



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
COLEGIADO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO

Renan Moreira Gomes

**Aplicação do método FrameWeb no
desenvolvimento de um sistema de informação
baseado na Web utilizando os frameworks Spring
Boot e NextJS**

Vitória, ES

2024

Renan Moreira Gomes

Aplicação do método FrameWeb no desenvolvimento de um sistema de informação baseado na Web utilizando os frameworks Spring Boot e NextJS

Monografia apresentada ao Curso de Ciência da Computação do Centro Tecnológico 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

Colegiado do Curso de Ciência da Computação

Orientador: Vítor E. Silva Souza

Vitória, ES

2024

Renan Moreira Gomes

Aplicação do método FrameWeb no desenvolvimento de um sistema de informação baseado na Web utilizando os frameworks Spring Boot e NextJS/ Renan Moreira Gomes. – Vitória, ES, 2024-

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

Orientador: Vítor E. Silva Souza

Monografia (PG) – Universidade Federal do Espírito Santo – UFES
Centro Tecnológico
Colegiado do Curso de Ciência da Computação, 2024.

1. SCAP. 2. FrameWeb. I. Souza, Vítor Estêvão Silva. II. Universidade Federal do Espírito Santo. IV. Aplicação do método FrameWeb no desenvolvimento de um sistema de informação baseado na Web utilizando os frameworks Spring Boot e NextJS

CDU 02:141:005.7

Renan Moreira Gomes

Aplicação do método FrameWeb no desenvolvimento de um sistema de informação baseado na Web utilizando os frameworks Spring Boot e NextJS

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

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

Vítor E. Silva Souza
Orientador

Monalessa Perini Barcellos
Universidade Federal do Espírito Santo

Rausth Borges Junqueira
Universidade Federal do Espírito Santo

Vitória, ES
2024

Dedico este trabalho à minha família, que sempre me apoiou incondicionalmente ao longo dessa jornada. Às noites em claro e aos desafios superados, que me trouxeram até aqui. E, especialmente, àqueles que acreditaram no meu potencial, oferecendo motivação nos momentos mais difíceis.

Agradecimentos

Primeiramente, agradeço à minha família, que sempre me ofereceu apoio incondicional, tanto financeiramente, permitindo que eu pudesse ingressar e me manter na faculdade, quanto emocionalmente, nos momentos em que mais precisei. Sou profundamente grato à minha mãe, Edineide, ao meu pai, Marcos, à minha irmã, Priscilla, e à minha vó, Celmi Maria, por todo o amor e suporte ao longo dessa jornada.

Aos meus amigos, tanto aqueles que me acompanharam antes da universidade quanto os que conheci durante minha trajetória na UFES, meu muito obrigado. Vocês foram um incentivo constante, me apoiaram e me ajudaram a trilhar esse caminho. Para os amigos que conheci na faculdade, foi um imenso prazer compartilhar esse tempo com vocês. A experiência foi mais leve graças à companhia de vocês, e enfrentamos várias dificuldades juntos, pelas quais sou profundamente grato por terem participado dessa jornada.

Agradeço a Deus pela oportunidade de realizar este curso incrível, por me abençoar em cada passo e por todas as conquistas ao longo desse caminho.

Por fim, agradeço de coração à minha namorada, Lara Beatriz, pelo carinho e compreensão. Também agradeço ao meu professor orientador, Vítor, pela orientação e sabedoria, e ao meu grande amigo Nathã Rosário, que sempre esteve ao meu lado desde antes da universidade.

Muito obrigado a todos que fizeram parte dessa caminhada.

*“A ciência é o que entendemos bem o suficiente
para explicar a um computador.
A arte é tudo o mais que fazemos.”
– Donald Knuth*

Resumo

Este trabalho apresenta a aplicação do método FrameWeb no desenvolvimento do Sistema de Controle de Afastamento de Professores (SCAP). O objetivo principal é demonstrar a eficácia do método FrameWeb no desenvolvimento de sistemas Web, utilizando tecnologias como Spring Boot e NextJS. O SCAP automatiza o processo de solicitação, acompanhamento e aprovação de afastamentos de professores, proporcionando uma gestão mais eficiente e menos propensa a erros. Ao longo do desenvolvimento, foram aplicados conceitos de Engenharia de Software e Engenharia Web, destacando a modularidade, a reusabilidade de componentes e a clareza arquitetural promovida pelo FrameWeb. A análise dos resultados mostrou que a combinação dessas tecnologias com o método proposto facilita a organização e a documentação do sistema, além de contribuir para a melhoria da escalabilidade e manutenibilidade do software.

Palavras-chaves: FrameWeb. SCAP. Engenharia de Software. Engenharia Web. Spring Boot. NextJS.

Lista de ilustrações

Figura 1 – Processo de Engenharia de Requisitos (FALBO, 2017)	18
Figura 2 – Variações do Modelo Incremental (FALBO, 2014)	22
Figura 3 – Um processo de desenvolvimento de software simples sugerido por FrameWeb (SOUZA, 2007)	23
Figura 4 – Arquitetura padrão para WIS baseada no padrão arquitetônico Service Layer (FOWLER, 2002)	24
Figura 5 – Arquitetura de Software.	32
Figura 6 – Modelo de Entidades	34
Figura 7 – Modelo de Tipos Enumerados do Domínio	35
Figura 8 – Modelo de Aplicação do Subsistema Usuário	36
Figura 9 – Modelo de Aplicação do Subsistema Pedido de Afastamento	36
Figura 10 – Modelo de Persistência do Subsistema Usuário	37
Figura 11 – Modelo de Persistência do Subsistema Pedido de Afastamento	38
Figura 12 – Modelo de Navegação Desautenticado	39
Figura 13 – Modelo de Navegação do Subsistema Usuário	40
Figura 14 – Modelo de Navegação do Subsistema Pedido de Afastamento	42
Figura 15 – Tela de login	43
Figura 16 – Tela de lista de pedidos de afastamento	43
Figura 17 – Tela de formulário de solicitação de um pedido de afastamento	44
Figura 18 – Tela de detalhes do pedido	45
Figura 19 – Tela de formulário para designação de relator	45
Figura 20 – Tela de formulário para emissão de parecer	46
Figura 21 – Tela de upload de documentos relacionados ao pedido de afastamento	46
Figura 22 – Tela de lista de usuários	47
Figura 23 – Tela de formulário para cadastro de usuário	48
Figura 24 – Tela de informações do usuário	49
Figura 25 – Tela de informações do usuário chefe	49
Figura 26 – Tela de edição de informações do usuário	50
Figura 27 – Tela de formulário para salvar parentesco entre usuários	50

Lista de tabelas

Tabela 1 – Especificação de Requisitos Não Funcionais.	30
Tabela 2 – <i>Frameworks</i> da arquitetura do sistema separados por categoria.	33

Lista de abreviaturas e siglas

UML Unified Modeling Language

Sumário

1	INTRODUÇÃO	13
1.1	Motivação e Justificativa	13
1.2	Objetivos	14
1.3	Método de Desenvolvimento do Trabalho	14
1.4	Organização da Monografia	15
2	FUNDAMENTAÇÃO TEÓRICA E TECNOLOGIAS UTILIZADAS	16
2.1	Engenharia de Software	16
2.1.1	Engenharia de Requisitos	17
2.1.2	Projeto de Arquitetura de Software	18
2.1.3	Implementação	19
2.1.4	Testes	19
2.2	Engenharia Web	20
2.3	Método FrameWeb	22
2.4	Frameworks	25
2.4.1	Frameworks Utilizados	26
2.4.1.1	Spring Boot	26
2.4.1.2	NextJS	27
2.4.1.3	Hibernate	27
2.5	Banco de Dados: PostgreSQL	27
2.6	Plataforma de Contêineres: Docker	28
2.6.1	Vantagens do Docker	28
2.6.2	Uso do Docker no Projeto SCAP	28
3	CONTRIBUIÇÃO DO TRABALHO	30
3.1	Requisitos	30
3.2	Arquitetura de Software	32
3.3	Modelagem FrameWeb	33
3.3.1	Camada de Negócio	34
3.3.2	Camada de Acesso a Dados	36
3.3.3	Camada de Apresentação	38
3.4	Telas do Sistema	42
4	CONCLUSÃO	51
4.1	Percepções do Desenvolvedor acerca do FrameWeb	51
4.2	Trabalhos Futuros	52

REFERÊNCIAS 54

APÊNDICES 56

1 Introdução

Os avanços tecnológicos e a crescente necessidade de sistemas de informação eficientes impulsionaram o desenvolvimento de metodologias que facilitam a criação de aplicativos Web. No contexto acadêmico, a gestão de afastamentos de professores é um processo complexo que envolve diferentes etapas e atores. O Sistema de Controle de Afastamento de Professores (SCAP) surge como uma resposta a essa necessidade, automatizando o processo de solicitação, acompanhamento e aprovação de afastamentos.

Este trabalho propõe a aplicação do método *FrameWeb* como uma prova de conceito para demonstrar suas vantagens no desenvolvimento de sistemas de informação. Como exemplo de aplicação prática, será utilizado o SCAP (Sistema de Controle de Afastamentos de Professores), mas o foco principal é a exploração das capacidades do *FrameWeb*. Este capítulo introduz o contexto do trabalho, apresenta a motivação e a justificativa para a escolha do tema, e descreve os objetivos, o método de desenvolvimento e a organização da monografia.

1.1 Motivação e Justificativa

A motivação para esse trabalho reside na oportunidade de experimentar o método *FrameWeb* para o desenvolvimento do sistema SCAP (Sistema de Controle de Afastamentos de Professores), que inicialmente foi experimentado por [Duarte \(2014\)](#), utilizando o Java EE, utilizando seu conjunto de *frameworks* padrão.

Com o mesmo propósito, este trabalho busca motivação em explorar o potencial utilizando as tecnologias Spring Boot e NextJS. O objetivo é avaliar o desempenho e a eficácia do *FrameWeb* ([SOUZA, 2007](#); [SOUZA, 2020a](#)) em conjunto com esse novo conjunto de tecnologias, que ainda não foi amplamente testado nessa abordagem. Ao realizar essa experimentação pretende-se explorar o uso do *FrameWeb*, contribuindo para o aprimoramento e a validação do método em cenários práticos.

