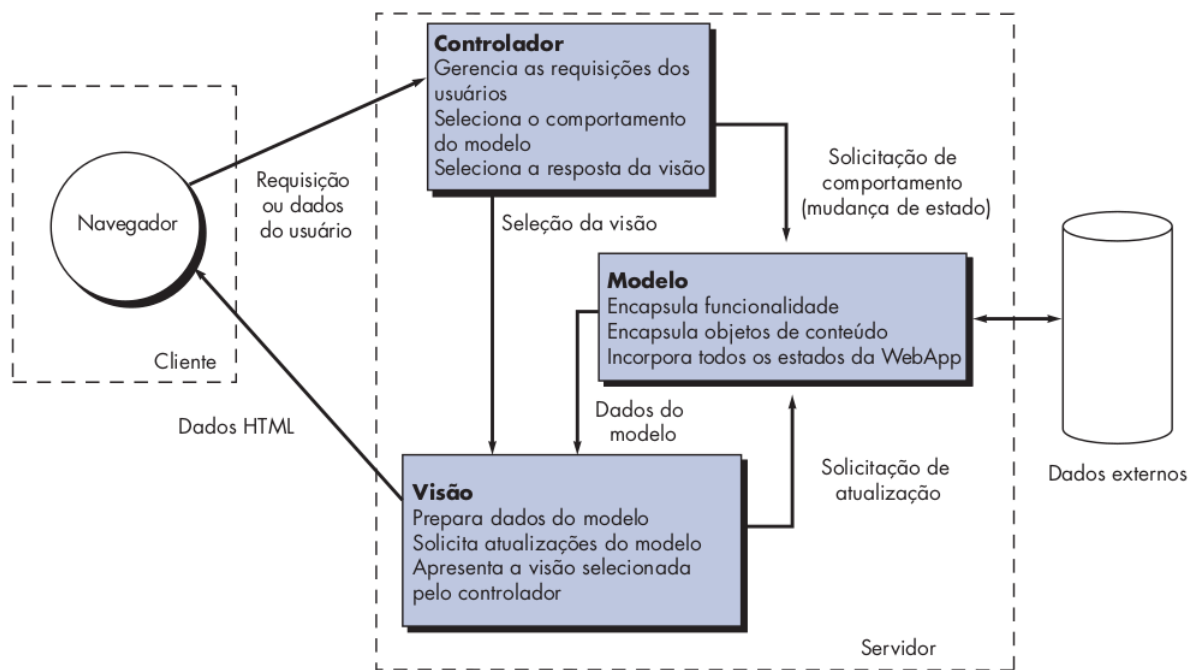


Figura 2 – Arquitetura MVC através de uma representação esquemática (PRESSMAN, 2011).

no navegador do lado do cliente. A comunicação ocorre sempre no lado do cliente. Então, apesar de MVC ser um nome bem conceituado, o nome adequado para esse padrão arquitetônico, quando aplicado à Web, seria “Controlador Frontal” (*Front Controller*) (ALUR; CRUPI; MALKS, 2003). O padrão Controlador Frontal pode ser visualizado na Figura 3.

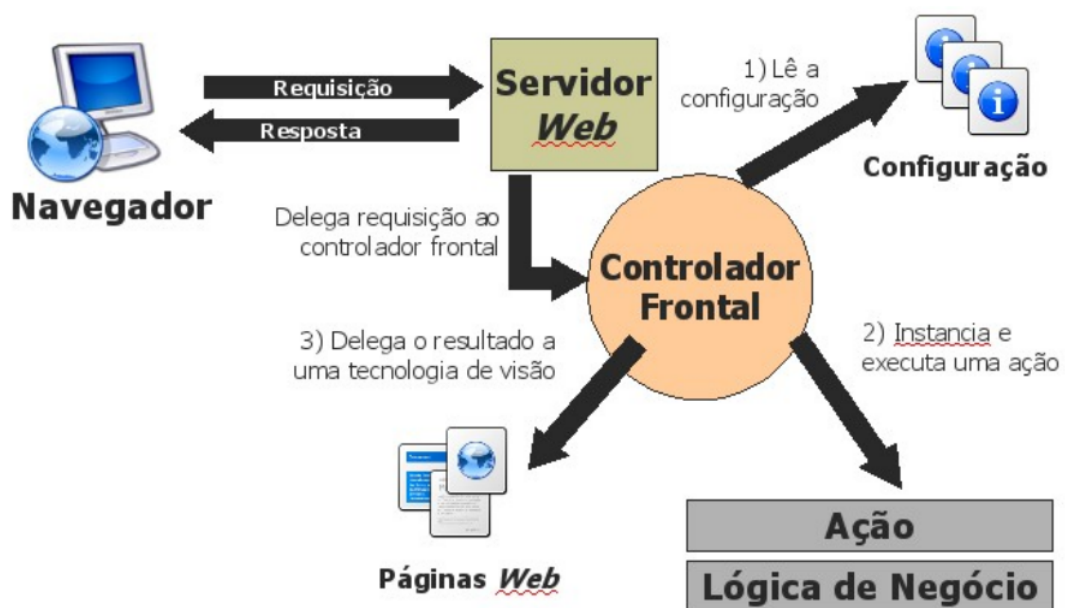


Figura 3 – Funcionamento do padrão arquitetônico Controlador Frontal na Web (SOUZA, 2007).

2.4.2 Frameworks MVVC

O padrão MVVM (*Model, View, View-Model*) foi introduzido em 2005 por John Gossman, arquiteto de *software* da Microsoft, para ser utilizado em aplicativos *Windows Presentation Foundation* (WPF). O MVVM é uma variação do padrão PM (*Presentation Model*) (FOWLER, 2002). Porém, nesse padrão é criada uma abstração da *View* e estados e comportamentos são separados da camada de apresentação (SMITH, 2009).

O MVVM é um padrão derivado do padrão MVC e, portanto, semelhante ao MVC, o *Model* representa as informações sobre o domínio do problema, representando uma camada de acesso aos dados, enquanto a *View* representa a camada de apresentação, ou seja, os componentes da interface do usuário. Porém diferentemente do *Controller* no MVC, temos um elemento chamado *View-Model* que não é separado da visualização. Ele pode invocar métodos provenientes da camada de Modelo e provê um mecanismo para sincronizar a *View* que suporta o fluxo de dados nos dois sentidos (*two-way data binding*), possibilitando, assim, uma mudança automática entre *View-Model* e *View*. Na Figura 4 é possível ver a representação desse padrão.

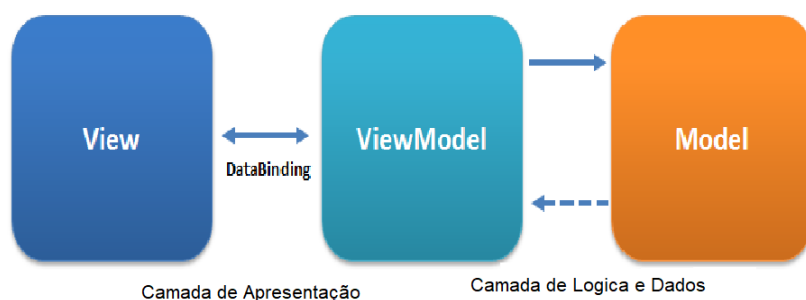


Figura 4 – Padrão MVVM

2.4.3 Angular

Angular é um poderoso *framework* que utiliza o padrão MVVM, pois oferece o recurso de *data binding* que possibilita a sincronização automática de dados entre *View-Model* e *View*. Ele tem o foco na criação *Single-Page Applications* (SPA), que é uma aplicação Web consumida em uma única página. O HTML e o *TypeScript*, que é um *superset* do *JavaScript* porém com tipagem estática, são utilizados para a criação da interface com o usuário, ou seja, o *front-end* para aplicações Web.

O *framework* Angular é voltado para camada de visualização, para a implementação de interfaces com o usuário. Por outro lado, para o *back-end* foi utilizado um ambiente de execução *JavaScript* chamado NodeJS (descrito a seguir). A comunicação entre estas partes ocorre por meio da REST API, que provê um conjunto restrições para transferir os dados em formato JSON (*JavaScript Object Notation*) no Angular. O Angular utiliza a biblioteca

Angular JWT 5.0.2, que é padrão de mercado (RFC-7519) que define como transmitir e armazenar objetos JSON de forma compacta e segura entre diferentes aplicações e possui mecanismo de injeção de independência embutida.

2.4.4 NodeJS

NodeJS é um ambiente de execução *JavaScript* no *server-side*, que é baseado no motor *JavaScript* V8 do navegador Chrome. Com ele, é possível criar aplicações *JavaScript* no lado do servidor. Ele é utilizado principalmente para a criação de aplicativos de rede altamente rápidas e escaláveis, pois consegue controlar um grande volume de conexões simultâneas com alto rendimento.

O NodeJS é uma tecnologia assíncrona que trabalha com uma única *thread* de execução, ou seja, é diferente das outras linguagens de programação como PHP, Java e C# que são *multi-thread*. Devido a esse fato, consegue lidar com um grande número de conexões simultâneas.

O NodeJS vem acompanhado do gerenciador de pacotes Node, o NPM (*Node Package Manager*), que fornece *plugins* e bibliotecas que podem ser integrados ao seu código. Para o *frameworks* de Mapeamento Objeto/Relacional foi utilizado a biblioteca TypeORM responsável pela persistência no banco de dados.

3 Especificação de Requisitos

Este capítulo tem como objetivo apresentar uma descrição do escopo do SCAP (Sistema de Controle de Afastamento de Professores), assim como o modelo de casos de uso e o diagrama de classes, conforme requisitos levantados por Duarte (2014) e posteriormente analisados por Prado (2015).

3.1 Descrição do Escopo

O SCAP tem como intuito auxiliar os professores do Departamento de Informática (DI) no gerenciamento das solicitações de afastamento para participações em eventos realizados no Brasil e no exterior. As solicitações de afastamento precisam ser avaliadas pelos professores do DI. Em alguns casos, devem passar também pela avaliação do Conselho do Centro Tecnológico (CT) e também pela Pró-Reitoria de Pesquisa e Pós Graduação (PRPPG). Após a aprovação em todos esses estágios, o professor é afastado temporariamente e autorizado a participar do evento.

Para um evento realizado no Brasil, a solicitação de afastamento precisa ser aprovada apenas pelo DI (representada pelos funcionários e os representantes discentes do departamento). Sendo assim, a solicitação deve ser enviada para a lista de e-mails da Câmara Departamental, endereçada ao Chefe de Departamento, cargo que é exercido temporariamente por um professor do DI. Se nenhum integrante da Câmara Departamental se manifestar contra o pedido, o afastamento é autorizado.

Para a solicitação de afastamento de um evento realizado no exterior, é escolhido um professor (sem parentesco com o solicitante) para se tornar relator do pedido, então o pedido deve passar por uma serie de avaliações para ser aprovado: primeiramente pelo CT, em seguida pela PRPPG e, após a aprovação, o pedido é publicado no Diário Oficial da União. Todavia, o SCAP não tem nenhuma integração com os processos do CT e da PRPPG, fazendo com que o controle das tramitações permaneça dentro do DI, restringindo assim o escopo do sistema.

Com o objetivo de auxiliar os professores e os secretários do DI na tramitação das solicitações de afastamento, tornando o processo mais rápido e simples, propôs-se o SCAP. Ele pretende alcançar esse objetivo por meio do envio automático de e-mails aos envolvidos e com uso de formulários para inserção e atualização de informações.

3.2 Casos de Uso

Na Tabela 1 estão listados os atores identificados por Duarte (2014) no levantamento de requisitos.

Tabela 1 – Atores do sistema SCAP (DUARTE, 2014).

Ator	Descrição
Professor	Professor efetivo do DI.
Chefe de Departamento	Professor eleito para um mandato temporário e realiza a função administrativa de chefe ou sub-chefe.
Secretário	Secretário do DI.

O secretário é responsável pela parte cadastral do sistema: ele é responsável por cadastrar professores e seus parentes e também de manter o histórico e registro de quem ocupa os cargos de Chefe e Sub-Chefe do departamento. Além disso, é responsável por cadastrar as decisões do CT e da PRPPG quando forem eventos internacionais e arquivar os processos de solicitação de afastamento já concluídos.

O professor é responsável por fazer a solicitação de afastamento e emitir um parecer, que pode ser favorável ou não, quando solicitado a ser Relator de algum pedido de afastamento internacional. Professores podem, também, manifestar-se contra um afastamento que encontra-se em aprovação.

O Chefe do Departamento tem a função de indicar um relator para cada afastamento internacional que for cadastrado no sistema, além de poder realizar todas as tarefas exercidas pelo professor.

Nas figuras 5 e 6 podemos observar os diagramas de casos de uso do SCAP. Ele foi dividido em dois subsistemas, sendo o primeiro responsável pela parte cadastral e o segundo responsável por gerenciar professores e chefes de departamento. A versão mais completa dos modelos podem ser observadas em (DUARTE, 2014; PRADO, 2015).

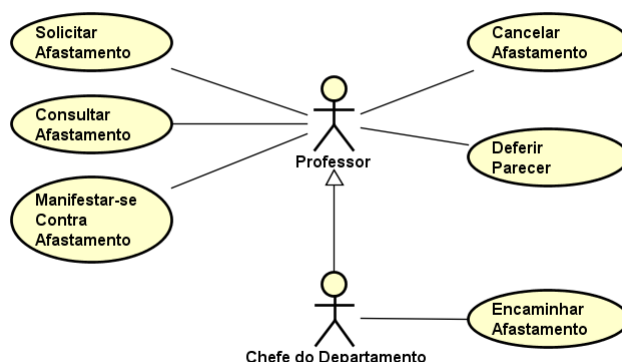


Figura 5 – Diagrama de Casos de Uso do subsistema Núcleo (PRADO, 2015).

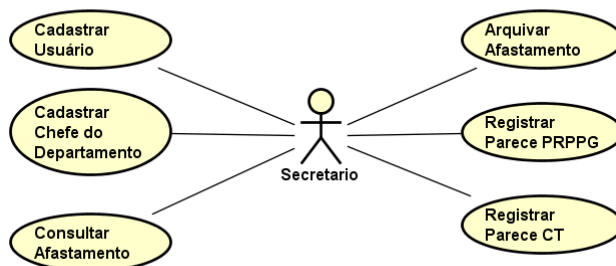


Figura 6 – Diagrama de Casos de Uso do subsistema Secretaria (PRADO, 2015).

Um Professor pode realizar um pedido de afastamento através do caso de uso **Solicitar Afastamento**, informando todos os dados necessários para realizar a solicitação. Ele pode também cancelar uma solicitação de afastamento utilizando o caso de uso **Cancelar Afastamento**.

Por meio do caso de uso **Consultar Afastamentos** um professor, chefe de departamento ou secretário, pode visualizar todas as informações sobre solicitações feitas por outros professores. Um professor pode também se **Manifestar-se Contra Afastamento**, onde ele preenche o motivo contra a manifestação. O Chefe de Departamento deve, no caso de evento internacional, **Encaminhar Afastamento** para um Relator. Este deve, quando solicitado pelo Chefe de Departamento, **Deferir Parecer**.

Por meio do caso de uso **Cadastrar Usuário**, um secretário pode realizar o cadastramento de um novo professor ou secretário, onde são informados todos os dados necessários. Um secretário pode também utilizar o caso de uso **Cadastrar Chefe do Departamento** para especificar o período do mandato do novo chefe do departamento.

Para solicitação de afastamento internacional, um secretário realiza o cadastro do parecer do Centro Tecnológico e da Pró-Reitoria de Pesquisa e Pós-Graduação através dos casos de uso **Registrar Parecer CT** e **Registrar Parecer PRPPG**.

O caso de uso **Arquivar Afastamento** é executado após a tramitação de uma solicitação de afastamento ser realizada, fazendo com que um secretário realize a alteração do status para “Arquivado”.

3.3 Análise do SCAP

O diagrama de classes SCAP mostrado na Figura 7 é uma adaptação do levantamento feito por Prado (2015), junto com o que havia antes sido levantado anteriormente por Duarte (2014).

Os professores são representados pela classe **Professor** e os secretários são representados pela classe **Secretário**, e as duas classes herdam os atributos da classe **Pessoa**,

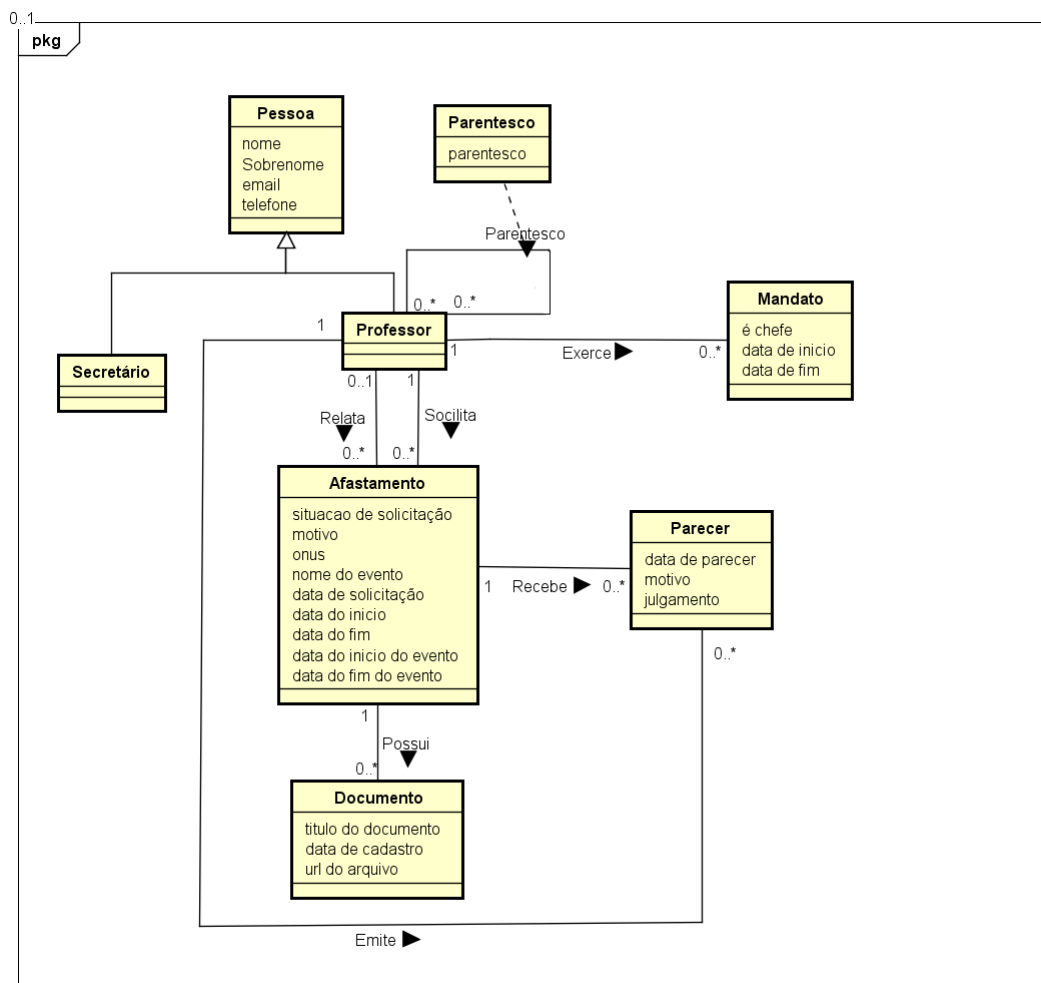


Figura 7 – Diagrama de Classes do SCAP

que contém as informações sobre os usuários do sistema.

A classe **Parentesco** guarda a relação de parentesco que existem entre os professores do Departamento. A classe **Mandato** contém as informações referente tempo de permanência no cargo chefe ou subchefe do departamento que os professores ocupam.

A classe **Afastamento** guarda informações sobre os pedidos de afastamento solicitados pelos professores. A classe **Afastamento** pode ou não possuir **Documentos** associados, mas possui um Relator para pedidos de afastamento em algum evento internacional.

O parecer criado pelo professor é representado pela classe **Parecer**, que é usada quando um professor efetua a emissão de um parecer com relação a um afastamento. Um professor pode se tornar relator e criar pareceres de vários afastamentos distintos.

O modelo de classes é complementado por restrições de integridade, identificadas por Duarte (2014) e complementadas por Prado (2015), a saber:

- O secretário do departamento não pode abrir uma solicitação de afastamento;
- Um professor não pode ser solicitado para dar um parecer sobre sua própria solicitação

de afastamento;

- Não pode haver mais de dois professores (chefe e subchefe de departamento) exercendo um mandato ao mesmo tempo;
- A data de início de um mandato de professor não pode ser posterior a data de fim do mesmo mandato;
- A data de início de um afastamento não pode ser posterior a data de fim do mesmo afastamento.

4 Projeto Arquitetural e Implementação

Na Tabela 2 abaixo é possível visualizar as diferentes implementações do SCAP que já foram elaboradas anteriormente por outros estudantes de graduação utilizando método FrameWeb, a tabela informa o *framework* MVC, a tecnologia ORM e a tecnologia DI utilizada.

Tabela 2 – Implementações anteriores do SCAP.

Trabalho	Framework MVC	Tecnologia ORM	Tecnologia DI
Duarte (2014)	JSF (Java)	Hibernate (Data Mapper)	CDI
Prado (2015)	VRaptor (Java)	Hibernate (Data Mapper)	CDI
Pinheiro (2017)	Laravel (PHP)	Eloquent (Active Record)	Não utilizou
Matos (2017)	Spring (Java)	Hibernate (Data Mapper)	Embutido no Spring
Matos (2017)	Vaadin (Java)	Hibernate (Data Mapper)	Não utilizou
Avelar (2018)	Ninja (Java)	Hibernate (Data Mapper)	Google Guice
Ferreira (2018)	Wicket (Java)	Hibernate (Data Mapper)	CDI
Ferreira (2018)	Tapestry (Java)	Hibernate (Data Mapper)	Embutido no Tapestry
Meirelles (2019)	CodeIgniter (PHP)	Embutido no CodeIgniter (Active Record)	Não utilizou
Meirelles (2019)	NodeJS (JavaScript)	Não utilizou	Não utilizou
Guterres (2019)	Play (Scala)	Slick (Padrão não identificado)	Não utilizou
Dalapicola (2021)	AdonisJS e Quasar (JavaScript)	Lucid (Active Record)	Embutido no AdonisJS
Berger (2021)	Yii (PHP)	Yii (Data Mapper)	Embutido no Yii
Berger (2021)	Symfony (PHP)	Doctrine (Active Record)	Embutido no Symfony
<i>Este</i>	Angular e NodeJS(JavaScript)	TypeORM (Data Mapper)	Embutido no Angular

O objetivo do trabalho em questão é verificar os resultados obtidos da aplicação e testar a eficácia do método do FrameWeb com os *frameworks* Angular e NodeJS, na implementação do SCAP não foi necessário implementar todas as suas funcionalidades. Assim, algumas restrições de integridade não foram implementadas, o sistema não possui a funcionalidade de gerar atas de reuniões, anexar documentos e enviar e-mails.

As seções que se seguem apresentam o processo de planejamento e desenvolvimento do SCAP. A Seção 4.1 apresenta as tecnologias utilizadas no projeto. Na Seção 4.2 é descrita a arquitetura utilizada no sistema. A Seção 4.3 apresenta os modelos gerados

por meio da aplicação do método `FrameWeb` e a Seção 4.4 exibe as capturas de telas do sistema SCAP que foi implementado.

4.1 Tecnologias Utilizadas

Na Tabela 3 são listadas as tecnologias utilizadas no desenvolvimento da ferramenta, bem como o propósito de sua utilização.

Tabela 3 – Plataforma de Desenvolvimento e Tecnologias Utilizadas

Tecnologia	Versão	Descrição	Propósito
Angular	13.1.4	Conjunto de implementações, em TypeScript, de padrões de projeto e componentes comumente utilizados no ambiente Web.	Redução da complexidade no desenvolvimento de aplicações Web a partir de seus componentes prontos para o uso.
TypeScript	4.4.3	Linguagem de programação tipada, que se baseia em JavaScript, adequada para o desenvolvimento Web.	Ampliar os recursos da linguagem de programação JavaScript.
Angular Material UI	13.1.3	Framework Web, de código-fonte aberto, que disponibiliza componentes de interface baseados em HTML, CSS e JavaScript	Reutilização de componentes visuais Web de alto nível.
Angular JWT	5.0.2	Padrão de mercado (RFC-7519) que define como transmitir e armazenar objetos JSON de forma compacta e segura entre diferentes aplicações.	Prover segurança na autenticação e troca de informações.
NodeJS	14.16.0	Plataforma de aplicação para desenvolvimento em JavaScript.	Criar API REST para ser consumida pelo <i>front-end</i> .
Express	4.17.2	<i>Framework</i> para aplicativo da Web do NodeJS	Facilitar a criação da API com métodos e utilitários HTTP.
PostgreSQL	14.0	Sistema Gerenciador de Banco de Dados Relacional gratuito.	Armazenamento dos dados manipulados pela ferramenta.
TypeORM	0.2.41	API para persistência de dados por meio de mapeamento objeto/-relacional.	Persistência dos objetos de domínio sem necessidade de escrita dos comandos SQL.
NPM	6.0	Gerenciador de Pacotes NodeJS.	Gerenciar e facilitar o download de pacotes NodeJS para utilização em aplicações baseadas em JavaScript.

Na Tabela 4 vemos os softwares que apoiaram o desenvolvimento de documentos e também do código fonte.

4.2 Arquitetura do Sistema

A estrutura principal utilizada pelo NodeJS é o MVC, porém essa estrutura foi modificada, pois a parte da Visualização (*View*) do MVC é de responsabilidade do *framework* Angular. Além dessa alteração foi utilizado o padrão de camadas de Serviço

Tabela 4 – Softwares de Apoio ao Desenvolvimento do Projeto

Tecnologia	Versão	Descrição	Propósito
Astah UML	8.0	Ferramenta de construção de modelos UML.	Criação dos modelos de Entidades, Aplicação, Persistência e Navegação.
TeXstudio	4.2.2	Editor de \LaTeX .	Escrita da documentação do sistema, sendo usado o <i>template abn-TeX</i> . ¹
PgAdmin	4.0	Plataforma gratuita de administração e desenvolvimento para PostgreSQL.	Facilitar o gerenciamento dos dados da aplicação.
Visual Studio Code	1.52	Ambiente de desenvolvimento (IDE) com suporte ao desenvolvimento para várias linguagens.	Implementação, implantação e testes da aplicação Web NodeJS.

e Repositório como implementado por Dalapicola (2021). essas camadas ajudam na modularização, teste do código e principalmente na escalabilidade.

Na Figura 8 podemos observar a representação do modelo arquitetural: os retângulos representam as camadas do sistema, as elipses são os *middlewares*, a camada de apresentação é de responsabilidade do Angular que se comunica através da REST API.

Nos *middlewares* de autenticação é verificado se o usuário foi identificado e na camada de autorização verifica se o usuário pode ter acesso àquele módulo. Após a autenticação, é chamada a camada de controle, que chama a camada de serviço que lida com a parte da lógica do sistema. A camada de serviço utiliza a camada de repositório, esta camada é responsável por persistir as informações no banco de dados, atualizar, recuperar e remover modelos obtidos pelo mapeamento objeto/relacional, que é feito através do *framework* TypeOrm. Foi utilizado o PostgreSQL como Sistema Gerenciador de Banco de Dados.

Este projeto utiliza o *framework* Angular para o *front-end* e o NodeJS para o *back-end*. A comunicação entre estas partes ocorre através da REST API que provê um conjunto restrições para transferir os dados em formato JSON.

4.2.1 Implementação do Back-end

Na implementação do *back-end* foi utilizado o NodeJS como *framework* principal, que é um ambiente de execução JavaScript *server-side*, com servidores Web altamente eficientes e escaláveis. Como podemos ver na Figura 9, a estrutura de diretórios do NodeJS é dividida em:

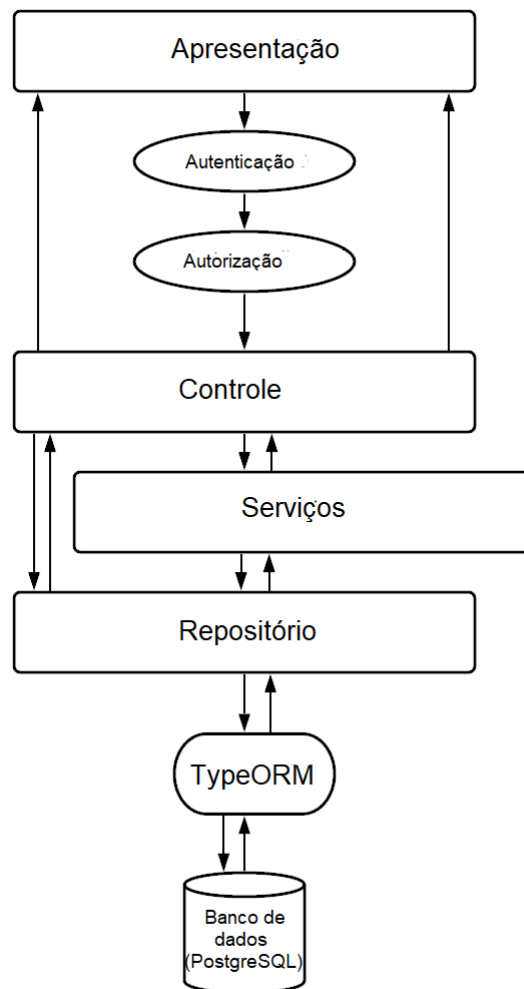


Figura 8 – Representação das camadas utilizando Angular e NodeJS.