Ao aplicá-lo a um conjunto de tecnologias diferentes e verificar a sua adaptação, é possível identificar os pontos fortes e as possíveis limitações dessa abordagem, bem como orientação para projetistas que desejam adotar essa combinação de tecnologias no desenvolvimento de sistemas de informação baseados na Web.

1.2 Objetivos

O objetivo geral deste trabalho de conclusão de curso é aplicar o método *FrameWeb* (SOUZA, 2007; SOUZA, 2020a) para implementar o SCAP (Sistema de Controle de Afastamento de Professores), utilizando os requisitos previamente levantados por Duarte (2014) e refinados por Prado (2015), e as tecnologias Spring Boot e NextJS, a fim de contribuir para a avaliação do método.

Para alcançar esse objetivo geral, os seguintes objetivos específicos devem ser cumpridos:

- Compreender e analisar o método *FrameWeb* (SOUZA, 2007; SOUZA, 2020a), identificando suas principais características e como ele pode ser aplicado para organizar o desenvolvimento de sistemas Web;
- Projetar o sistema SCAP utilizando o *FrameWeb* e implementá-lo integrando os *frameworks* *Spring Boot* e *Next.js*, seguindo os modelos arquiteturais propostos pelo *FrameWeb*;
- Contribuir para a avaliação do método *FrameWeb* a partir do seu uso com um conjunto de *frameworks* diferente dos já utilizados anteriormente.

1.3 Método de Desenvolvimento do Trabalho

O método de desenvolvimento consiste em seguir um método de pesquisa baseado em abordagens da Engenharia de Software e Engenharia Web. Conduzir as atividades de acordo com os seguintes procedimentos:

1. Levantar bibliografia: Realizar um estudo aprofundado da literatura científica relacionada à Engenharia de Software, Engenharia de Requisitos, desenvolvimento de software, Engenharia Web, método *FrameWeb* (SOUZA, 2007; SOUZA, 2020a), requisitos específicos do SCAP e *frameworks* relevantes. Consultar fontes acadêmicas confiáveis, como artigos científicos, livros e publicações relevantes.
2. Analisar requisitos: Analisar e compreender detalhadamente os requisitos de software do SCAP, previamente levantados por Duarte (2014) e refinados por Prado (2015). Fazer uma avaliação crítica desses requisitos para garantir que atendam às necessidades e objetivos do sistema.
3. Aplicar o método *FrameWeb*: Aplicar o método *FrameWeb* para projetar os modelos *FrameWeb* específicos para o SCAP. Seguir as diretrizes e abordagens propostas pelo método, visando à definição da arquitetura do sistema e à organização dos componentes reutilizáveis.

4. Implementar o SCAP: Desenvolver o sistema SCAP com base nos modelos FrameWeb projetados, utilizando os *frameworks* selecionados, como o *Spring Boot* e o *Next.js*. Aplicar técnicas e práticas de programação adequadas para garantir a implementação eficiente e funcional do sistema.
5. Elaborar a monografia: Documentar adequadamente as etapas realizadas durante todo o desenvolvimento do trabalho. Elaborar a monografia final seguindo as normas e diretrizes acadêmicas, apresentando de forma clara e organizada todas as contribuições, análises e conclusões alcançadas ao longo da pesquisa.

1.4 Organização da Monografia

Esta monografia está estruturada em três capítulos, além desta introdução. A seguir, apresento um breve resumo de cada um dos capítulos:

- O **Capítulo 2**, intitulado “Fundamentação Teórica e Tecnologias Utilizadas”, aborda os conceitos e teorias fundamentais que embasam o trabalho. São discutidos temas como Engenharia de Software, Engenharia Web, o método FrameWeb, e as tecnologias utilizadas, como Spring Boot, NextJS e Hibernate.
- O **Capítulo 3**, intitulado “Contribuição do Trabalho”, descreve a principal contribuição deste estudo, que é a aplicação do método FrameWeb no desenvolvimento do Sistema de Controle de Afastamento de Professores (SCAP), destacando as fases de desenvolvimento, a arquitetura do sistema, e os resultados obtidos com a implementação.
- O **Capítulo 4**, intitulado “Considerações Finais”, apresenta as conclusões do trabalho, discutindo os principais aprendizados e desafios encontrados ao longo do desenvolvimento, além de sugerir possíveis trabalhos futuros que podem expandir os estudos realizados.

Com esta organização, busca-se apresentar de forma clara e sequencial o conteúdo abordado, desde a fundamentação teórica até a contribuição prática e as conclusões do trabalho.

2 Fundamentação Teórica e Tecnologias Utilizadas

Este capítulo tem como objetivo fornecer os fundamentos teóricos utilizados no desenvolvimento do trabalho de conclusão de curso. Para isso, serão abordados diferentes temas relacionados à Engenharia de Software e Engenharia Web.

Na Seção 2.1, será apresentado um resumo conciso sobre Engenharia de Software, abordando conceitos essenciais relacionados à Engenharia de Requisitos (2.1.1) e Projeto de Arquitetura de Software (2.1.2). Além disso, serão discutidas as fases de Implementação (2.1.3) e Testes (2.1.4), que completam o ciclo de desenvolvimento de software. A seção abordará o processo de desenvolvimento de software de forma integrada, destacando desde a coleta de requisitos e o planejamento arquitetural até a transformação do projeto em código executável e a verificação de qualidade por meio de testes.

A Seção 2.2 será dedicada ao conceito de Engenharia Web, que é uma vertente da Engenharia de Software voltada para o desenvolvimento de aplicações Web. Serão explorados os principais conceitos e técnicas utilizados na criação de sistemas Web eficientes e de qualidade.

Em seguida, na Seção 2.3, será apresentado o método FrameWeb e suas propostas. O FrameWeb é uma abordagem específica para o desenvolvimento de aplicações Web, que combina conceitos de Engenharia de Software e Engenharia Web. Serão discutidas as características e vantagens desse método.

Na Seção 2.4, será apresentado um resumo geral sobre *frameworks*, incluindo os tipos que serão utilizados no desenvolvimento do sistema SCAP com o método FrameWeb.

Na seção 2.5, será apresentado um resumo sobre o banco de dados utilizado.

Na seção 2.6, será apresentado um resumo sobre contêineres e o uso no desenvolvimento.

Essas seções fornecerão uma base teórica sólida para o entendimento do projeto de graduação, oferecendo insights sobre as abordagens e técnicas utilizadas no desenvolvimento de aplicações Web com o método FrameWeb.

2.1 Engenharia de Software

A Engenharia de Software é um campo que engloba um conjunto de métodos, técnicas e ferramentas voltados para o desenvolvimento de sistemas complexos de alta

qualidade e dentro dos prazos estabelecidos (PRESSMAN, 2011). Essa disciplina é aplicada em todas as etapas do processo de desenvolvimento de software, desde as fases iniciais de especificação do sistema até a implantação e a utilização do sistema em produção, e vai além disso, abrangendo também a manutenção e evolução contínua do software ao longo do seu ciclo de vida.

Um processo de software é composto por uma série de atividades que visam produzir um produto de software (SOMMERVILLE, 2011). Segundo Sommerville (2011), existem quatro atividades fundamentais durante os processos de software:

- Especificação de Software: nessa fase, ocorre a comunicação entre as partes interessadas, como o cliente e o engenheiro de software. O objetivo é definir claramente os objetivos do software, os requisitos a serem atendidos e as restrições envolvidas no projeto;
- Desenvolvimento de Software: essa etapa envolve a criação de modelos e a codificação do software. São realizadas atividades de design, arquitetura, implementação e testes para transformar as especificações em um produto de software funcional;
- Validação de Software: nessa fase, é verificado se os requisitos estabelecidos foram implementados corretamente no software. São realizados testes, revisões e validações para garantir que o sistema atenda às necessidades e expectativas dos usuários;
- Evolução de Software: ao longo do tempo, as necessidades dos usuários e do mercado podem mudar. Nessa etapa, o software é modificado e atualizado para se adaptar às novas demandas. Isso envolve a implementação de novas funcionalidades, correção de erros e melhorias no desempenho e na usabilidade.

Essas quatro atividades são essenciais no desenvolvimento de software, formando um ciclo contínuo de planejamento, desenvolvimento, validação, manutenção e evolução do sistema. Cada etapa requer habilidades técnicas, colaboração entre as partes envolvidas e o uso de metodologias adequadas para garantir a entrega de um software de qualidade.

2.1.1 Engenharia de Requisitos

A Engenharia de Requisitos desempenha um papel fundamental na Engenharia de Software, pois tem como objetivo coletar informações relevantes para a produção de um sistema. Nessa fase, o engenheiro de requisitos trabalha em estreita colaboração com o cliente, buscando compreender o domínio do problema e oferecer sugestões de melhorias. A Figura 1 ilustra o processo de engenharia de requisitos sugerido por Falbo (2017).

Segundo Sommerville (2011), os requisitos de um sistema podem ser definidos como descrições de como o sistema deve funcionar, os serviços que ele deve oferecer e as restrições

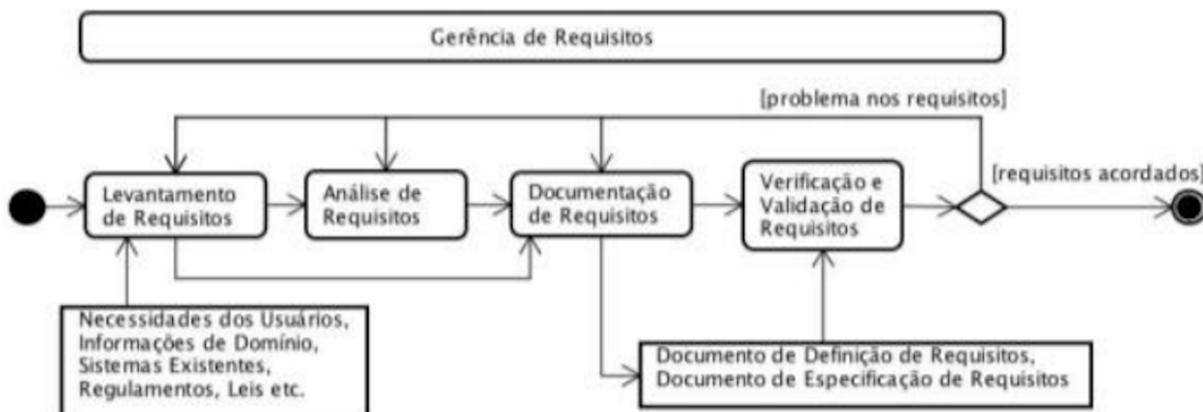


Figura 1 – Processo de Engenharia de Requisitos (FALBO, 2017)

que devem ser consideradas. Em outras palavras, os requisitos refletem as necessidades dos clientes em relação a um determinado sistema. Os requisitos desempenham um papel central no processo de desenvolvimento de software e podem determinar o sucesso ou o fracasso do projeto. Requisitos bem elaborados garantem que as necessidades dos clientes sejam atendidas (FALBO, 2017).