- **src:** pasta principal da lógica de negócio, que contém os *Repositories*, *Controllers*, *Routes*, *Models* e *Services*:
 - **Routes:** tem a função de encaminhar as solicitações em URLs suportadas para as funções apropriadas do *Controllers*;
 - **Controllers:** os *controller* tem a função de receberem as requisições HTTP via REST API do cliente. São referentes aos *Controllers* do padrão MVC;
 - **Models:** contém os modelos, que é resultado do mapeamento Objeto/Relacional. Cada classe no modelo é referente a uma tabela no banco de dados;
 - **Repositories:** pasta que contém os arquivos referentes à camada de persistência do banco de dados, se comunica diretamente com os *Models*. O objetivo destes arquivos é prover dados para a aplicação;
 - **Services:** pasta que contém os arquivos referentes à parte da camada de aplicação. Esses arquivos utilizam as camadas de repositórios para prover uma funcionalidade para o sistema.

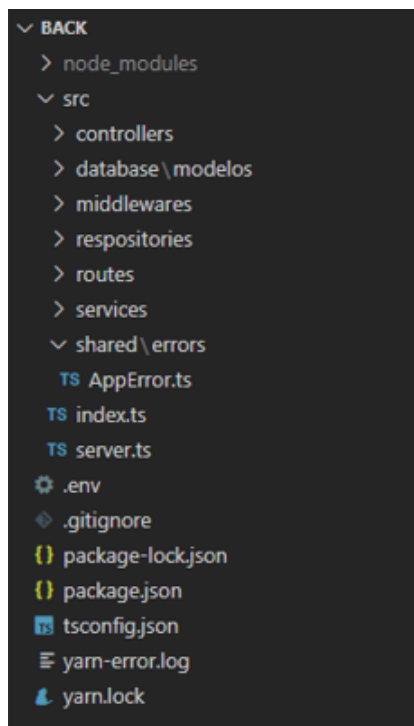


Figura 9 – Estrutura de diretórios do NodeJS.

- `node_modules`: pasta que contém as dependências dos plugins instalados pelo NPM;
- `server.js`: arquivo que é responsável pela inicialização do servidor;
- `env`: arquivo que contém as informações das variáveis de ambiente e configurações para comunicação com o banco de dados;
- `package.json`: arquivo JSON (textitJavaScript Object Notation) que exibe algumas informações do projeto, bem como formas de executá-lo e também lista as dependências necessária para execução do mesmo;
- `tsconfig.json`: arquivo JSON (textitJavaScript Object Notation) que especifica os arquivos e configurações de compilação necessárias para o projeto.

4.2.2 Implementação do Front-end

Na implementação do *front-end* foi utilizado o Angular, sendo responsável por apresentar a parte visual e interagir com o usuário, este *framework* utiliza o conceito de SPA (*Single Page Application*) e sua estrutura de diretórios, conforme ilustra a Figura 10, é dividida em:

- `src/app`: pasta principal, onde encontram-se as principais partes do código fonte do *front-end*:

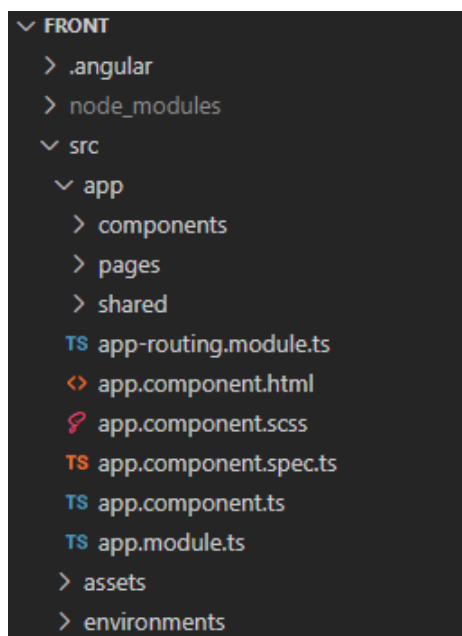


Figura 10 – Diretórios do Angular

- **pages**: esta pasta contém todos os componentes que formam as páginas do sistema. Geralmente um componente no *framework* Angular é composto por um arquivo HTML, *template* do componente, um arquivo TypeScript, contendo a lógica de programação por trás da página, um arquivo de testes e um arquivo CSS, responsável pela estilização;
- **share/services**: os serviços geralmente são usados no Angular para manter a lógica de negócio, compartilhando informações e funcionalidades entre os componentes. Sua principal função é realizar a interação do *front* com o *back* através de chamadas REST pelo protocolo HTTP;
- **app.component.html**: este arquivo é o componente raiz, pai de todos os outros componentes, ele chamará cada um de acordo com as solicitações de rotas;
- **app.module.ts**: este arquivo é o módulo geral do projeto, que importará todos os outros módulos, componentes e serviços para disponibilização e utilização dos mesmos;
- **app-routing.module.ts**: este arquivo conterá as configurações de rotas de navegação da aplicação;
- **/src/assets**: contém arquivos estáticos do sistema, geralmente as imagens;
- **tsconfig.json**: arquivo JSON (textitJavaScript Object Notation), que especifica os arquivos e configurações de compilação necessárias para o projeto;
- **package.json**: arquivo JSON (*JavaScript Object Notation*), que exhibe algumas informações do projeto, bem como formas de executá-lo e também lista as dependências necessárias para execução do mesmo.

4.3 Modelos FrameWeb

Nas seções a seguir são mostrados os modelos FrameWeb construídos durante o projeto arquitetural do SCAP.

4.3.1 Modelo de Entidades

Na Figura 11 é apresentado o Modelo de Entidades. As classes desse diagrama são as classes de domínio do projeto, complementando o modelo, alguns atributos são enumerações, como mostrado na Figura 12. Para a construção destes modelos, utilizou-se como base o diagrama de classes do SCAP (na Figura 7) e ajustou-se para a plataforma de implementação escolhida.

No modelo é possível observar o tipo referente a cada atributo, se ele pode ser inicializado com o valor nulo ou não, assim como alguns mapeamentos objeto/relacionais, como por exemplo: o valor `timestamp` indica que o atributo do tipo `Date`, pode armazenar as horas além da data, além disso, foi acrescentado um atributo do tipo `boolean` chamado `admin`.

4.3.2 Modelo de Persistência

A Figura 13 apresenta o Modelo de Persistência. Neste trabalho foi utilizado o padrão de projeto *Repository* que, semelhante ao conceito dos DAOs, prevê a existência de classes que possuem os métodos de persistência das informações no banco de dados. As classes *Repository* implementam os métodos das suas respectivas interfaces. Cada classe repositório é composta por um atributo chamado `repo`. Esse atributo é do tipo `Repository` que, por sua vez, é uma classe da biblioteca TypeORM. Ou seja, através do *generics* é possível identificar a qual classe de domínio o repositório fará referência. Assim, os métodos internos de comunicação com o banco de dados sempre serão de acordo com a classe modelo referenciada na declaração dessa variável.

Em `AfastamentoRepository`, o método `create` serve para criar um registro de um afastamento no banco de dados; o método `GetAll` resgata todos os afastamentos no banco de dados; o método `GetByld` com parâmetro `idProfessor` resgata todos os afastamentos associados ao id do professor; o método `GetByld` com parâmetro `idafastamento` resgata um único afastamento especificado pelo id; o método `Update` atualiza as informações do afastamento no banco de dados; o método `delete` exclui as informações do afastamento que foi especificado pelo id no banco de dados; `buscanomeEvento` realiza a busca de um afastamento pelo nome do evento passado; e `buscaNomeProfessor` retorna os afastamentos do professor especificado. As outras classes *repository* seguem esse mesmo padrão.

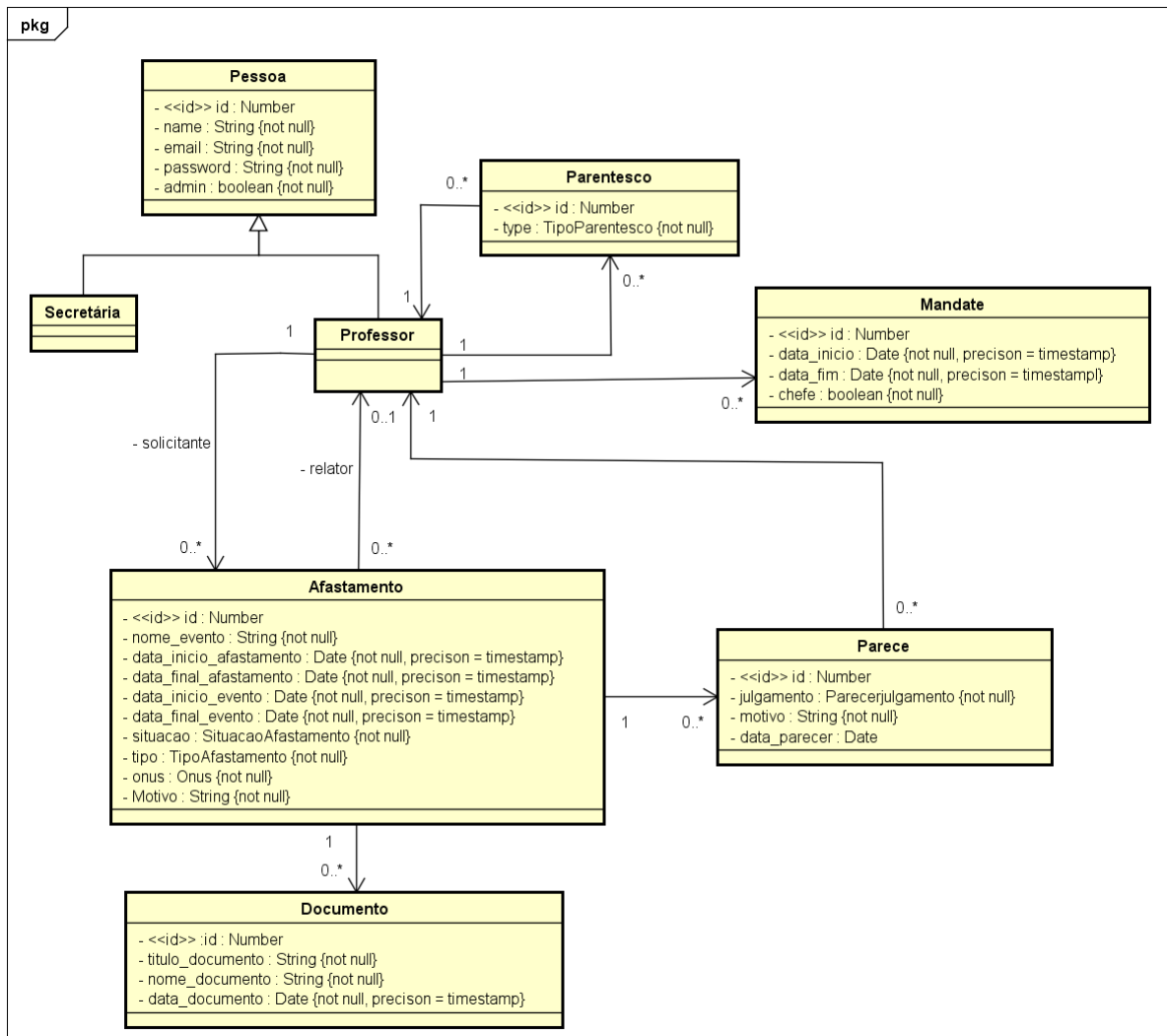


Figura 11 – Modelo de Entidades do SCAP.

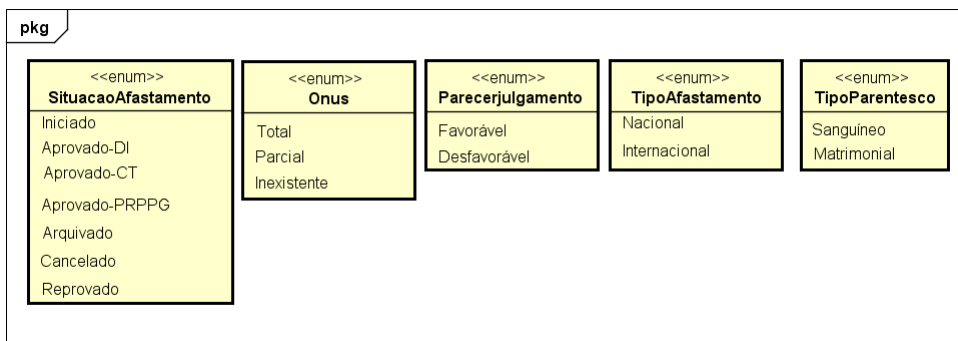


Figura 12 – Tipos Enumerados do SCAP.

4.3.3 Modelo de Aplicação

A Figura 14 mostra o Modelo de Aplicação do sistema. Nesse modelo, a camada de *controller* tem a função de receber as requisições que vem do usuário e enviar para a camada de serviços que utilizam as classes de repositório para persistir no banco de dados.

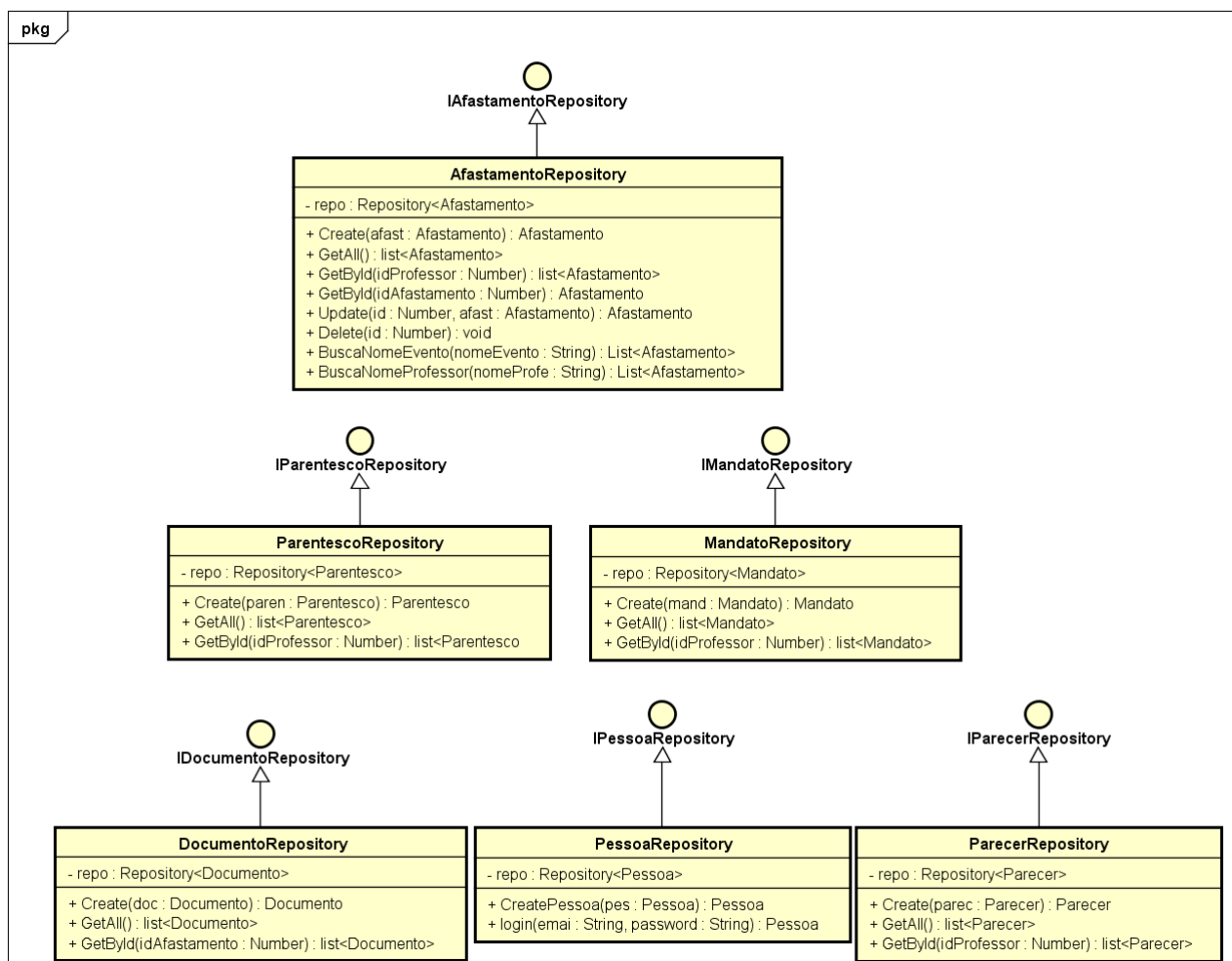


Figura 13 – Modelo de Persistência do SCAP

Nesse modelo foi necessário realizar uma adaptação em relação ao método `FrameWeb`, pois não foram utilizadas as interfaces entre os controladores e os serviços, assim como entre os serviços e as classes de repositório. Isso porque a linguagem utilizada, JavaScript, não tem suporte para criação de interfaces nativas. Com isso os métodos são chamados diretamente.

4.3.4 Modelo de Navegação

A Figura 15 apresenta o Modelo de Navegação para autenticação de *Login* pelos usuários do sistema. A página `Login` é composta por um formulário com `e-mail` e `password`. Quando os dados do usuário estão corretos, o método `login`, da classe `PessoaController`, redireciona para a página de `showhome` do SCAP. Porém, se houver alguma inconsistência nessa requisição, o sistema permanece na mesma página.

A Figura 16 apresenta o Modelo de Navegação para o cadastro de um de professor, a página `Professor` é composta por um formulário com os campos necessários para registrar um professor. Após realizar a requisição do método `create`, da classe `PessoaController`, o

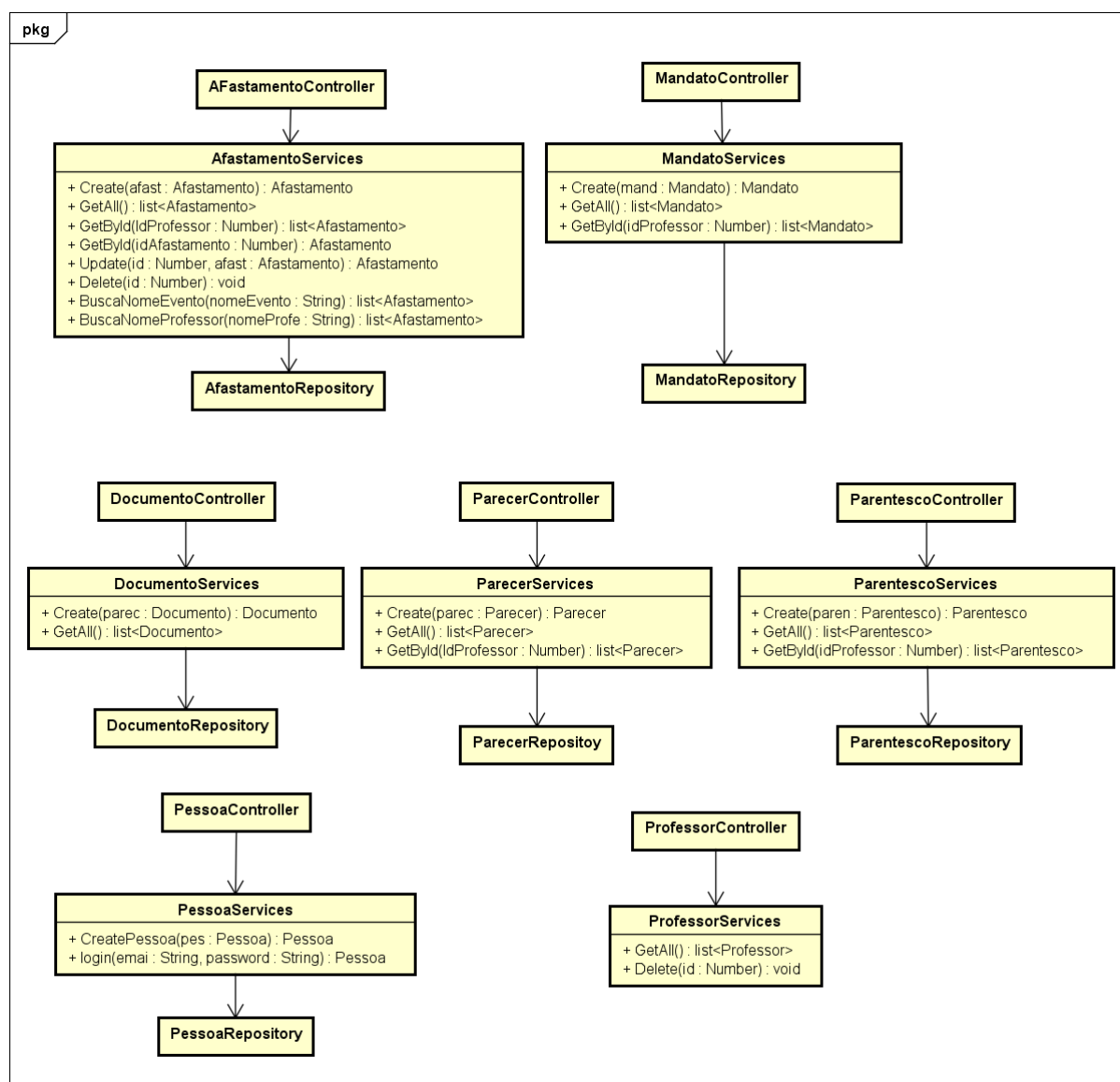


Figura 14 – Modelo de Aplicação do SCAP.

sistema redireciona para a página **showhome** em caso de sucesso ou permanece na mesma página, em caso de erro.

A Figura 17 apresenta o Modelo de Navegação para o cadastro de um afastamento, a página **Afastamento** é composta por um formulário com os campos necessários para registrar um afastamento. Após realizar a requisição do método **create**, da classe **AfastamentoController**, o sistema redireciona para a página **showhome** em caso de sucesso ou permanece na mesma página, em caso de erro.

A Figura 18 apresenta o Modelo de Navegação para buscar um afastamento pelo nome do evento, a página **listAfastamento** contém as listas dos afastamentos solicitados, porém para fazer um busca o usuário tem que digitar o nome do evento e clicar no botão para realizar a busca. Esse buton direciona para **AfastamentoController** que tem o método **BuscaNomeEvento**. Caso seja encontrado, é retornada uma lista de afastamentos com

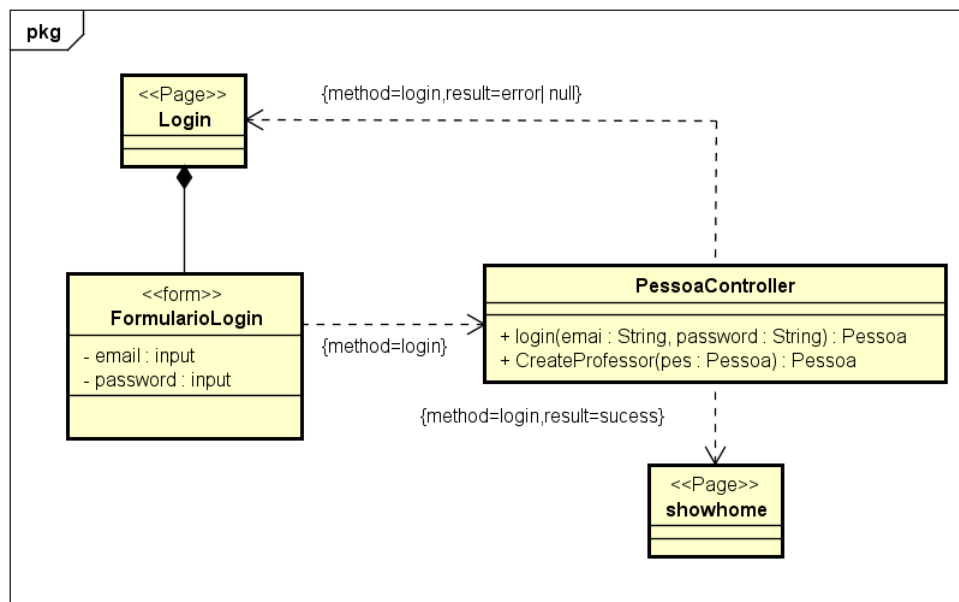


Figura 15 – Modelo de Navegação para autenticação do usuário

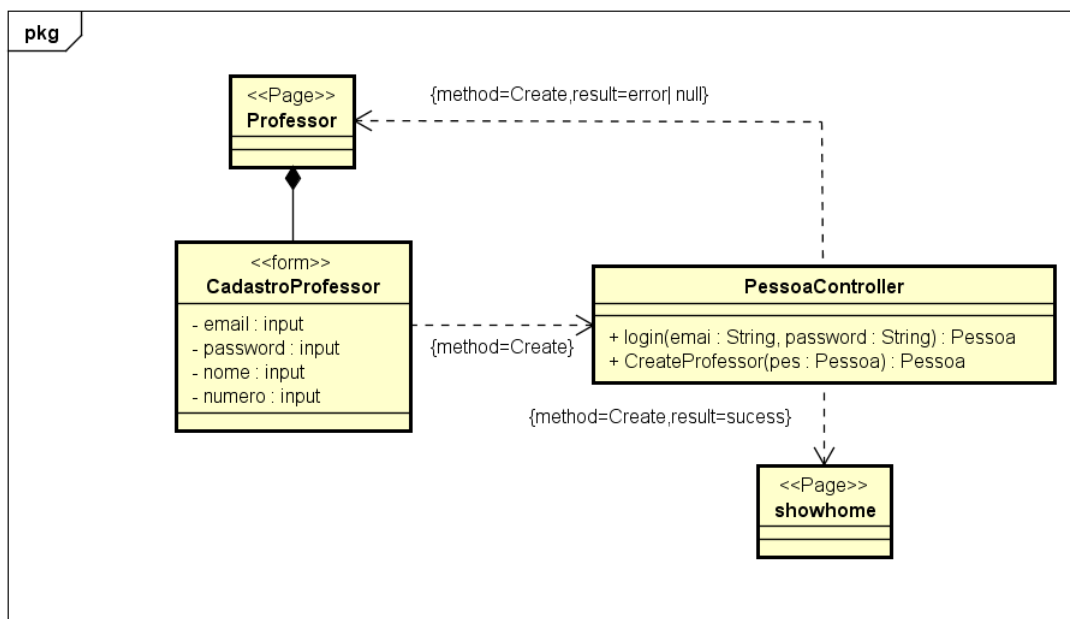


Figura 16 – Modelo de Navegação para cadastro de professor

aquele nome, caso contrário o retorno é nulo.

4.4 Apresentação de Resultados

Para ter acesso ao sistema do SCAP, o usuário deve fornecer o login e a senha que são cadastrados pelos secretários do sistema, o sistema possui controle de segurança utilizando a autenticação de usuários, não sendo possível acessar o conteúdo das páginas modificando a URL (*Uniform Resource Locator*). A Figura 19 mostra a tela de login.