Os requisitos podem ser classificados em funcionais e não funcionais. Os requisitos funcionais descrevem os serviços que o sistema deve realizar e como ele deve responder a uma determinada entrada. Em essência, eles definem o que o sistema deve fazer. Já os requisitos não funcionais impõem restrições aos serviços oferecidos pelo sistema e geralmente estão relacionados ao desempenho, usabilidade, confiabilidade e outros aspectos do uso da aplicação (SOMMERVILLE, 2011).

Dessa forma, uma abordagem sólida na Engenharia de Requisitos é essencial para garantir o desenvolvimento de sistemas que atendam às necessidades e expectativas dos clientes, bem como para estabelecer os parâmetros de desempenho e qualidade que o sistema deve atingir.

2.1.2 Projeto de Arquitetura de Software

O projeto de arquitetura desempenha um papel crucial no desenvolvimento de sistemas, pois visa compreender como o sistema deve ser organizado e estabelecer a estrutura geral desse sistema. É um processo criativo que busca encontrar uma solução que satisfaça os requisitos funcionais e não funcionais do sistema (SOMMERVILLE, 2011). No entanto, as atividades específicas desse processo podem variar dependendo do tipo de sistema a ser desenvolvido.

A fase de projeto arquitetural tem como objetivo aproveitar as informações levantadas nas fases de levantamento e análise de requisitos, adaptando-as e incorporando as tecnologias relevantes para a implementação. Nessa etapa é fundamental ter conhecimento das tecnologias disponíveis no mercado, a fim de definir a melhor abordagem para resolver

o problema em questão (FALBO, 2018).

De acordo com Sommerville (2011), o projeto de arquiteturas pode ser dividido em dois níveis de abstração. O primeiro nível conhecido como arquitetura em pequena escala, concentra-se na arquitetura de programas individuais. Nesse caso, o foco está na decomposição de um programa em componentes que trabalham juntos para alcançar os objetivos do sistema. Já o segundo nível, chamado de arquitetura em grande escala, está relacionado à arquitetura de sistemas corporativos complexos, que podem incluir outros sistemas, programas e componentes de programas.

Em resumo, o projeto de arquitetura é responsável por definir a organização e a estrutura do sistema, levando em consideração os requisitos e as tecnologias envolvidas. É uma etapa fundamental que influencia diretamente na qualidade e no desempenho do sistema final.

2.1.3 Implementação

A fase de implementação é onde o projeto arquitetural é convertido em código executável. Embora um projeto bem estruturado facilite essa etapa, ela pode apresentar desafios significativos. Muitas vezes, os desenvolvedores não estão totalmente familiarizados com todos os detalhes da plataforma de implementação, o que pode impedir uma implementação direta e eficiente. Além disso, questões como legibilidade, alterabilidade e reutilização devem ser consideradas durante o processo (FALBO, 2014).

A utilização de padrões organizacionais, como convenções para nomes de variáveis, formatação de cabeçalhos, comentários, recuo e espaçamento, é fundamental para garantir que o código seja compreensível e fácil de manter. Tais práticas são essenciais em equipes de desenvolvimento, onde o código produzido por diferentes membros precisa ser integrado, testado e modificado de maneira eficiente. Padrões para cabeçalhos, por exemplo, podem incluir informações sobre a funcionalidade do código, autor, datas de criação e revisão, bem como requisitos de teste e entrada e saída esperadas, o que facilita o processo de integração, testes e manutenção (FALBO, 2014). O uso de nomes significativos para variáveis e a aplicação adequada de espaçamento também são essenciais para garantir uma melhor visualização da estrutura de controle do programa, contribuindo para a manutenibilidade do sistema.

2.1.4 Testes

Verificação e validação (V&V) são processos fundamentais para garantir que o software esteja sendo desenvolvido de forma correta e atenda às expectativas do cliente. A verificação foca em assegurar que o sistema foi construído conforme as especificações e utiliza os métodos e processos adequados ao longo do desenvolvimento. Já a validação tem

como objetivo garantir que o produto final entregue é o que o cliente realmente deseja, ou seja, que ele atende aos requisitos estabelecidos (FALBO, 2014). Essas atividades são divididas em estáticas e dinâmicas: a verificação estática analisa o código sem executá-lo, enquanto a validação dinâmica envolve a execução do software por meio de testes.

Um bom teste é ao mesmo tempo econômico e encontra o máximo de defeitos (KOSCIANSKI; SOARES, 2006). Embora os testes não possam garantir a ausência de falhas, eles ajudam a aumentar a confiança na correção do software ao identificar comportamentos indesejados durante a execução (MCGREGOR; SYKES, 2001).

2.2 Engenharia Web

A Engenharia Web, como uma disciplina própria, emergiu no final dos anos 90 em resposta à necessidade crescente de técnicas, metodologias e ferramentas específicas para lidar com a complexidade do desenvolvimento e manutenção de sistemas baseados na Web. Como Pressman (2011) observou, a Engenharia de Software tradicional nem sempre se adequa às demandas e desafios únicos apresentados pela Web. Isso inclui a necessidade de uma rápida evolução, o amplo público de usuários com variadas expectativas, a natureza interativa e multimídia do conteúdo, bem como a necessidade de considerar aspectos como acessibilidade e usabilidade. A Engenharia Web, conforme proposto por Murugesan et al. (2001), sugere métodos sistemáticos e disciplinados para o desenvolvimento, a implementação e a manutenção de sistemas e aplicações fundamentadas na Web.

Olsina et al. (2001) definem uma abordagem estruturada para avaliar a qualidade de sistemas baseados na Web, identificando um conjunto de atributos de qualidade. A usabilidade é vista não apenas como um requisito de qualidade, mas também como um objetivo para aplicações Web. Isso implica um design que permita fácil acesso, forneça *feedback* e assistência online, considere a estética e ofereça acessibilidade para usuários com deficiências. A funcionalidade é outra característica crucial, requerendo que o software Web tenha a capacidade de buscar e recuperar informações, *links* e funções de forma eficaz e que esteja alinhado com as expectativas de navegação do usuário e as necessidades específicas do domínio da aplicação.

Em relação à eficiência, é desejável que o tempo de resposta seja inferior a 10 segundos para a geração de páginas ou gráficos. Já a confiabilidade está relacionada à capacidade do sistema em processar *links* corretamente, se recuperar de erros, validar e recuperar entradas do usuário de maneira eficiente. A manutenibilidade, dada a rápida evolução tecnológica e a constante necessidade de atualização de conteúdo, é um atributo de suma importância, exigindo que o software Web seja fácil de corrigir, adaptar e expandir. Desta forma, essas características formam uma estrutura para avaliar e aprimorar a qualidade de sistemas baseados na Web (OLSINA et al., 2001).

As características distintivas de uma aplicação Web, conforme descrito por [Pressman e Lowe \(2009\)](#), diferem significativamente das de um software convencional. Tais aplicações têm uma natureza orientada à rede, destinada a atender a uma variedade de necessidades do cliente no contexto das redes de computadores, incluindo Internet, Intranet e Extranet. Devido à sua natureza em tempo real e à possibilidade potencial para um grande número de usuários, as aplicações Web precisam ser projetadas para suportar concorrência, lidar com cargas de usuários imprevisíveis e oferecer um desempenho robusto. Para reter usuários, as aplicações devem exibir rapidamente o conteúdo do cliente e garantir uma disponibilidade ideal, apesar de a disponibilidade total de 24 horas por dia, 7 dias por semana, ser pouco provável.

Adicionalmente, as aplicações Web são predominantemente orientadas a dados e sensíveis ao conteúdo, com o objetivo de apresentar dados na forma de texto, gráficos, áudio e vídeo para o usuário final, enquanto garantem uma alta qualidade de exibição de conteúdo. Diferentemente das aplicações convencionais, que evoluem por meio de uma série de versões lançadas em intervalos fixos de tempo, as aplicações Web são dinâmicas e podem ser atualizadas a qualquer momento ([PRESSMAN, 2005](#)). Também é de suma importância que as aplicações Web sejam lançadas no mercado o mais rápido possível, implementem medidas robustas de segurança para proteger o conteúdo e ofereçam modos seguros de transmissão de dados, além de apresentarem uma estética atraente para facilitar a venda de produtos ou ideias.

A Engenharia de Software tradicional e a Engenharia Web compartilham princípios fundamentais, mas diferem significativamente em seus contextos e abordagens. Enquanto a Engenharia de Software tradicional foca no desenvolvimento de sistemas com ciclos de vida mais longos e versões lançadas em intervalos regulares, a Engenharia Web lida com um ambiente dinâmico e em constante evolução, exigindo atualizações frequentes e rápidas, além de priorizar aspectos como acessibilidade, usabilidade e eficiência no carregamento de conteúdo multimídia. Ademais, a Engenharia Web deve considerar um público mais amplo e diverso ([PANDOLFI; MELOTTI, 2006](#)), bem como a necessidade de escalabilidade para lidar com um grande número de usuários simultâneos, o que adiciona complexidade em comparação com sistemas convencionais.