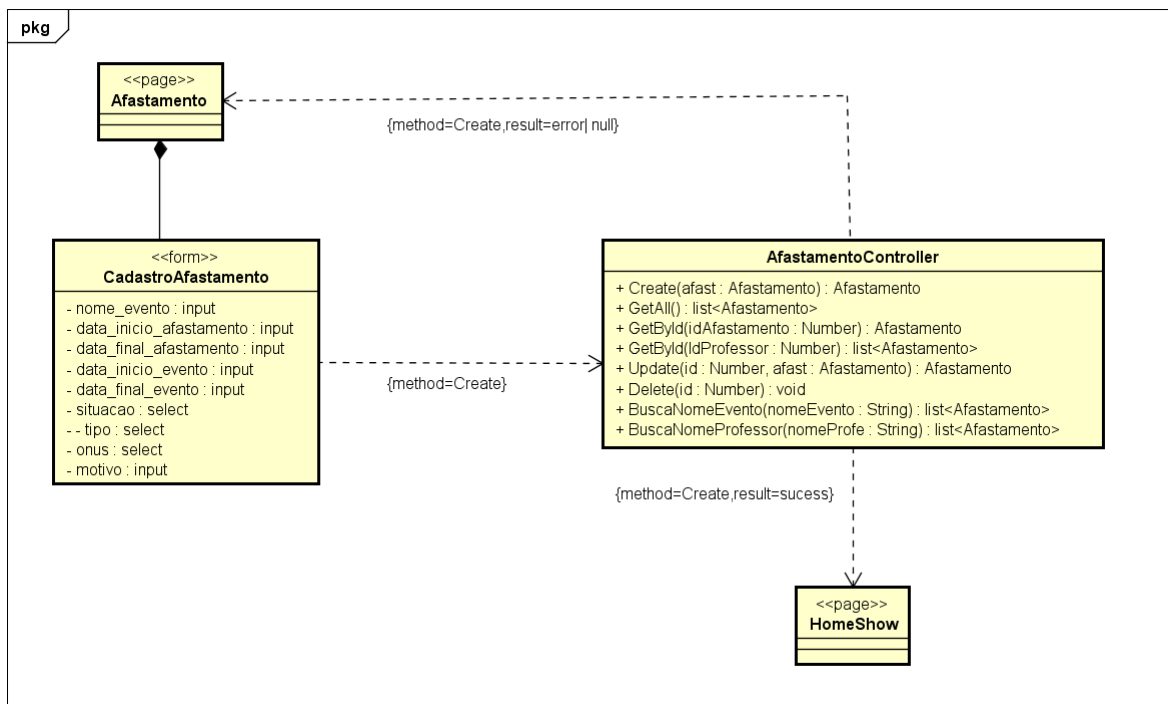


Figura 17 – Modelo de Navegação para solicitação de um afastamento

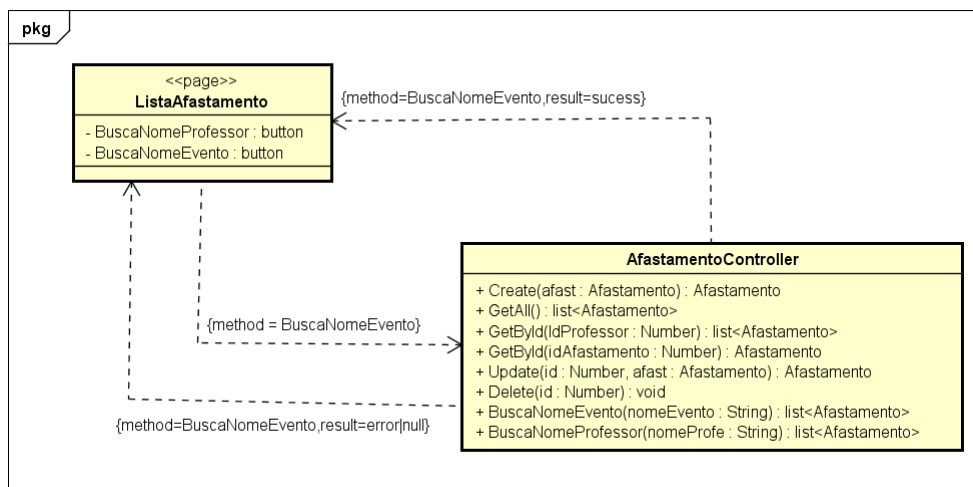


Figura 18 – Modelo de Navegação para buscar um afastamento

Cada usuário possui diferentes regras de permissões para visualizar a tela. Os secretários possuem a regra de permissão de administrador, pois são responsáveis pelo controle da maioria das funcionalidades. Já os professores tem permissão de usuário. A Figura 20 demonstra a tela inicial do usuário secretário. Na parte de baixo da página do sistema, estão localizados os botões que são responsáveis por realizarem as funcionalidades, como realizar os cadastramentos e as suas devidas associações.

Ao clicar no botão “Cadastro Professor”, o usuário secretário é redirecionado para uma página contendo um formulário onde é necessário passar as informações do professor para realizar o cadastramento. No botão “Cadastro Parentesco”, o secretário seleciona os



Figura 19 – Tela de Login do SCAP.



Figura 20 – Tela Inicial do Secretário.

professores disponíveis e informa se o parentesco é sanguíneo ou matrimonial. Na Figura 21 é mostrado o cadastramento de professor e na Figura 22 o cadastramento do parentesco.

No botão “Cadastro Mandato”, o usuário secretário pode realizar o cadastramento do mandato de um professor. O secretário será redirecionado para a página de cadastro, onde será necessário informar o período referente ao mandato, se é chefe ou não e seleciona um professor já cadastrado. Este cadastramento pode ser visualizado através da Figura 23

Os professores visualizam a tela inicial com as funcionalidades diferentes. A tela do usuário professor pode ser vista na Figura 24. Na tela inicial de um professor é possível

Figura 21 – Tela Cadastrar Professor.

Figura 22 – Tela Cadastrar Parentesco.

Figura 23 – Tela de Cadastramento De um Mandato

visualizar os afastamentos solicitados e os parentescos.

A Figura 25 demonstra a principal funcionalidade do sistema. Ao clicar no botão “Solicitar Afastamento”, o professor é redirecionado para um formulário onde precisa preencher todas as informações necessárias para realizar uma solicitação de afastamento, com isso o processo de solicitação é iniciado.

SCAP [Login](#) [Sair](#)

Olá, Professor(a): professor1 silva email:professor1@hotmail.com

Meus afastamentos

Nome	Data Inicio	Data Final	Nome	Situação	editar
evento4	Aug 23, 2022, 9:00:00 PM	Aug 8, 2022, 9:00:00 PM	professor1 silva	Iniciado	editar
evento 5	Aug 23, 2022, 9:00:00 PM	Aug 10, 2022, 9:00:00 PM	professor1 silva	Iniciado	editar

Meus Parentes

Professor	Tipo
professor2	Matrimonial

BEM-VINDO AO SCAP

Sistema de controle de afastamento de Professores foi criado com intuito de auxiliar o departamento no controle de solicitações de afastamentos de seus professores efetivo

[Home](#)
[Criar Afastamento](#)
[Deferir Parecer](#)
[Anexo Documento](#)
[Lista Afastamento](#)

Figura 24 – Tela Inicial do Professor no SCAP.

SCAP [Login](#) [Sair](#)

Solicitar Afastamento

Nome do Evento *
teste10

Data de Inicio Afastamento *
11/08/2022

Data Final de Afastamento *
18/08/2022

Data de Inicio Evento *
10/08/2022

Data Final de evento *
25/08/2022

Tipo Afastamento
Nacional

Onus

Motivo *
pq sim

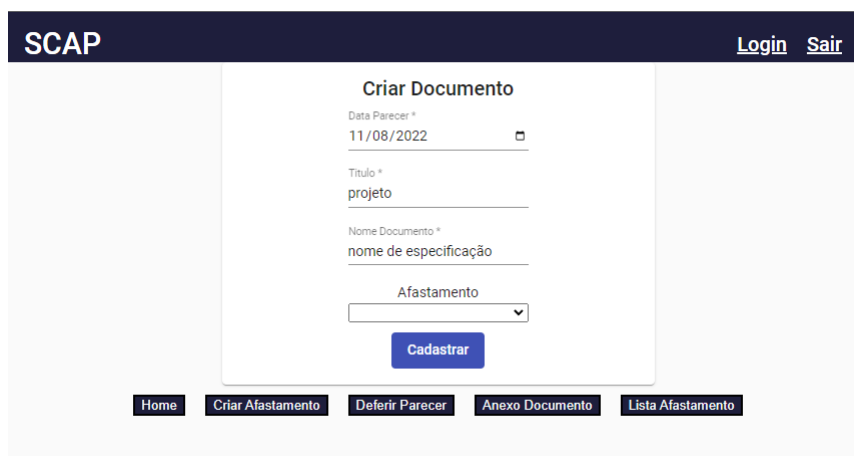
Cadastrar

[Home](#)
[Criar Afastamento](#)
[Deferir Parecer](#)
[Anexo Documento](#)
[Lista Afastamento](#)

Figura 25 – Tela de Solicitação de Afastamento.

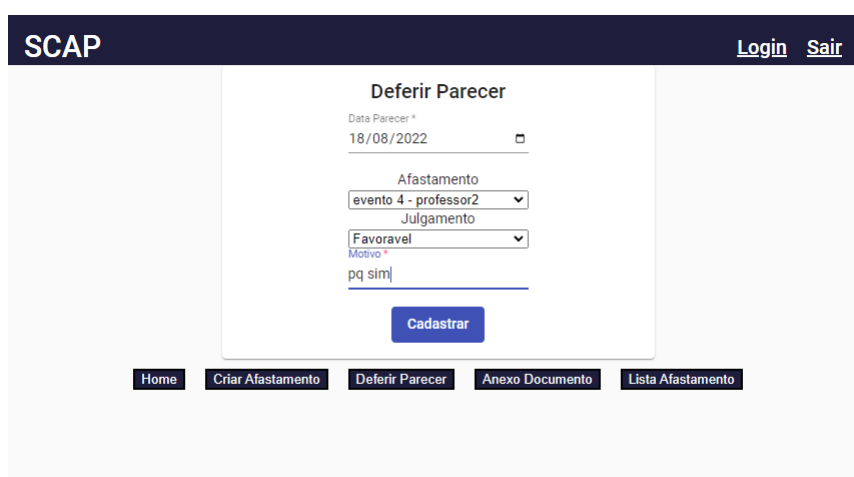
Através do botão “Anexo Documento” o professor pode cadastrar documentos para que eles sejam associados ao afastamento. É necessário informar título, nome e selecionar um afastamento, como mostrado na Figura 26.

O cadastro de um parecer é mostrado na Figura 27. Utilizando o botão “Parecer”, o professor que foi cadastrado pelo chefe do departamento como sendo relator de uma solicitação de afastamento internacional pode preencher as informações, justificando o motivo da sua decisão.



The screenshot shows the 'Criar Documento' (Create Document) form in the SCAP system. The form is contained within a white box on a dark blue background. At the top left of the page is the 'SCAP' logo, and at the top right are 'Login' and 'Sair' links. The form fields are: 'Data Parecer *' with the value '11/08/2022' and a calendar icon; 'Titulo *' with the value 'projeto'; 'Nome Documento *' with the value 'nome de especificação'; and 'Afastamento' with a dropdown menu. A blue 'Cadastrar' button is positioned below the form. At the bottom of the page, there is a navigation bar with buttons for 'Home', 'Criar Afastamento', 'Deferir Parecer', 'Anexo Documento', and 'Lista Afastamento'.

Figura 26 – Tela de Cadastramento de Documento.



The screenshot shows the 'Deferir Parecer' (Approve Opinion) form in the SCAP system. The form is contained within a white box on a dark blue background. At the top left of the page is the 'SCAP' logo, and at the top right are 'Login' and 'Sair' links. The form fields are: 'Data Parecer *' with the value '18/08/2022' and a calendar icon; 'Afastamento' with a dropdown menu showing 'evento 4 - professor2'; 'Julgamento' with a dropdown menu showing 'Favoravel'; and 'Motivo *' with the value 'pq sim'. A blue 'Cadastrar' button is positioned below the form. At the bottom of the page, there is a navigation bar with buttons for 'Home', 'Criar Afastamento', 'Deferir Parecer', 'Anexo Documento', and 'Lista Afastamento'.

Figura 27 – Tela para Deferir um Parecer.

Tanto o professor como o secretário podem realizar uma consulta no sistema para obterem informações sobre uma solicitação de afastamento. Os usuários podem realizar as buscas pelo nome do professor ou pelo nome do evento, na Figura 28 é possível visualizar uma lista com todos os afastamentos cadastrados no sistema, os secretários podem editar e excluir um afastamento, caso necessário. Os professores só podem alterar os seus afastamentos.

Por meio do link “ver mais” de um afastamento específico é possível visualizar as informações com maiores detalhes. A Figura 29 exhibe os detalhes do afastamento selecionado.

No final do processo de um afastamento, o secretário deve efetuar a atualização do afastamento em questão, fazendo a modificação no seu status, mudando o campo “Situação” de acordo com a etapa, como mostrado na Figura 30.

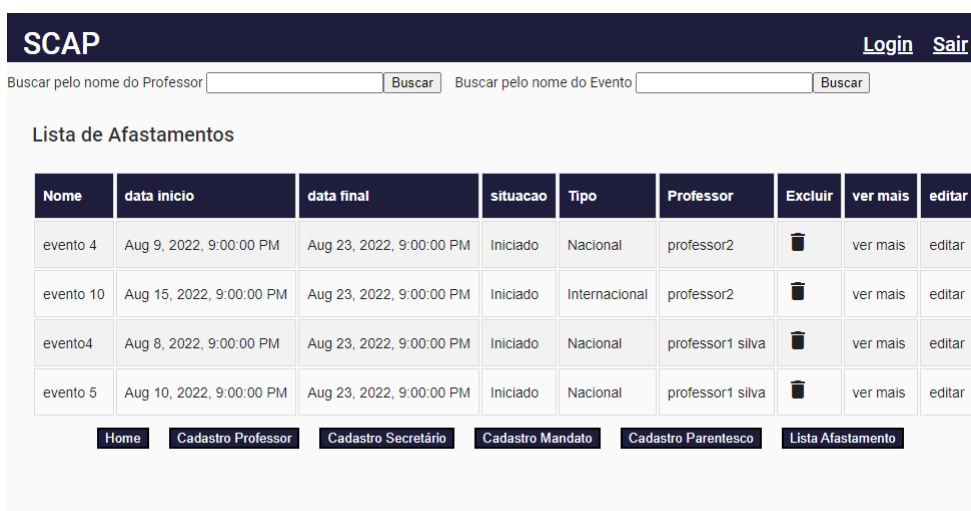


Figura 28 – Tela Lista de Afastamento

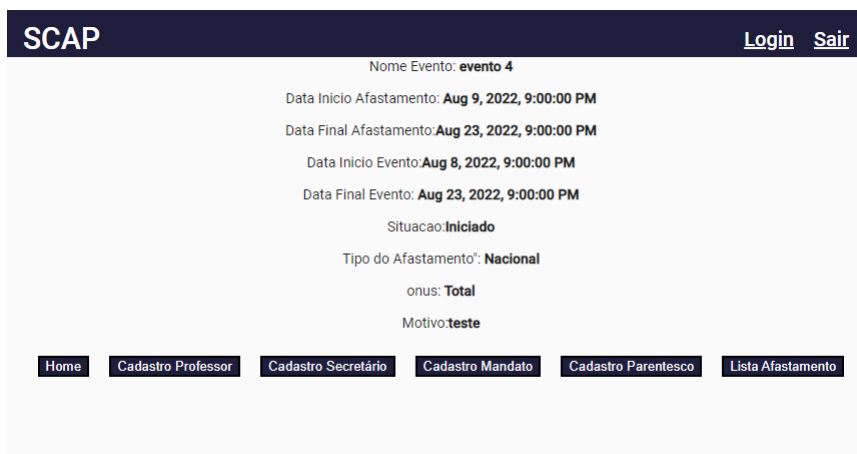


Figura 29 – Tela de Visualizar Afastamento.

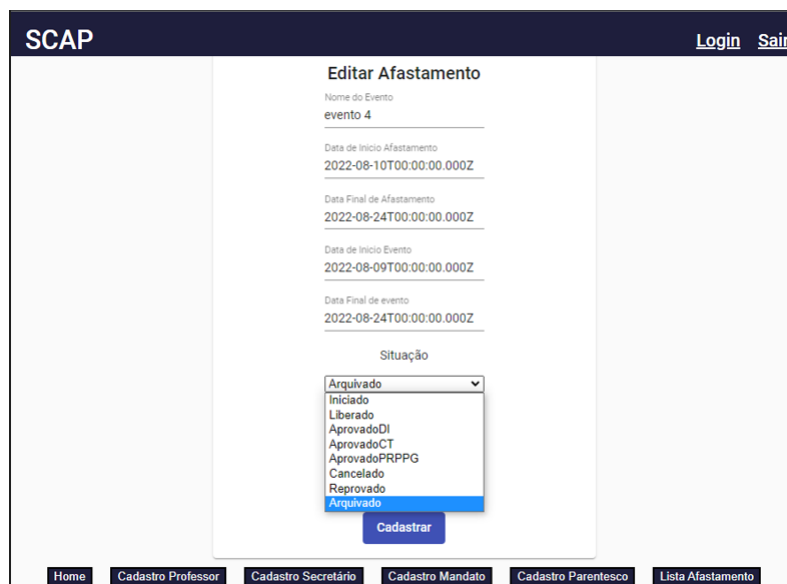


Figura 30 – Tela Editar Afastamento.

5 Conclusão

Este capítulo traz algumas considerações finais e elenca trabalhos futuros a partir do que foi realizado neste trabalho.

5.1 Considerações Finais

Os conhecimentos adquiridos ao longo do curso contribuíram para os resultados finais obtidos. Disciplinas como Banco de Dados, Engenharia de Software, Linguagens de Programação, Programação III e principalmente Desenvolvimento Web deram o conhecimento necessário e um excelente direcionamento do que deve ser feito. Na implementação do SCAP, foi necessário reunir e integrar os diversos conhecimentos adquiridos ao longo do curso num único projeto. Mesmo não implementando todas as funcionalidades, foi possível aplicar os conhecimentos adquiridos e principalmente adquirir novos conhecimentos. A implementação do SCAP não foi realizada com todas as funcionalidades, porém mesmo assim foi possível utilizar o método FrameWeb e gerar os modelos com algumas adaptações.

O uso dos *frameworks* foram de extrema importância na implementação, pois permitiram a utilização de componentes prontos, facilitando e agilizando o processo de desenvolvimento. Os modelos gerados pelo método FrameWeb facilitaram na etapa de implementação, pois proporcionaram um direcionamento do que deveria ser implementado e contribuíram significativamente para organizar e desenvolver. No entanto, os modelos gerados inicialmente foram alterados para se adequar nos *frameworks* utilizados.

Com relação às limitações do trabalho, como citado por Dalapicola (2021), os modelos de aplicação e persistência tiveram que ser adaptados em relação ao método FrameWeb, pois não foram utilizadas interfaces entre os controladores e os serviços, visto que o JavaScript não implementa interfaces de forma nativa. Além disso, os modelos de navegação foram adaptados para representar o padrão REST de comunicação entre *front-end* e *back-end*.

O método FrameWeb possui um editor gráfico, Editor FrameWeb (CAMPOS, 2017), porém não foi possível utilizá-lo devido a dificuldades de usabilidade encontradas no editor. Foi preferível, então, utilizar o Astah,¹, por ser mais simples e também dar suporte à linguagem UML(*Unified Modeling Language*).

¹ <<https://astah.net/>>

5.2 Trabalhos Futuros

Como sugestões para trabalho futuros que envolvam o método FrameWeb e seu conjunto de ferramentas de apoio, são sugeridas algumas mudanças, são elas:

- Implementar o sistema SCAP em algum *framework* que utiliza a linguagem Python, pois ainda não foi implementado nessa linguagem programação, isso poderia contribuir para o enriquecimento do método FrameWeb;
- Fazer um estudo comparativo das diversas implementações que já foram realizadas usando o método FrameWeb e encontrar os prós e contras em comum para cada linguagem de programação e depois resumir, ou seja, vai contribuir para dar uma visão geral sobre o método em cada linguagem de programação;
- Ampliar o escopo do método FrameWeb para dar suporte as funcionalidades dos *frameworks* modernos, por exemplo, suporte a arquitetura REST.

Referências

- ALUR, D.; CRUPI, J.; MALKS, D. *Core J2EE patterns: best practices and design strategies*. [S.l.]: Prentice Hall Professional, 2003. Citado 2 vezes nas páginas 20 e 22.
- AVELAR, R. A. dos. *Aplicação do método FrameWeb no desenvolvimento de um sistema de informação utilizando o framework Ninja*. [S.l.], 2018. Vitória, ES, Brasil. Citado 2 vezes nas páginas 12 e 30.
- BEDER, D. M. *Engenharia Web : uma abordagem sistemática para o desenvolvimento de aplicações web*. [S.l.: s.n.], 2012. 215 p. ISBN 978-85-7600-290-1. Citado na página 17.
- BERGER, V. *Aplicação do método FrameWeb no desenvolvimento de um sistema de informação utilizando os frameworks Yii e Symfony*. [S.l.], 2021. Vitória, ES, Brasil. Citado 2 vezes nas páginas 13 e 30.
- CAMPOS, S. L. *FrameWeb Editor: Uma Ferramenta CASE para suporte ao Método FrameWeb*. Vitória, ES, Brasil, 2017. Citado na página 47.
- DALAPICOLA, E. G. *Aplicação do método FrameWeb no desenvolvimento do sistema SCAP utilizando os frameworks Quasar e AdonisJS*. [S.l.], 2021. Vitória, ES, Brasil. Citado 5 vezes nas páginas 13, 16, 30, 32 e 47.
- DUARTE, B. B. *Aplicação do Método FrameWeb no Desenvolvimento de um Sistema de Informação na Plataforma Java EE 7*. [S.l.], 2014. Vitória, ES, Brasil. Citado 9 vezes nas páginas 8, 12, 13, 14, 25, 26, 27, 28 e 30.
- FALBO, R. d. A. *Engenharia de Requisitos*. [s.n.], 2017. 178 p. Disponível em: <http://www.inf.ufes.br/~vitorsouza/falbo/Notas_Aula_Engenharia_Requisitos_Falbo_2017.pdf>. Citado na página 16.
- FALBO, R. de A. *Projeto de Sistema de Software*. 2018. Notas de Aula. Disponível em: <http://www.inf.ufes.br/~jssalamon/wp-content/uploads/disciplinas/projsistsoft/Notas_Aula_Projeto_Sistemas_2018.pdf>. Citado na página 16.
- FERREIRA, M. T. *Aplicação do método FrameWeb no desenvolvimento do sistema SCAP utilizando os frameworks Wicket e Tapestry*. [S.l.], 2018. Vitória, ES, Brasil. Citado 2 vezes nas páginas 12 e 30.
- FOWLER, M. *Patterns of Enterprise Application Architecture*. [S.l.]: Addison-Wesley Professional, 2002. 560 p. ISBN 0-321-12742-0. Citado 4 vezes nas páginas 7, 19, 21 e 23.
- GAMMA, E. et al. *Padrões de Projeto: soluções reutilizáveis de software orientado a objetos*. 1. ed. Porto Alegre: Bookman, 2000. v. 1. Tradução Luiz A. Meirelles Salgado. ISBN 978-85-7780-046-9. Citado na página 21.
- GUTERRES, C. S. *Aplicação do método FrameWeb no desenvolvimento de um sistema de informação usando o framework Play*. [S.l.], 2019. Vitória, ES, Brasil. Citado 2 vezes nas páginas 13 e 30.

- MATOS, R. P. de. *Aplicação do método FrameWeb no desenvolvimento de um sistema de informação usando os frameworks Spring MVC e Vaadin*. [S.l.], 2017. Vitória, ES, Brasil. Citado 2 vezes nas páginas 12 e 30.
- MEIRELLES, L. *Aplicação do método FrameWeb no desenvolvimento de um sistema de informação utilizando os frameworks Codeigniter e NodeJS*. [S.l.], 2019. Vitória, ES, Brasil. Citado na página 30.
- PINHEIRO, F. G. *Aplicação do método FrameWeb no desenvolvimento de um sistema de informação utilizando um framework PHP e um framework JavaScript*. Vitória, ES, Brasil, 2017. Citado na página 30.
- PRADO, R. C. d. *Aplicação do método FrameWeb no desenvolvimento de um sistema de informação utilizando o framework VRaptor 4*. [S.l.], 2015. Vitória, ES, Brasil. Citado 9 vezes nas páginas 7, 12, 13, 14, 25, 26, 27, 28 e 30.
- PRESSMAN, R. S. *Engenharia de Software - Uma Abordagem Profissional 7.ed.* [S.l.]: McGraw-Hill, 2011. ISBN 9788563308337. Citado 4 vezes nas páginas 7, 12, 15 e 22.
- PRESSMAN, R. S.; LOWE, D. *Engenharia Web. Um Enfoque Profissional - 1ª edição*. [S.l.]: LTC, 2009. ISBN 8521616961. Citado 2 vezes nas páginas 17 e 18.
- SMITH, J. Wpf apps with the model-view-viewmodel design pattern. *MSDN Magazine*. Disponível em: <http://msdn.microsoft.com/enus/magazine/dd419663.aspx>, v. 9, 2009. Citado na página 23.
- SOMMERVILLE, I. *Engenharia de Software*. 9. ed. [S.l.]: Pearson Universidades, 2011. 568 p. ISBN 978-85-7936-108-1. Citado 3 vezes nas páginas 15, 16 e 17.
- SOUZA, V. E. S. *FrameWeb: um Método baseado em Frameworks para o Projeto de Sistemas de Informação Web*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2007. Citado 7 vezes nas páginas 7, 12, 13, 18, 20, 21 e 22.
- SOUZA, V. E. S. The FrameWeb Approach to Web Engineering: Past, Present and Future. In: ALMEIDA, J. P. A.; GUIZZARDI, G. (Ed.). *Engineering Ontologies and Ontologies for Engineering*. 1. ed. Vitória, ES, Brazil: NEMO, 2020. cap. 8, p. 100–124. ISBN 9781393963035. Disponível em: <<http://purl.org/nemo/celebratingfalbo>>. Citado 3 vezes nas páginas 12, 13 e 18.
- TANENBAUM. *Redes de Computadores - 5ª Ed.* [S.l.]: Pearson Education, 2011. ISBN 9788576059240. Citado na página 12.

Apêndices



Documento de Projeto de Sistema

SCAP

Vitória, ES

2022

Registro de Alterações:

Versão	Responsável	Data	Alterações
1.0	Danilo Felicio	15/04/2022	Escrita da introdução
1.1	Danilo Felicio	20/04/2022	Escrita das secções 2,3
1.2	Danilo Felicio	29/05/2022	Escrita das secções 2,3 e os modelos Frameweb
1.3	Danilo Felicio	02/05/2022	Ajustes nas secções 2,3,4,5
1.5	Danilo Felicio	12/08/2022	correções nas secções 2,3,4,5
1.6	Danilo Felicio	22/08/2022	correções nas secções 1,5

1 Introdução

Este documento apresenta o projeto (*design*) do sistema SCAP. O objetivo desse trabalho é apresentar o Sistema de Controle de Afastamento de Professores visando simplificar e agilizar o processo de pedidos de afastamento.

Além desta introdução, este documento está organizado da seguinte forma: a Seção 2 apresenta a plataforma de software utilizada na implementação do sistema; a Seção 3 apresenta a especificação dos requisitos não funcionais (atributos de qualidade), definindo as táticas e o tratamento a serem dados aos atributos de qualidade considerados condutores da arquitetura; a Seção 4 apresenta a arquitetura de software; por fim, a Seção 5 apresenta os modelos FrameWeb que descrevem os componentes da arquitetura.

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
Angular	13.1.4	Essa plataforma é voltada para a construção da interface de aplicações SPA (<i>Single Page Application</i>), usando HTML, CSS e, principalmente, JavaScript.	Redução da complexidade no desenvolvimento de aplicações Web a partir de seus componentes prontos para o uso.
TypeScript	4.4.3	Linguagem de programação estaticamente tipada, que se baseia em JavaScript, adequada para o desenvolvimento Web.	Ampliar os recursos da linguagem de programação JavaScript.
Angular Material UI	13.1.3	<i>Framework</i> Web, de código-fonte aberto, que disponibiliza componentes de interface baseados em HTML, CSS e JavaScript	Reutilização de componentes visuais Web de alto nível.
Angular JWT	5.0.2	Padrão de mercado (RFC-7519) que define como transmitir e armazenar objetos JSON de forma compacta e segura entre diferentes aplicações.	Prover segurança na autenticação e troca de informações.
NPM	8.1.2	Gerenciador de Pacotes NodeJS e Angular.	Gerencia e facilita o download de pacotes NodeJS e Angular para utilização em aplicações baseadas em JavaScript.

Tecnologia	Versão	Descrição	Propósito
NodeJS	14.16.0	Plataforma de aplicação para desenvolvimento <i>back-end</i> em JavaScript.	Criar API REST para ser consumida pelo <i>front-end</i> .
Express	4.17.2	<i>Framework</i> para aplicativo da Web do NodeJS	Facilitar a criação da API com métodos e utilitários HTTP.
PostgreSQL	14	Sistema Gerenciador de Banco de Dados Relacional gratuito.	Armazenamento dos dados manipulados pela ferramenta.
TypeORM	0.2.41	API para persistência de dados por meio de mapeamento objeto/-relacional.	Persistência dos objetos de domínio sem necessidade de escrita dos comandos SQL.

Na Tabela 2 vemos os softwares que apoiaram o desenvolvimento de documentos e também do código fonte.

Tabela 2 – Softwares de Apoio ao Desenvolvimento do Projeto

Tecnologia	Versão	Descrição	Propósito
TeXstudio	4.2.2	Editor de \LaTeX .	Escrita da documentação do sistema, sendo usado o <i>template abnTeX</i> . ¹
Astah UML	8.0	Ferramenta de construção de modelos UML.	Criação dos modelos de Entidades, Aplicação, Persistência e Navegação.
PgAdmin	4.0	Plataforma gratuita de administração e desenvolvimento para PostgreSQL.	Facilitar o gerenciamento dos dados da aplicação.
Visual Studio Code	1.52	Ambiente de desenvolvimento (IDE).	Implementação, implantação e testes da aplicação Web em Angular e da API em NodeJS.