[Pressman e Lowe \(2009\)](#) também detalham cinco atividades principais durante o desenvolvimento de uma aplicação Web, e recomendam o uso de uma abordagem incremental/evolutiva. Isso significa que as atividades de comunicação, planejamento, modelagem, construção e implantação ocorrerão repetidamente à medida que cada incremento da aplicação Web for desenvolvido e entregue. O princípio fundamental é que, a cada ciclo ou iteração (ilustrado na [Figura 2](#)), uma versão operacional do sistema é produzida e entregue para uso ou avaliação detalhada do cliente ([FALBO, 2014](#)). A comunicação é a fase inicial, onde ocorre a interação entre as partes interessadas para entender as

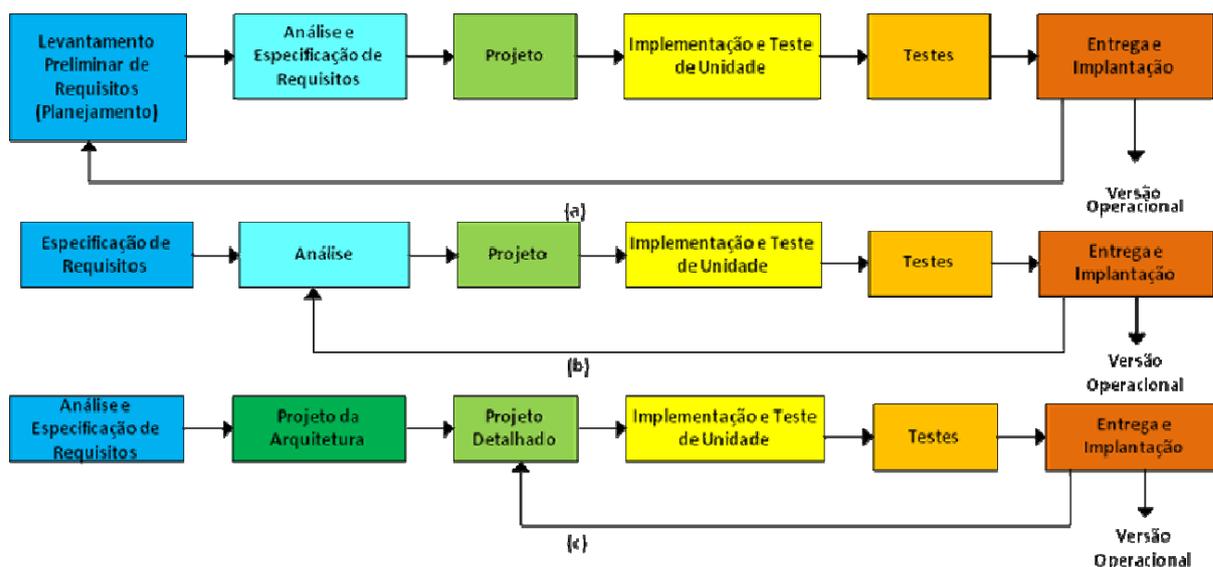


Figura 2 – Variações do Modelo Incremental (FALBO, 2014)

restrições e necessidades do software. O planejamento estabelece um plano incremental para o trabalho de desenvolvimento. A modelagem envolve a criação de modelos para auxiliar o processo de desenvolvimento. A construção envolve a geração de código e o teste para detectar erros. Por fim, a implantação consiste na entrega de um incremento ao cliente para avaliação e *feedback*.

2.3 Método FrameWeb

A reutilização de software tem sido praticada desde o início da programação, empregando recursos como bibliotecas, engenharia de domínio, padrões de design e componentes, entre outros. Um método de reutilização amplamente adotado é a utilização de *frameworks* de software, como o Hibernate, ou arquiteturas de plataforma, como a JavaTM Enterprise Edition. Essas ferramentas atuam como *middleware* onde as aplicações podem ser desenvolvidas. O recurso a tais *frameworks* permite evitar a reinvenção constante de padrões arquitetônicos e componentes fundamentais, o que acarreta em redução de custos e melhora na qualidade de software graças a adoção de arquiteturas e designs já testados e comprovados (SOUZA, 2020a).

O FrameWeb (SOUZA, 2007; SOUZA, 2020a) é um método de design voltado para o desenvolvimento de Sistemas de Informação Web (*Web Information Systems – WISs*), fundamentado na utilização de *frameworks*. Apesar da profusão de *frameworks* disponíveis no mercado, a lacuna de uma metodologia que integrasse suas características para um apoio mais robusto durante a fase de projeto de WISs era evidente.

A proposta do FrameWeb surgiu para atender essa necessidade, visando construir modelos que espelham de maneira fiel a implementação do sistema, sem impor nenhum

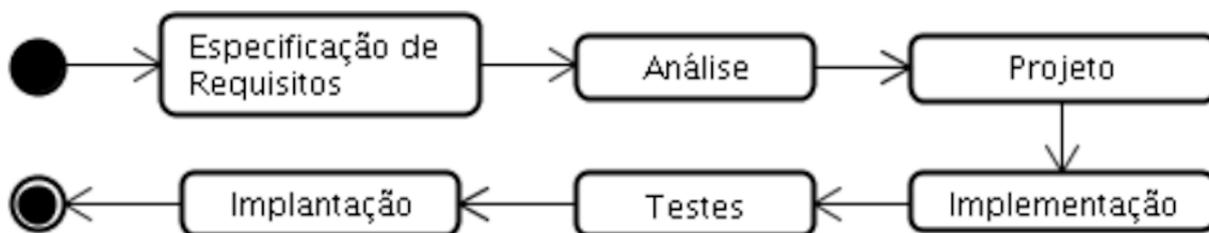


Figura 3 – Um processo de desenvolvimento de software simples sugerido por FrameWeb (SOUZA, 2007)

processo de desenvolvimento específico (SOUZA, 2007). Esta abordagem tem como seu cerne oferecer assistência durante as etapas de projeto arquitetural. Ao fazer isso, proporciona aos projetistas uma visão clara e detalhada das necessidades do sistema e facilita o trabalho dos programadores, uma vez que as fases do projeto arquitetural ficarão a cargo do projetista.

De forma geral, o FrameWeb pressupõe que certos tipos de *frameworks* serão empregados no decorrer da construção da aplicação. Além disso, o método estabelece uma arquitetura base para o WIS e sugere modelos de projeto que se alinham de perto com a implementação do sistema através do uso desses *frameworks* (SOUZA, 2007).

Como um método destinado à fase de Projeto, o FrameWeb não estipula um processo de software completo. Contudo, ele recomenda a adoção de um processo de desenvolvimento que inclua as fases apresentadas na Figura 3. Para maximizar a eficácia do FrameWeb, é esperado que diagramas de casos de uso e de classes de domínio (ou equivalentes) sejam elaborados durante as fases de Requisitos e Análise (SOUZA, 2007).

A etapa de projeto abriga as principais propostas do método FrameWeb: (i) a delimitação de uma arquitetura padrão que estratifica o sistema, de modo a integrar-se de forma eficaz aos *frameworks* adotados; (ii) a oferta de um conjunto de modelos de projeto que incorporam conceitos desses *frameworks* a esta fase do processo, por meio da instauração de um perfil UML que aproxima os diagramas da implementação (SOUZA, 2007).

Segundo Souza (2007), a etapa de codificação é otimizada pelo uso dos *frameworks*, em especial porque os modelos de projeto evidenciam componentes relacionados a eles. A utilização de *frameworks* também pode influenciar as fases de teste e implantação.

A arquitetura lógica padrão do FrameWeb para WISs, ilustrada na Figura 4, divide o sistema em três camadas: lógica de apresentação, lógica de negócio e lógica de acesso a dados, subdivididas em pacotes: Visão, Controle, Aplicação, Domínio e Persistência.

A camada de Lógica de Apresentação, abrigando os pacotes de Visão e Controle, gerencia a interação entre o usuário e o sistema. O pacote de Visão inclui elementos como páginas Web, imagens, scripts executados no cliente, todos focados na apresentação de

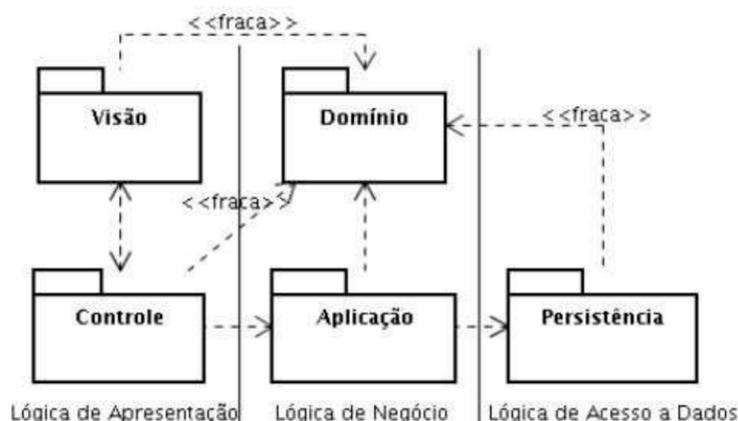


Figura 4 – Arquitetura padrão para WIS baseada no padrão arquitetônico Service Layer (FOWLER, 2002)

informação ao usuário. O pacote de Controle, por sua vez, possui classes de controle e arquivos do *framework* Controlador Frontal, estabelecendo uma dependência recíproca com o pacote de Visão, visto que a Visão aciona as classes do Controle que, em seguida, apresentam as informações pertinentes ao cliente.

Na camada de Lógica de Negócio, encontram-se os pacotes de Domínio e de Aplicação. O pacote de Domínio engloba classes que representam conceitos do domínio do problema, seguindo o diagrama de classes de domínio criado na fase de análise. O pacote de Aplicação implementa as classes que executam os casos de uso identificados, necessitando manipular objetos do domínio, o que configura uma relação de dependência entre os pacotes de Aplicação e Domínio.

A terceira camada, Lógica de Acesso a Dados, abriga o pacote de Persistência, com classes responsáveis pela gravação de atributos de classes que devem ser persistidas no sistema. FrameWeb propõe a utilização de um *framework* para o mapeamento Objeto/Relacional, juntamente com o padrão de projeto *Data Access Object* (DAO) (ALUR; CRUPI; MALKS, 2003), que separa a tecnologia de persistência de dados da Lógica de Negócio, criando uma camada de abstração e tornando a aplicação independente do *framework* ORM.

Para retratar eficazmente os conceitos integrados nos *frameworks*, tornou-se imprescindível a utilização de linguagens de modelagem, uma abordagem seguida por linguagens como WAE (CONALLEN, 2002) e UWE (KOCH, 2000). Nesse contexto, o FrameWeb estabelece extensões sutis ao metamodelo da UML para representar componentes típicos da Web e componentes vinculados aos *frameworks*, concebendo assim uma linguagem de modelagem baseada na UML (SOUZA, 2007).

O método em destaque propõe quatro tipos distintos de modelos que são fundamentados no diagrama de classes da UML:

- Modelo de Entidades: incorpora o diagrama de classes UML, criado durante a fase de análise, que representa o domínio do problema, especificando os tipos de dados utilizados e mapeando as classes de domínio que serão persistidas;
- Modelo de Persistência: este abrange a representação de um diagrama de classes que ilustra as classes DAO. Estas são responsáveis por persistir instâncias das classes de entidade no banco de dados e guiar a implementação do sistema, mostrando todas as interfaces e métodos específicos de cada uma;
- Modelo de Navegação: contém um diagrama de classe da UML que demonstra os componentes variados que compõem a camada lógica de apresentação, incluindo páginas Web, formulários HTML e classes de ação do *framework Front Controller*. Este modelo basicamente ilustra o funcionamento da camada de Lógica de Apresentação;
- Modelo de Aplicação: inclui a representação do diagrama de classes da UML responsável por exibir as classes de serviço, que encarnam os casos de uso e suas dependências correlatas. É usado para guiar a implementação dessas classes e demonstrar suas dependências de classes de outros pacotes.

2.4 Frameworks

Os Sistemas de Informação Web (WISs) apresentam infraestruturas arquitetônicas similares, tornando os *frameworks* uma ferramenta relevante no desenvolvimento desses sistemas. Um *framework* é um artefato de código que fornece componentes reutilizáveis, reduzindo o esforço de codificação necessário para a construção de sistemas Web complexos (SOUZA, 2007).

Os *frameworks* são elementos centrais nas “arquiteturas baseadas em *containers*” Estes *containers* são sistemas que gerenciam objetos com ciclos de vida bem definidos e oferecem serviços como persistência, gerenciamento de transações, distribuição e serviços de diretórios. Um exemplo desses *containers* são os encontrados na plataforma *Java Enterprise Edition* (SOUZA, 2007; SHANNON, 2003).

Segundo Souza (2007), os *frameworks* podem ser categorizados em tipos, por exemplo:

- *Frameworks* MVC (Controladores Frontais): ajudam a separar a lógica de negócios da interface do usuário;
- *Frameworks* Decoradores: encapsulam um objeto para adicionar comportamentos e responsabilidades;
- *Frameworks* de Mapeamento Objeto/Relacional: auxiliam na persistência de dados,

simplificando a conversão entre banco de dados relacional e linguagem de programação orientada a objetos;

- *Frameworks* de Injeção de Dependência (Inversão de Controle): permitem maior flexibilidade e reutilização do código, facilitando o gerenciamento de dependências entre classes;
- *Frameworks* para Programação Orientada a Aspectos (AOP): separam as preocupações transversais em uma aplicação, melhorando a modularidade;
- *Frameworks* para Autenticação e Autorização: garantem a segurança ao controlar o acesso a recursos e serviços.

A escolha de uma linguagem de implementação, como o Java, pode determinar a utilização de *frameworks* específicos. Arquiteturas baseadas em *containers* estão intimamente relacionadas a estes *frameworks*, proporcionando uma estrutura robusta e eficiente para o desenvolvimento de WISs (SOUZA, 2007).

2.4.1 Frameworks Utilizados

Nesta seção, são apresentados os *frameworks* que serão empregados para validar o método proposto, em associação com o FrameWeb (SOUZA, 2007; SOUZA, 2020a). Os *frameworks* selecionados para essa tarefa são o Spring Boot e o NextJS. Como indicado na Seção 1.2, a intenção é utilizar esses *frameworks* na implementação do Sistema de Controle de Afastamento de Professores (SCAP). A escolha por Spring Boot e NextJS deve-se à suas funcionalidades robustas, que facilitam o desenvolvimento de aplicações complexas como o SCAP.

2.4.1.1 Spring Boot

O Spring Boot, como parte do amplo ecossistema Spring, é um *framework* robusto e flexível baseado na linguagem Java. Ele simplifica a configuração e o desenvolvimento de aplicações, oferecendo uma plataforma prática para o desenvolvimento de serviços web RESTful. Sua abordagem de “convenção sobre configuração” reduz a necessidade de tarefas repetitivas e permite aos desenvolvedores focar no desenvolvimento.

Além disso, o Spring Boot integra-se facilmente com outros componentes do ecossistema Spring, como o Spring Data para acesso a dados, o Spring Security para autenticação e autorização, entre outros.

Neste estudo, o Spring Boot será empregado no desenvolvimento do *backend* do Sistema de Controle de Afastamento de Professores (SCAP). O uso deste *framework* irá proporcionar um desenvolvimento eficiente e rápido, facilitando a criação da aplicação.

2.4.1.2 NextJS

NextJS é um *framework* JavaScript desenvolvido sobre o React, oferecendo uma série de funcionalidades para facilitar a criação de aplicações Web de página única (SPA) e aplicações universais JavaScript. O NextJS tem se destacado por sua capacidade de renderização no lado servidor (SSR), o que melhora significativamente o tempo de carregamento das páginas e a otimização para motores de busca (SEO).

O NextJS também oferece recursos como geração de site estático (SSG), roteamento baseado em sistema de arquivos, suporte a CSS e SASS incorporados, além de suportar TypeScript, um *superset* de JavaScript que adiciona tipos estáticos à linguagem. A flexibilidade do NextJS permite que os desenvolvedores escolham entre SSR, SSG ou uma combinação de ambos, dependendo das necessidades do projeto.

Para a implementação do Sistema de Controle de Afastamento de Professores (SCAP), o NextJS será usado para o desenvolvimento do *frontend*. A integração entre o NextJS e o Spring Boot permitirá a construção de uma aplicação Web completa, eficiente e escalável.

2.4.1.3 Hibernate

O Hibernate é um *framework* de mapeamento objeto-relacional (ORM) que simplifica a interação entre a aplicação Java e o banco de dados. Ele permite que desenvolvedores trabalhem diretamente com objetos Java em vez de lidar com consultas SQL, automatizando a tradução entre o modelo de dados orientado a objetos e o modelo relacional.

O Hibernate é especialmente útil para o desenvolvimento de sistemas que requerem uma camada de persistência robusta, como o SCAP. Ao utilizar o Hibernate em conjunto com o *Spring Boot*, a persistência dos dados no banco *PostgreSQL* é facilitada, garantindo a consistência e a integridade dos dados. Além disso, o *framework* suporta transações, cache e otimizações de desempenho, o que o torna uma escolha ideal para sistemas que exigem alta performance e escalabilidade.

2.5 Banco de Dados: PostgreSQL

Para a persistência dos dados, foi utilizado o banco de dados relacional *PostgreSQL*. O *PostgreSQL* é conhecido por sua robustez, escalabilidade e conformidade com os padrões SQL. Ele foi escolhido para o presente trabalho devido à sua capacidade de lidar com grandes volumes de dados de forma eficiente e por ser uma solução open-source amplamente adotada em sistemas de produção.

A integração entre o *Spring Boot* e o *PostgreSQL* foi facilitada pelo uso do *Hibernate*, que realiza o mapeamento objeto-relacional, permitindo que as operações de persistência

no banco de dados sejam realizadas sem a necessidade de escrever SQL manualmente.

2.6 Plataforma de Contêineres: Docker

O Docker é uma plataforma de contêineres que automatiza a implantação de aplicações em ambientes isolados, conhecidos como contêineres. Um contêiner inclui tudo o que uma aplicação precisa para ser executada — código, dependências, bibliotecas e variáveis de ambiente — garantindo que ela funcione de forma consistente, independentemente do ambiente onde é implantada. Essa tecnologia permite que desenvolvedores criem e configurem seus ambientes de desenvolvimento de forma rápida e eficiente, sem as complicações de configuração manual.

2.6.1 Vantagens do Docker

O Docker proporciona diversas vantagens no desenvolvimento de software, entre elas:

- **Portabilidade:** Um contêiner Docker pode ser executado em qualquer máquina com Docker instalado, garantindo consistência entre os ambientes de desenvolvimento.
- **Isolamento de Ambiente:** Cada contêiner funciona de forma isolada, o que evita que a configuração de um serviço interfira no funcionamento de outros.
- **Facilidade de Configuração:** O Docker permite que serviços, como bancos de dados, sejam rapidamente inicializados sem a necessidade de instalação manual, facilitando o trabalho em equipe e a padronização do ambiente de desenvolvimento.

2.6.2 Uso do Docker no Projeto SCAP

No contexto do desenvolvimento do Sistema de Controle de Afastamento de Professores (SCAP), o Docker foi utilizado exclusivamente para subir o banco de dados *PostgreSQL* durante o desenvolvimento. Isso permitiu uma configuração mais rápida e padronizada do banco de dados, eliminando problemas relacionados a diferentes versões ou configurações manuais no ambiente de cada desenvolvedor.

No entanto, o Docker não foi utilizado no ambiente de produção do SCAP. Sua aplicação foi restrita ao ambiente de desenvolvimento, onde o *PostgreSQL* foi executado em um contêiner para garantir a consistência e facilitar a integração com o backend desenvolvido em *Spring Boot*. Ao utilizar o Docker dessa forma, foi possível evitar a necessidade de instalação manual do banco de dados em diferentes ambientes de desenvolvimento, reduzindo o tempo de configuração e as possíveis inconsistências entre os desenvolvedores.

O uso do Docker para subir apenas o *PostgreSQL* foi feito através de um **Dockerfile** simples, garantindo que o banco de dados estivesse sempre disponível e corretamente configurado durante o processo de desenvolvimento. Isso também facilitou a reinicialização rápida do ambiente de desenvolvimento, quando necessário, sem a necessidade de reinstalar ou reconfigurar o banco de dados.

3 Contribuição do Trabalho

Este trabalho apresenta uma contribuição prática ao aplicar o método *FrameWeb* em conjunto com os frameworks *Spring Boot* e *Next.js* no desenvolvimento de um sistema de informação Web. O objetivo principal é avaliar o uso do *FrameWeb* como um método que organiza o desenvolvimento em camadas e facilita a modularidade, reusabilidade e documentação de sistemas complexos. Para demonstrar essa aplicação, foi desenvolvido o Sistema de Controle de Afastamento de Professores (SCAP), que serve como exemplo prático da aplicação do método e das tecnologias selecionadas.

Nesta seção, são detalhados os principais aspectos da contribuição deste trabalho, organizados em quatro áreas-chave, além de um relato sobre os requisitos do sistema: a plataforma de desenvolvimento utilizada, a arquitetura de software proposta, modelagem com *FrameWeb* e telas do sistema. Cada uma dessas áreas representa um componente fundamental do desenvolvimento e implementação do SCAP, contribuindo para uma análise aprofundada das capacidades e limitações do método aplicado.