3 Atributos de Qualidade e Táticas

Na Tabela 3 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.

¹ <<http://www.abntex.net.br>>.

Tabela 3 – Atributos de Qualidade e Táticas Utilizadas

Categoria	Requisitos Não Funcionais	Condutor da Arquitetura	Tática
Facilidade de Aprendizado, Operabilidade	RNF02, RNF03	Sim	<ul style="list-style-type: none"> • Prover ao usuário uma interface de fácil compreensão, que permite entender os conceitos intuitivamente. Então, devem ser fornecidos diálogos simples para garantir que nenhuma dúvida seja gerada no momento da inserção dos dados.
Segurança de Acesso	RNF01	Sim	<ul style="list-style-type: none"> • Somente pessoas autorizadas devem acessar os dados, o usuário deve ser autenticado através do e-mail e da senha.
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. • Separar a interface com o usuário do restante da aplicação.
Reusabilidade	RNF07	Não	<ul style="list-style-type: none"> • Reutilizar componentes e frameworks existentes. • Desenvolver novos componentes para reuso, quando não houver componentes disponíveis e houver potencial para reuso.

4 Arquitetura de Software

A Figura 1 mostra a arquitetura do sistema SCAP, que está dividida em três camadas: *Presentation Tier*, *Business Tier* e *Data Access Tier*. Esta arquitetura é uma adaptação da arquitetura sugerida pelo FrameWeb (SOUZA, 2020), baseada no padrão Camada de Serviço (FOWLER, 2002). Este projeto utiliza o *framework* Angular para o *front-end* e o NodeJS para o *back-end*. A comunicação entre estas partes ocorre através da REST API que provê um conjunto de restrições para transferir os dados em formato JSON.

A *Presentation Tier* é composta pelos pacotes **View** e **Control**. O pacote **View** trata da apresentação das páginas Web (formulários HTML), utilizando a biblioteca Angular e o Material UI. O pacote **Control** detém as classes de ação do *front-end*, implementadas em Angular.

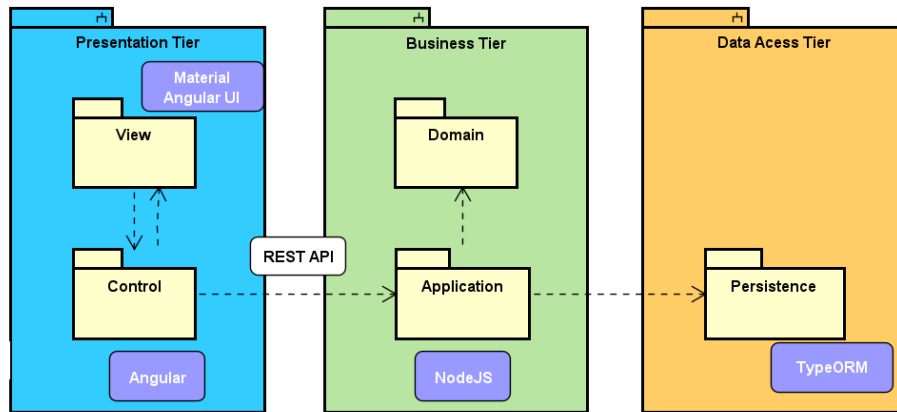


Figura 1 – Arquitetura de Software.

A *Business Tier* contém os pacotes *Domain* e *Application*. O primeiro pacote agrupa as classes de domínio do projeto, enquanto o segundo refere-se às classes de serviço, onde estão implementadas as regras de negócio, utilizando o NodeJS.

Por fim, a *Data Access Tier* contém o pacote *Persistence* que é responsável pela persistência (no banco de dados PostgreSQL) das instâncias das classes de domínio. Para este fim, é utilizado a biblioteca TypeORM.

5 Modelagem FrameWeb

SCAP é um sistema Web cuja arquitetura utiliza *frameworks* comuns no desenvolvimento para esta plataforma. Desta forma, o sistema pode ser modelado utilizando a abordagem FrameWeb (SOUZA, 2020), utilizando a arquitetura apresentada anteriormente.

A Tabela 4 indica os *frameworks* presentes na arquitetura do sistema que se encaixam em cada uma das categorias de *frameworks* que o FrameWeb dá suporte. Em seguida, os modelos FrameWeb são apresentados para cada camada da arquitetura.

Tabela 4 – *Frameworks* da arquitetura do sistema separados por categoria.

Categoria de <i>Framework</i>	<i>Framework</i> Utilizado
Controlador Frontal	Angular
Injeção de Dependências	Angular
Mapeamento Objeto/Relacional	TypeORM
Segurança	JWT

5.1 Camada de Negócio

Na Figura 2 é apresentado o Modelo de Entidades. As classes desse diagrama são as classes de domínio do projeto, complementando o modelo, alguns atributos são enumerações, como mostrado na Figura 3.

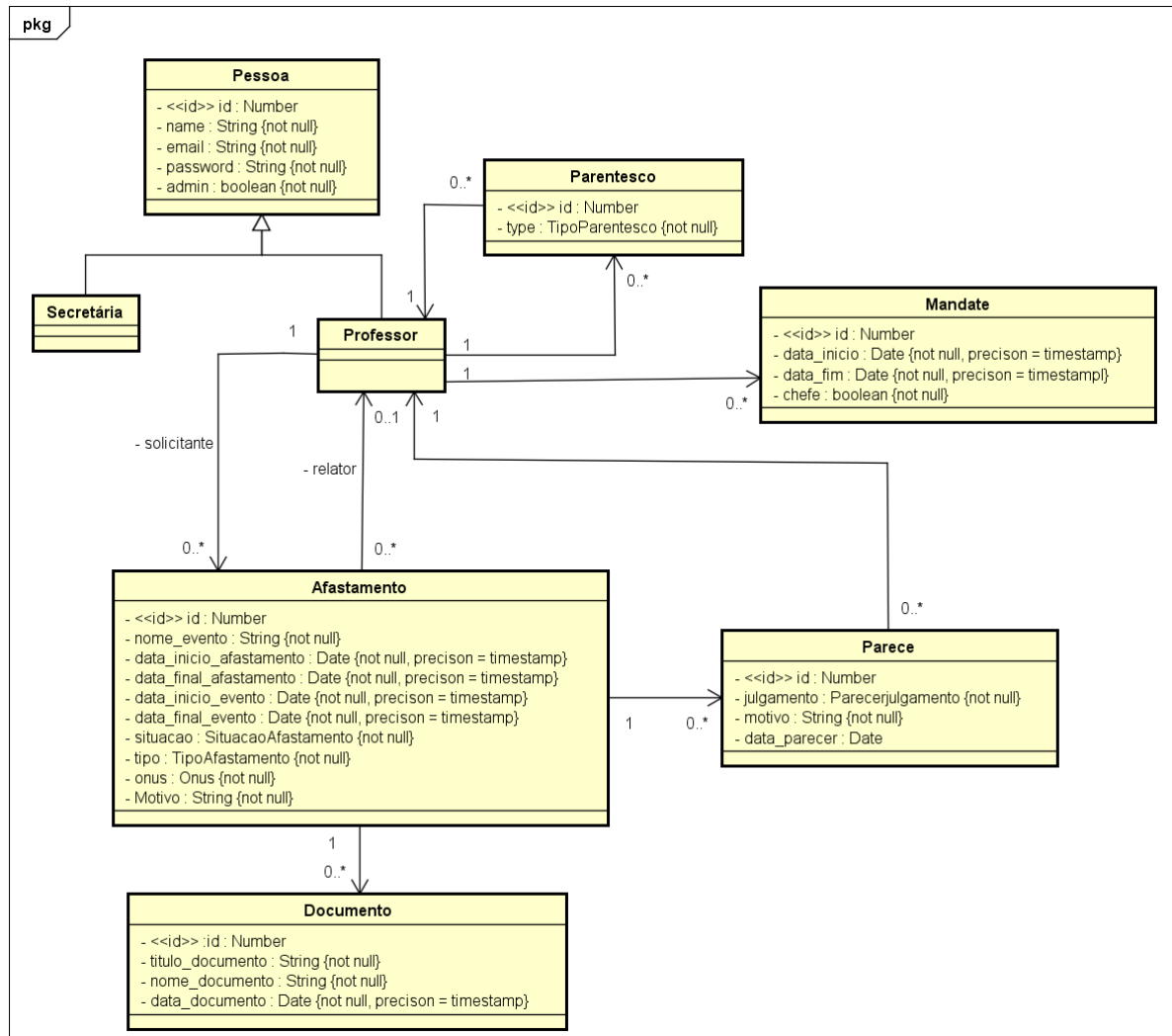


Figura 2 – Modelo de Entidades do SCAP.

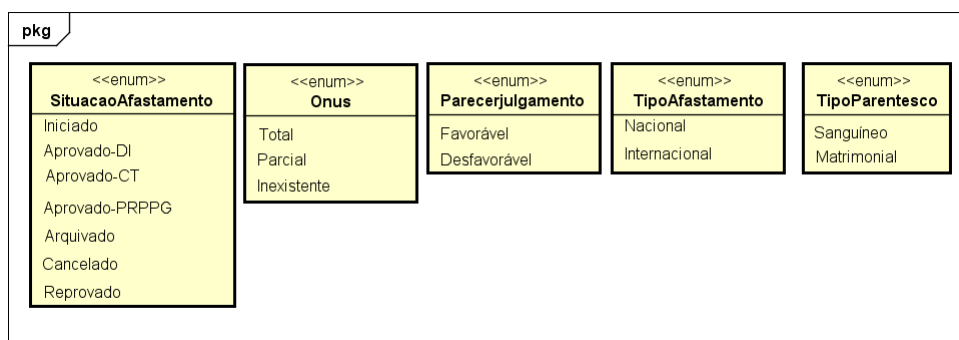


Figura 3 – Tipos Enumerados do SCAP.

A Figura 4 mostra o Modelo de Aplicação do sistema. Nesse modelo, a camada de *controller* tem a função de receber as requisições que vem do usuário e enviar para a camada de serviços que utilizam as classes de repositório para persistir no banco de dados.

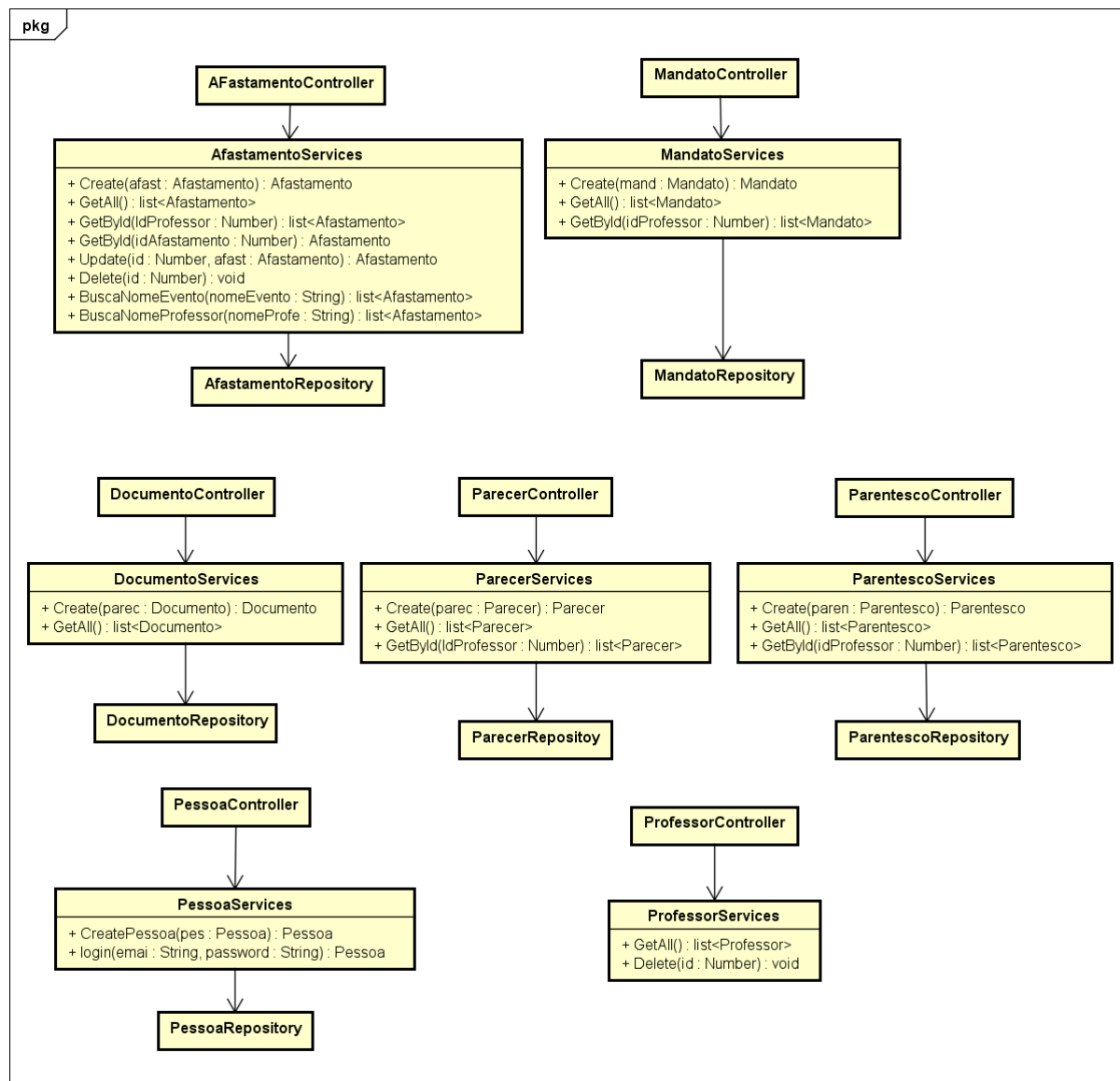


Figura 4 – Modelo de Aplicação do SCAP.

5.2 Camada de Acesso a Dados

A Figura 5 apresenta o Modelo de Persistência. Neste trabalho foi utilizado o padrão de projeto *Repository* que, semelhante ao conceito dos DAOs, prevê a existência de classes que possuem os métodos de persistência das informações no banco de dados. As classes *Repository* implementam os métodos das suas respectivas interfaces. Cada classe repositório é composta por um atributo chamado *repo*. Esse atributo é do tipo *Repository* que, por sua vez, é uma classe da biblioteca TypeORM. Ou seja, através do *generics*

é possível identificar a qual classe de domínio o repositório fará referência. Assim, os métodos internos de comunicação com o banco de dados sempre serão de acordo com a classe modelo referenciada na declaração dessa variável.