Mais detalhes da arquitetura e dos modelos *FrameWeb* podem ser encontrados no apêndice. A seguir, estão apresentadas as principais seções que detalham a contribuição deste trabalho:

3.1 Requisitos

Os requisitos deste sistema são inicialmente levantados por [Duarte \(2014\)](#) e posteriormente refinados por [Prado \(2015\)](#). Neste trabalho, focamos exclusivamente nos requisitos não funcionais, que se referem aos atributos de qualidade do sistema, tais como desempenho, segurança, usabilidade e manutenibilidade.

Esses requisitos são ajustados para atender às necessidades específicas do projeto, garantindo que o sistema opere de maneira eficiente e segura, conforme as especificações definidas nos estudos anteriores.

A Tabela 1 apresenta a especificação dos requisitos não funcionais identificados no Documento de Especificação de Requisitos, os quais foram considerados condutores da arquitetura.

Tabela 1 – Especificação de Requisitos Não Funcionais.

RNF-1 – O sistema deve apresentar uma interface de usuário intuitiva, garantindo eficiência mesmo sem uso constante.
--

Categoria:	Usabilidade
Tática / Tratamento:	Utilizar design intuitivo e padronizado, com ícones claros e fluxo de trabalho lógico para facilitar a navegação e o uso.
Medida:	Quantidade de tarefas essenciais completadas em 30 minutos.
Critério de Aceitação:	O usuário novato consegue completar tarefas essenciais na ferramenta sem assistência em menos de 30 minutos.

RNF-2 – O sistema deve implementar um mecanismo de gerenciamento de acesso às suas funcionalidades.

Categoria:	Segurança
Tática / Tratamento:	Implementar um sistema de gerenciamento de usuários com diferentes níveis de permissões, garantindo que cada usuário acesse apenas as funcionalidades adequadas ao seu nível.
Medida:	Quantidade de acessos não autorizados detectados nas auditorias.
Critério de Aceitação:	Todos os usuários devem ter acesso apenas às funcionalidades permitidas pelo seu nível de permissão, sem nenhuma ocorrência de acesso não autorizado, conforme verificado nas auditorias.

RNF-3 – A ferramenta deve ser acessível como uma aplicação Web, compatível com os navegadores mais populares do mercado.

Categoria:	Portabilidade
Tática / Tratamento:	Desenvolver a aplicação utilizando tecnologias Web padrão e responsivas, garantindo compatibilidade com os principais navegadores do mercado.
Medida:	Número de navegadores nos quais a aplicação é testada e funciona corretamente.
Critério de Aceitação:	A aplicação deve funcionar corretamente e de forma consistente nas últimas versões dos principais navegadores, como Chrome, Firefox, Safari e Edge, sem falhas de renderização ou funcionalidades comprometidas.

RNF-4 – O sistema deve possuir uma estrutura que facilite a manutenção, permitindo a inclusão de novas funcionalidades ou adaptações para necessidades específicas.

Categoria:	Manutenibilidade
Tática / Tratamento:	Utilizar padrões de código claros e documentação interna detalhada, seguindo as melhores práticas do <i>framework</i> escolhido, para facilitar a manutenção e expansão do sistema.
Medida:	Tempo médio necessário para implementar novas funcionalidades ou realizar adaptações.
Critério de Aceitação:	Novas funcionalidades ou adaptações específicas devem ser implementadas em um tempo médio pré-definido, com uma margem de erro mínima, sem comprometer a estabilidade do sistema.

RNF-5 – O desenvolvimento do sistema deve empregar os *frameworks* Spring Boot e Next.JS, aderindo ao método FrameWeb.

Categoria:	-
------------	---

Tática / Tratamento:	Implementar o sistema utilizando Spring Boot e Next.js, aplicando as diretrizes do método FrameWeb para estruturar o projeto de forma eficiente.
Medida:	Percentual de aderência às diretrizes do FrameWeb nas revisões de código.
Critério de Aceitação:	O desenvolvimento do sistema deve ser completado com sucesso utilizando apenas Spring Boot e Next.js, e a documentação do projeto deve evidenciar a aplicação efetiva do método FrameWeb em todas as etapas do desenvolvimento.

RNF-6 – O sistema deve ser desenvolvido focando na reutilização de componentes, tanto na perspectiva de desenvolver utilizando componentes reutilizáveis quanto na criação de componentes para reuso futuro.

Categoria:	Manutenibilidade
Tática / Tratamento:	Adotar uma abordagem modular no desenvolvimento com Spring Boot e Next.js, criando componentes reutilizáveis e mantendo um repositório de componentes para facilitar tanto o desenvolvimento com reuso quanto o desenvolvimento para reuso.
Medida:	Percentual de código reutilizável em comparação ao total de código desenvolvido.
Critério de Aceitação:	Um percentual significativo do código (por exemplo, 60%) deve ser composto por componentes reutilizáveis, e estes componentes devem ser implementados em múltiplas partes do sistema ou em projetos futuros, demonstrando efetiva reutilização.

3.2 Arquitetura de Software

A Figura 5 mostra a arquitetura do sistema SCAP, desenvolvido com base nos frameworks Spring Boot e NextJS.

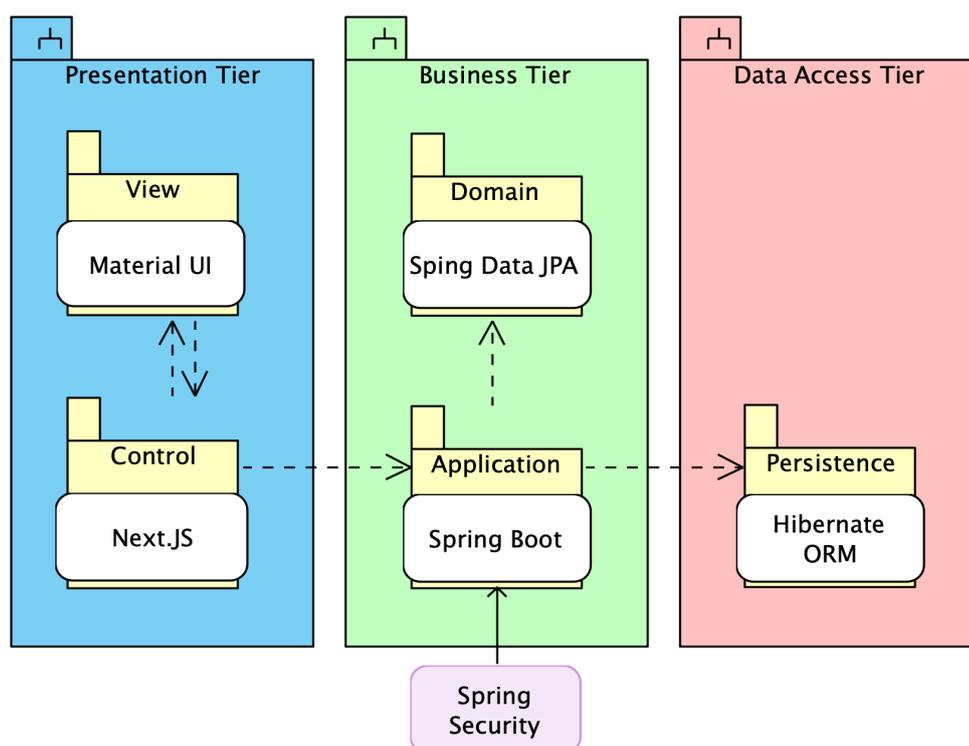


Figura 5 – Arquitetura de Software.

A arquitetura ilustrada na Figura 5 é estruturada em três camadas distintas: *Presentation Tier*, *Business Tier* e *Data Access Tier*. Esta configuração representa uma adaptação da arquitetura proposta pelo FrameWeb (SOUZA, 2020a), baseada no padrão *Service Layer* (FOWLER, 2002).

Adicionalmente, o padrão arquitetônico de camadas, adotado no *Service Layer*, foi integrado ao padrão arquitetônicas de partições. Essa abordagem foi necessária devido à elevada complexidade do sistema desenvolvido, resultando em sua divisão em subsistemas menores e mais gerenciáveis.

Camada de Apresentação: a interface do usuário é desenvolvida com *Material UI*, que oferece componentes visuais ricos e elegantes. *Next.js* é empregado como o *framework* de controle, facilitando a construção de aplicações Web do lado do servidor (*Server Side Rendering* – SSR) ou estáticas, além de gerenciar as interações do usuário e a comunicação com as camadas de negócio.

Camada de Negócios: *Spring Data JPA* é usado para a lógica de domínio e interação com o banco de dados, enquanto *Spring Boot* serve como a estrutura fundamental para o desenvolvimento da aplicação. Além disso, *Spring Security* é integrado para prover segurança robusta, cuidando de autenticação e proteção contra ameaças comuns.

Camada de Acesso a Dados: o *Hibernate* é utilizado para gerenciar a persistência dos dados, facilitando o mapeamento de objetos Java para tabelas de banco de dados e a execução eficiente das operações de banco de dados. Esta camada garante que os dados sejam manipulados de forma segura e eficiente.

3.3 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, 2020b).

A Tabela 2 indica os *frameworks* presentes na arquitetura do sistema que se encaixam em cada uma das categorias de *frameworks* que FrameWeb dá suporte. Em seguida, os modelos FrameWeb são apresentados para cada camada da arquitetura.

Tabela 2 – *Frameworks* da arquitetura do sistema separados por categoria.

Categoria de <i>Framework</i>	<i>Framework</i> Utilizado
Controlador Frontal	Next.JS
Injeção de Dependências	Spring Framework
Mapeamento Objeto/Relacional	Hibernate ORM
Segurança	Spring Security

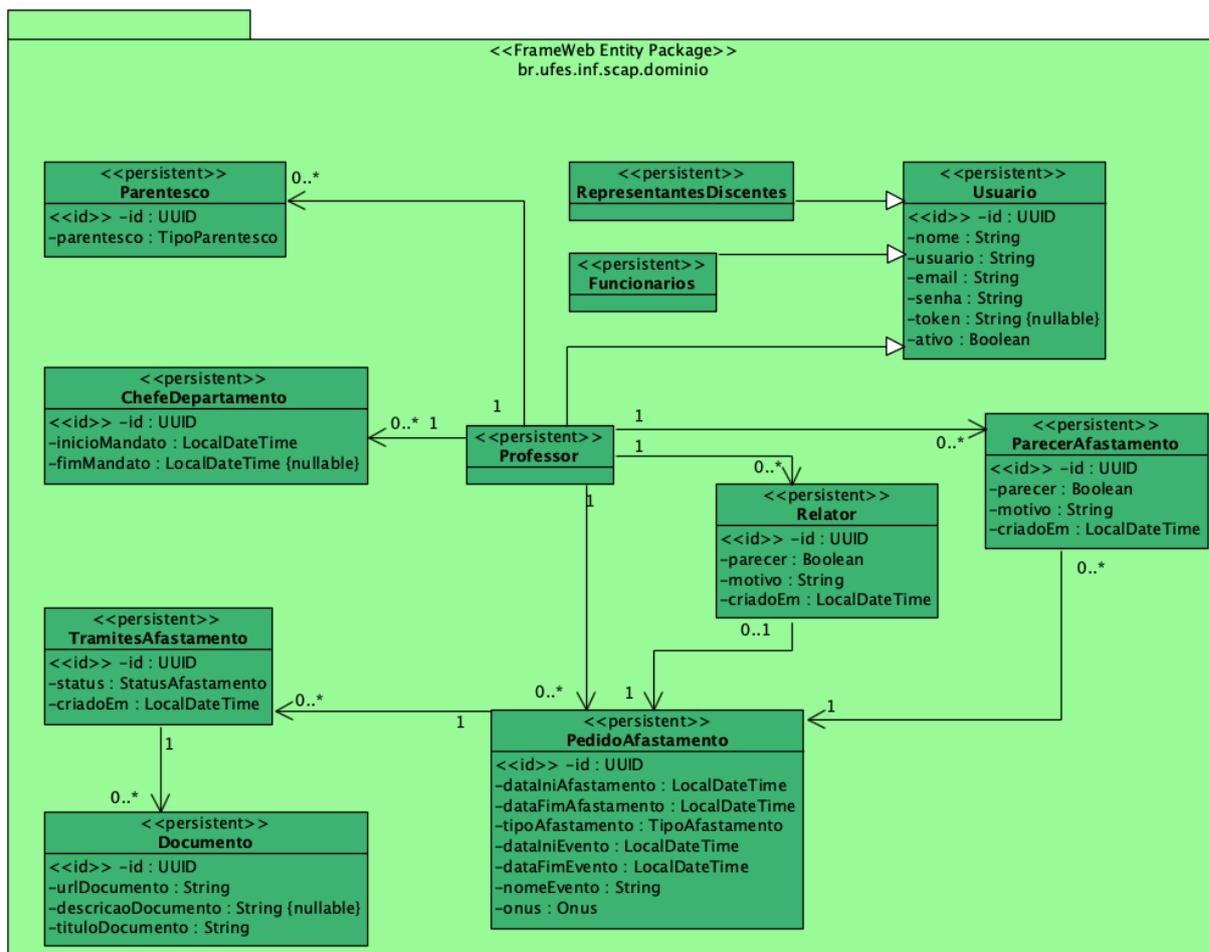


Figura 6 – Modelo de Entidades

3.3.1 Camada de Negócio

O cerne do pacote de domínio descrito na Figura 6 é a entidade **Usuário**, que se desdobra em três categorias principais: funcionário, representante discente e professor. Cada categoria serve a propósitos distintos dentro do sistema, refletindo os diferentes papéis na vida acadêmica.

Os professores, como uma das subclasses de **Usuário**, possuem atributos e relações particulares que destacam suas responsabilidades e conexões pessoais. Eles podem ter N vínculos de parentesco com outros professores, categorizados em relações sanguíneas ou matrimoniais. Além disso, professores podem assumir o papel de **Chefe de Departamento**, com um mandato definido que tem início e fim claros, onde é possível ter apenas um chefe de departamento por vez, ressaltando a natureza temporária e rotativa da administração acadêmica.

A mobilidade acadêmica é representada através dos pedidos de afastamento, permitindo que professores se ausentem para participar de eventos, sejam eles nacionais ou internacionais. Os professores podem solicitar N pedidos. A análise desses pedidos, dependendo do tipo de afastamento, é feita por professores designados como relatores, que



Figura 7 – Modelo de Tipos Enumerados do Domínio

avaliam e emitem pareceres sobre a viabilidade e relevância dos afastamentos solicitados. Um pedido pode ou não ter um relator, dependendo do tipo do afastamento. Outros professores, por sua vez, podem ser chamados a oferecer seus pareceres sobre esses pedidos, criando uma camada de revisão e supervisão colegiada. Um pedido pode ter vários pareceres por parte de outros professores.

O pedido de afastamento, como entidade, possui uma lista de trâmites que indicam o status atual do pedido, e cada status pode estar associado a uma lista de documentos requeridos. Esse mecanismo assegura que o processo de avaliação seja rigoroso e bem documentado, facilitando a gestão administrativa.

Note que algumas propriedades específicas da extensão do FrameWeb, como *persistent*, *id* e *nullable*, estão devidamente definidas e podem ser vistas na Figura 6. No entanto, demais mapeamentos como por exemplo, a restrição *not null*, que é aplicada por *default*, está omitida no diagrama de classes.

Importante destacar que os tipos enumeráveis do modelo, como os tipos de parentesco e os tipos de afastamento, estão claramente definidos na Figura 7. Este detalhamento garante que todas as nuances do modelo sejam facilmente acessíveis e compreensíveis, refletindo a complexidade e a dinâmica do ambiente acadêmico e garantindo que todos os aspectos da administração sejam meticulosamente considerados e geridos através do seu sistema.

O modelo de aplicação do subsistema de Usuário, descrito na Figura 8, é projetado para lidar com todas as operações relacionadas à criação e gerenciamento de perfis de usuários dentro da instituição. Isso inclui a capacidade de registrar novos usuários, seja como funcionários, representantes discentes ou professores, cada um com atributos e permissões específicas. Além disso, o subsistema facilita o registro de relações de parentesco entre professores e a gestão de mandatos para aqueles que ocupam posições de chefe de departamento.

Paralelamente, o modelo de aplicação do subsistema de Pedido de Afastamento, descrito na Figura 9 se concentra em automatizar e organizar o processo pelo qual os

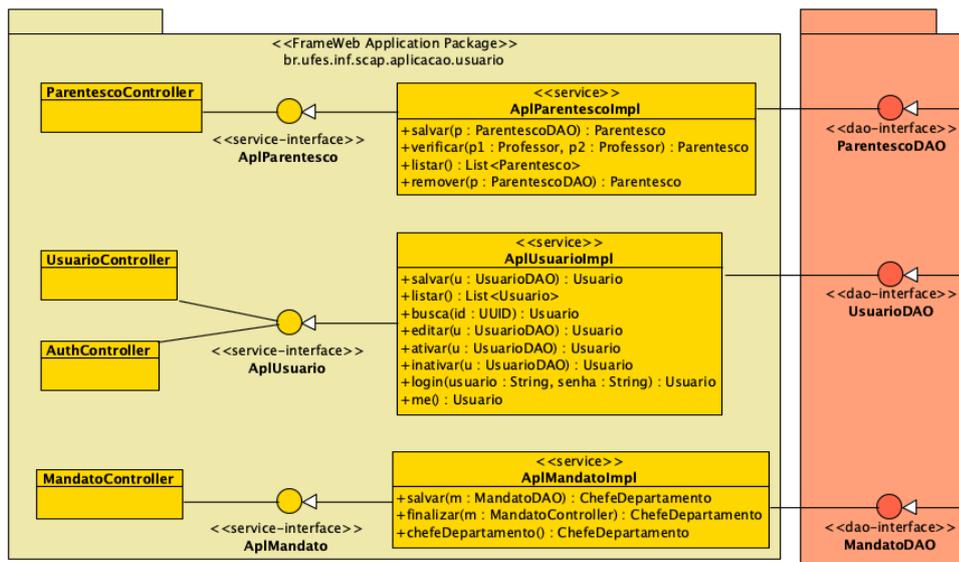


Figura 8 – Modelo de Aplicação do Subsistema Usuário

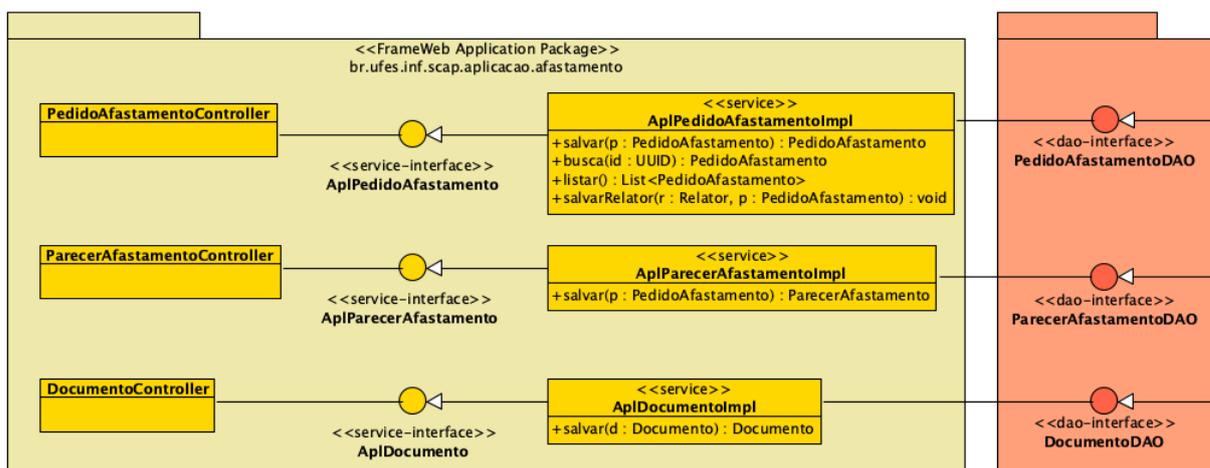


Figura 9 – Modelo de Aplicação do Subsistema Pedido de Afastamento

professores podem solicitar afastamentos. A abertura de novos pedidos, a adição de pareceres, tanto por parte de relatores quanto por outros professores autorizados a opinar sobre os pedidos e o registro metucioso de documentos em cada etapa do processo são elementos chave deste subsistema. Esses processos garantem que cada pedido seja tratado com a devida atenção e rigor, assegurando a transparência e a eficácia na gestão dos afastamentos.