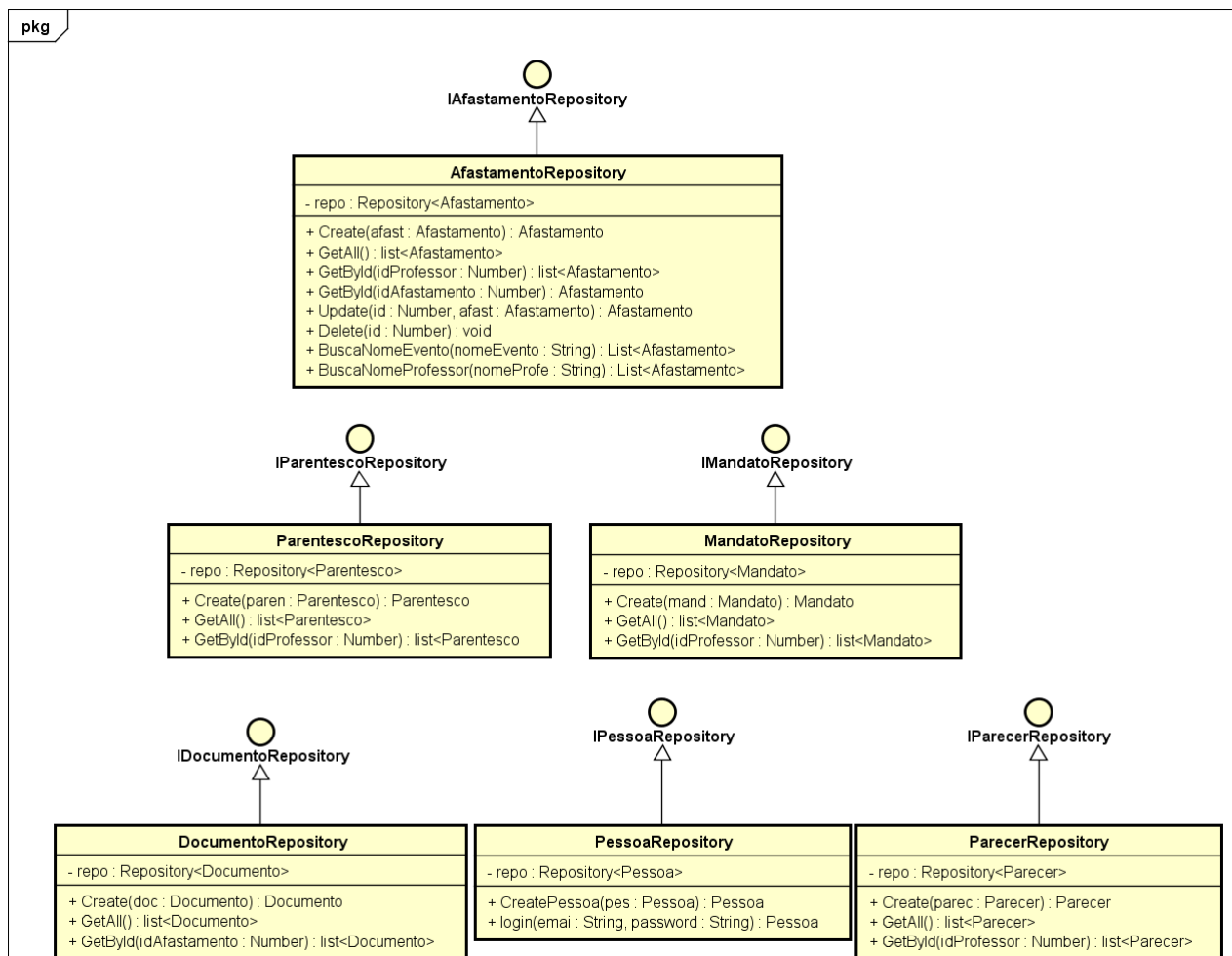


Figura 5 – Modelo de Persistência do SCAP

5.3 Camada de Apresentação

Nesta subseção serão apresentados três modelos de navegação.

A Figura 6 apresenta o Modelo de Navegação para autenticação de *Login* pelos usuários do sistema. A página *Login* é composta por um formulário com e-mail e password. Quando os dados do usuário estão corretos, o método *login*, da classe *PessoaController*, redireciona para a página de *showhome* do SCAP. Porém, se houver alguma inconsistência nessa requisição, o sistema permanece na mesma página.

A Figura 7 apresenta o Modelo de Navegação para o cadastro de um de professor, a página *Professor* é composta por um formulário com os campos necessários para registrar um professor. Após realizar a requisição do método *create*, da classe *PessoaController*, o

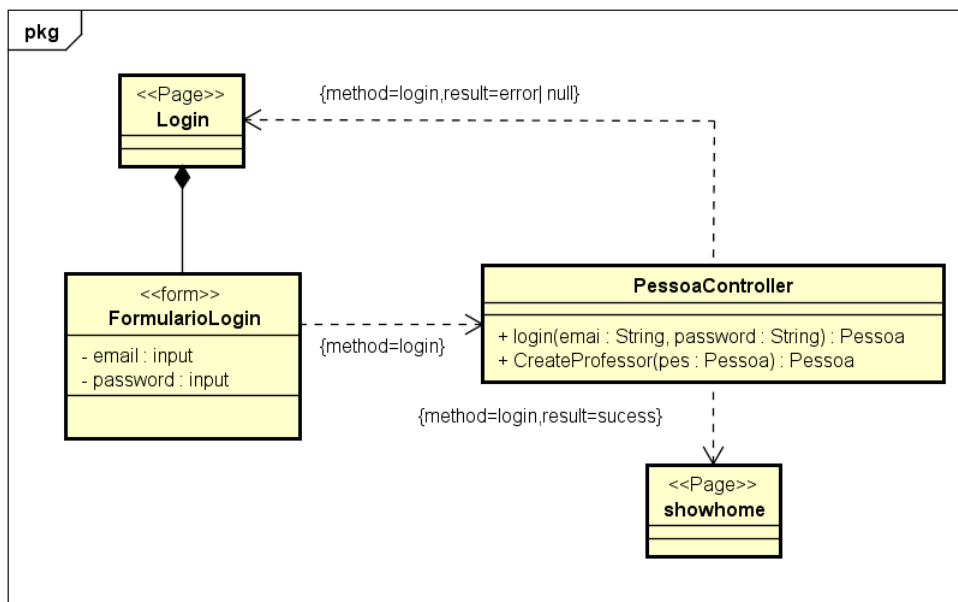


Figura 6 – Modelo de Navegação para autenticação do usuário

o sistema redireciona para a página **showhome** em caso de sucesso ou permanece na mesma página, em caso de erro.

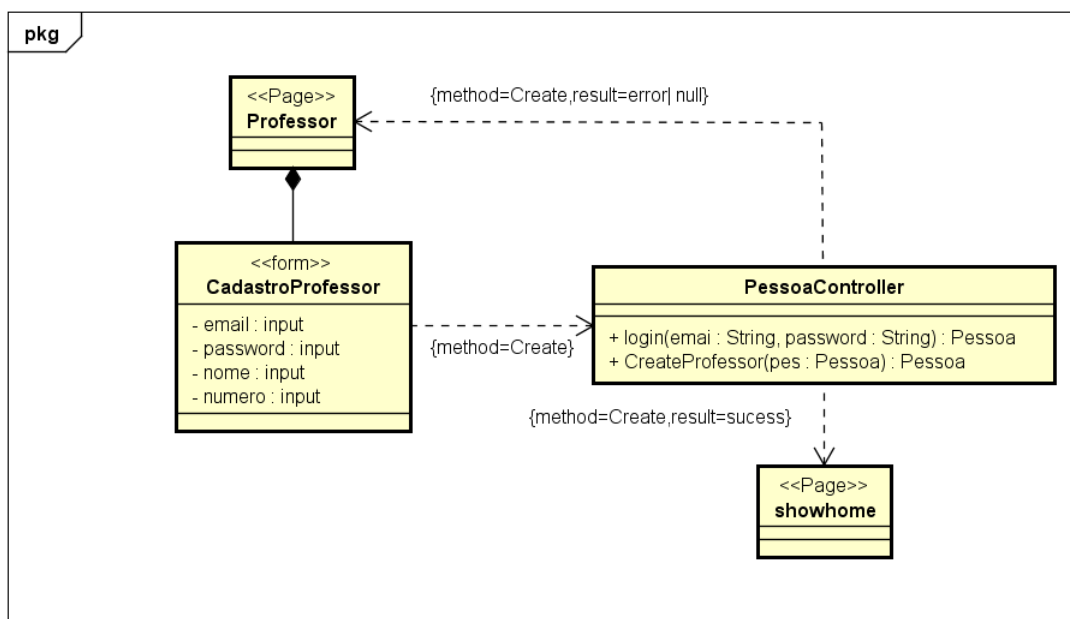


Figura 7 – Modelo de Navegação para cadastro de professor

A Figura 8 apresenta o Modelo de Navegação para o cadastro de um afastamento, a página **Afastamento** é composta por um formulário com os campos necessários para registrar um afastamento. Após realizar a requisição do método **create**, da classe **AfastamentoController**, o sistema redireciona para a página **showhome** em caso de sucesso ou permanece na mesma página, em caso de erro.

A Figura 9 apresenta o Modelo de Navegação para buscar um afastamento pelo

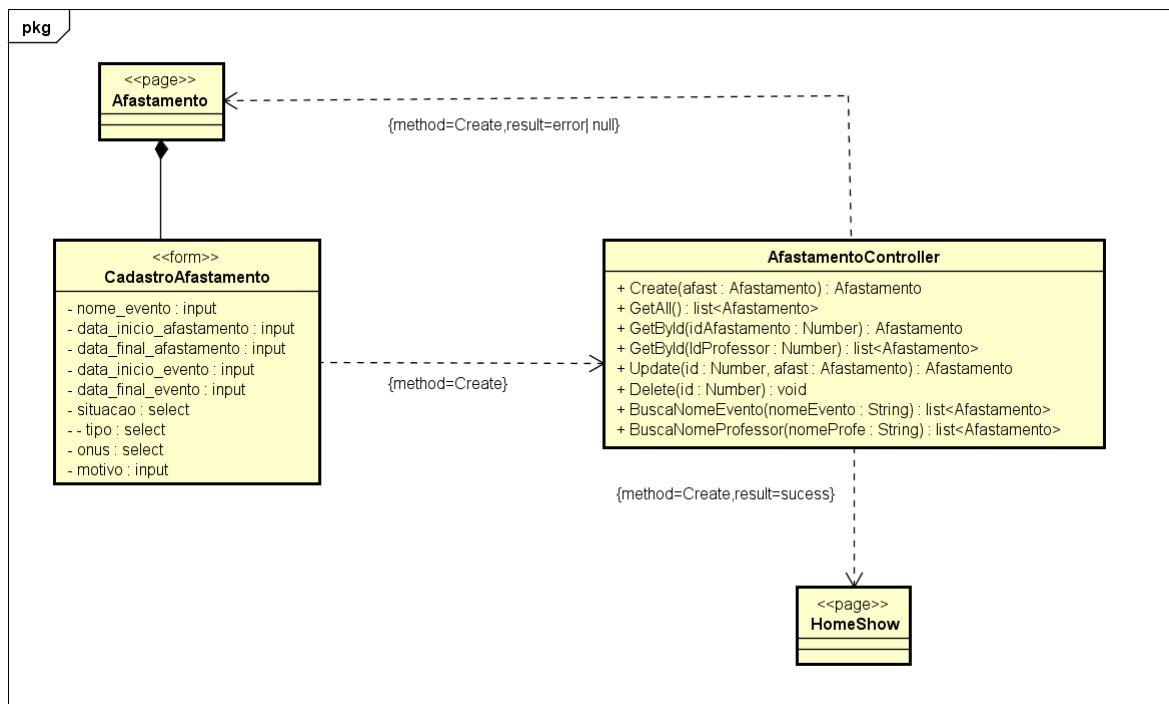


Figura 8 – Modelo de Navegação para solicitação de um afastamento

nome do evento, a página `listAfastamento` contém as listas dos afastamentos solicitados, porém para fazer um busca o usuário tem que digitar o nome do evento e clicar no botão para realizar a busca. Esse buton direciona para `AfastamentoController` que tem o método `BuscaNomeEvento`. Caso seja encontrado, é retornada uma lista de afastamentos com aquele nome, caso contrário o retorno é nulo.

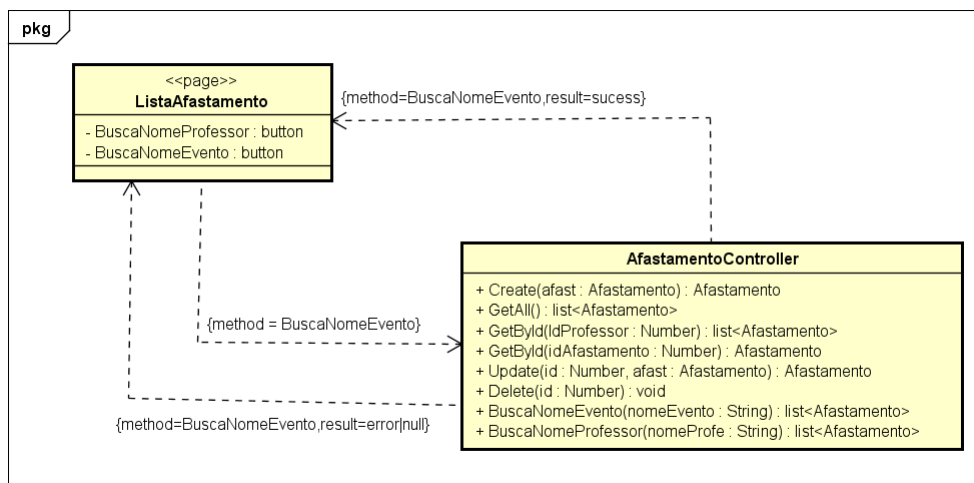


Figura 9 – Modelo de Navegação para buscar um afastamento

Referências

FOWLER, M. *Patterns of Enterprise Application Architecture*. 1. ed. [S.l.]: Addison-Wesley, 2002. ISBN 9780321127426. Citado na página 4.

SOUZA, V. E. S. The FrameWeb Approach to Web Engineering: Past, Present and Future. In: ALMEIDA, J. P. A.; GUIZZARDI, G. (Ed.). *Engineering Ontologies and Ontologies for Engineering*. 1. ed. Vitória, ES, Brazil: NEMO, 2020. cap. 8, p. 100–124. ISBN 9781393963035. Disponível em: <<http://purl.org/nemo/celebratingfalbo>>. Citado 2 vezes nas páginas 4 e 5.