Em ambos os modelos de aplicação, estão presentes as extensões que definem o papel de cada classe ou interface, *service* e *service-interface*.

3.3.2 Camada de Acesso a Dados

Os modelos de persistência em ambos os subsistemas definem uma série de métodos especializados que são fundamentais para as operações diárias dentro da instituição. Por

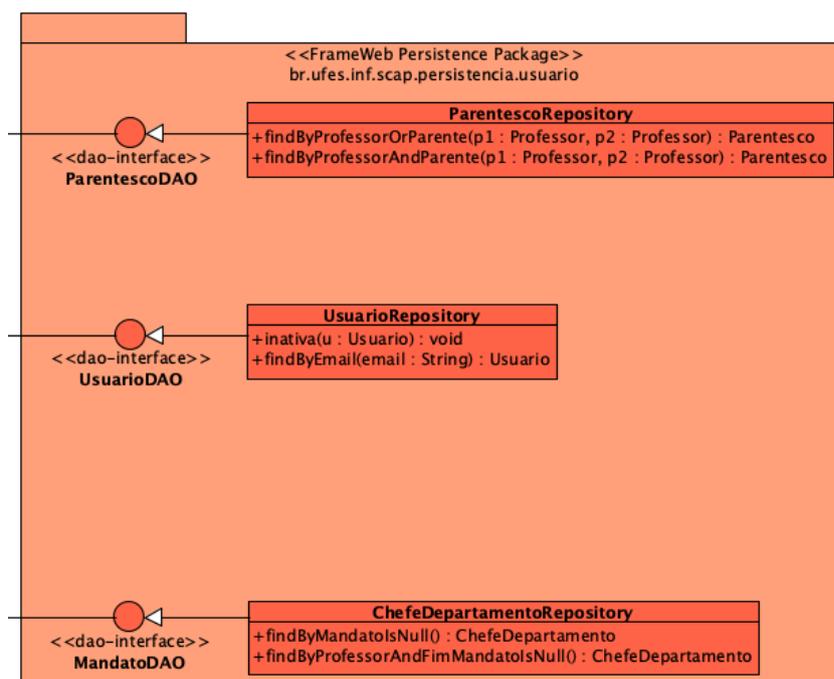


Figura 10 – Modelo de Persistência do Subsistema Usuário

exemplo, no contexto do subsistema de usuários, descrito na Figura 10, métodos como a checagem de parentesco entre professores e a verificação de mandato de chefes de departamento são implementados para assegurar que as políticas internas e as normas de conformidade sejam rigorosamente seguidas. Esses métodos ajudam a manter a integridade das relações e posições dentro da instituição, evitando conflitos de interesse e garantindo a transparência.

Simultaneamente, no subsistema de pedido de afastamento, descrito na Figura 11, a funcionalidade é enriquecida com métodos para listar pedidos de afastamento por professor, permitindo uma visão clara do histórico e do status atual de solicitações de cada docente. Além disso, a listagem de pareceres por pedidos e o registro de trâmites são vitais para monitorar o progresso e as decisões relacionadas a cada pedido, enquanto a listagem de documentos por pedido assegura que todos os requisitos documentais sejam facilmente acessíveis.

Esses modelos de persistência herdam funcionalidades de *repository*, que incorpora as operações básicas de CRUD. Esta herança permite que os modelos de persistência mantenham uma base comum de operações de dados, ao mesmo tempo que expandem suas capacidades com métodos específicos para atender às necessidades únicas de cada subsistema.

No modelo de persistência, o FrameWeb não adiciona extensões específicas. As classes representadas neste modelo são exclusivamente os DAOs (Data Access Objects), que são responsáveis pela comunicação com o banco de dados. O papel do FrameWeb,

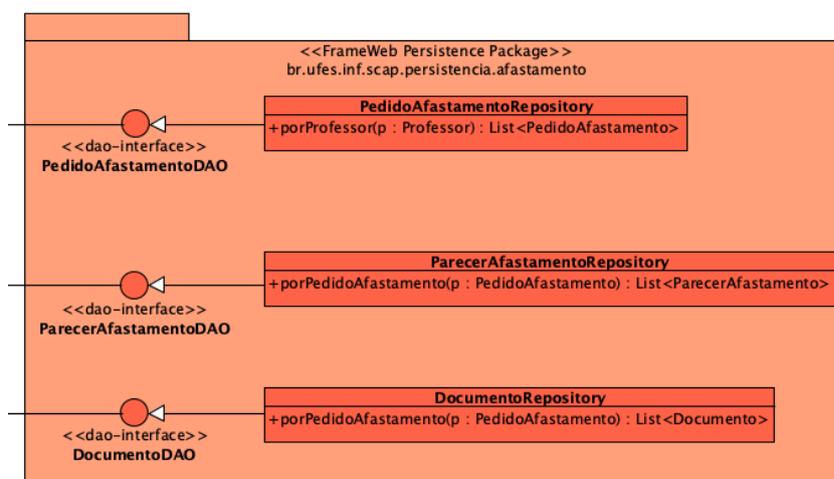


Figura 11 – Modelo de Persistência do Subsistema Pedido de Afastamento

neste caso, é apenas sugerir o uso de DAOs, sem definir diretamente extensões ou atributos adicionais para o modelo de persistência.

3.3.3 Camada de Apresentação

Nos modelos de apresentação, há uma particularidade na nomeação dos pacotes. Esses pacotes são denominados de acordo com os diretórios do código-fonte do Next.js, que incorporam as convenções do React. Temos alguns elementos que ajudam a representar a estrutura de uma aplicação Web de forma clara e organizada. Esses elementos são descritos usando estereótipos do UML (*Unified Modeling Language*) para indicar seu propósito e papel na navegação da aplicação:

- **«page»**: representa uma página web. No caso da Figura 12, o usuário acessa uma página de login chamada *indexDesautenticado*, que é representada por um elemento «page». Esse estereótipo indica que estamos falando de uma tela ou página completa da aplicação;
- **«form»**: indica um formulário que aparece em uma página Web. O formulário pode conter campos para entrada de dados, botões de envio, entre outros. No exemplo do formulário de login da Figura 12, temos um «form» que contém campos de entrada para o usuário inserir o nome de usuário e a senha. Esses campos são componentes da biblioteca de UI utilizada, como por exemplo o *TextField* no caso do Material UI, que é um componente bastante comum para capturar entradas de texto;
- **«partial»**: este elemento é usado para representar componentes que aparecem em várias páginas, mas não formam uma página completa. Um exemplo comum seria o *Header* da aplicação, que é um componente exibido no topo de várias páginas para fornecer navegação ou exibir o logo da aplicação.

O modelo de navegação desautenticado, descrito na Figura 12 é fundamental para gerenciar o acesso inicial ao sistema. Este modelo controla o processo de login, verificando as credenciais do usuário e determinando se o acesso pode ser concedido. Caso as credenciais sejam inválidas ou o acesso seja negado, o usuário permanece na página de login. Se o login for bem-sucedido, o usuário é encaminhado para a área autenticada do sistema. Este modelo é a primeira camada de segurança e serve como um filtro para garantir que apenas usuários autorizados possam prosseguir para as funcionalidades internas do sistema.

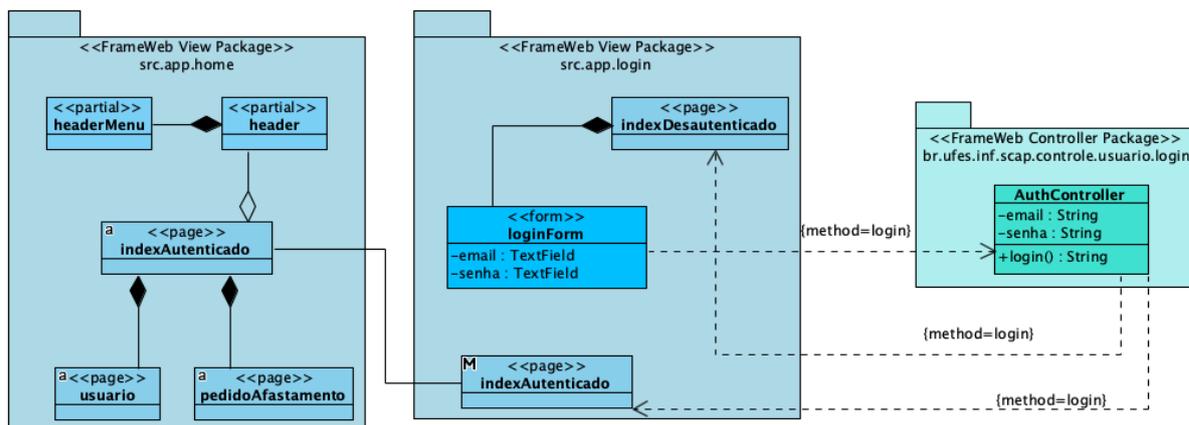


Figura 12 – Modelo de Navegação Desautenticado

Uma vez autenticado, o usuário ganha acesso à página inicial, que apresenta um menu que dá acesso às principais funcionalidades do sistema. Este modelo inclui links ou botões para as páginas de pedido de afastamento e a página de usuário, permitindo que os usuários naveguem facilmente entre essas áreas de acordo com suas necessidades e permissões. Este modelo de navegação é projetado para facilitar o acesso rápido e eficiente às funções que o usuário está autorizado a utilizar.

O modelo de navegação do subsistema Usuário, descrito na Figura 13, é específico para detalhar o comportamento dos componentes e controladores envolvidos na navegação dentro deste subsistema. Este modelo descreve como os usuários interagem com as páginas relacionadas ao gerenciamento de perfil, cadastro de novos usuários, e outras funcionalidades específicas do gerenciamento de usuários. Ele garante que as interações sejam intuitivas e que os usuários possam realizar suas tarefas relacionadas à administração de usuários sem dificuldades.

O modelo de navegação ilustrado na Figura 13 descreve uma página onde há uma tabela de usuários, possibilitando ao usuário acessar informações detalhadas sobre cada usuário. Ao selecionar um usuário na tabela, o sistema exibe um componente de detalhes que inclui três formulários: *infoUsuárioForm*, *ParentescoForm* e *ChefeDepartamentoForm*, permitindo uma interação mais profunda com os dados desse usuário específico.

Por exemplo, se o usuário quiser adicionar uma relação de parentesco, ele deve acessar as informações detalhadas do usuário e submeter o *ParentescoForm* com os

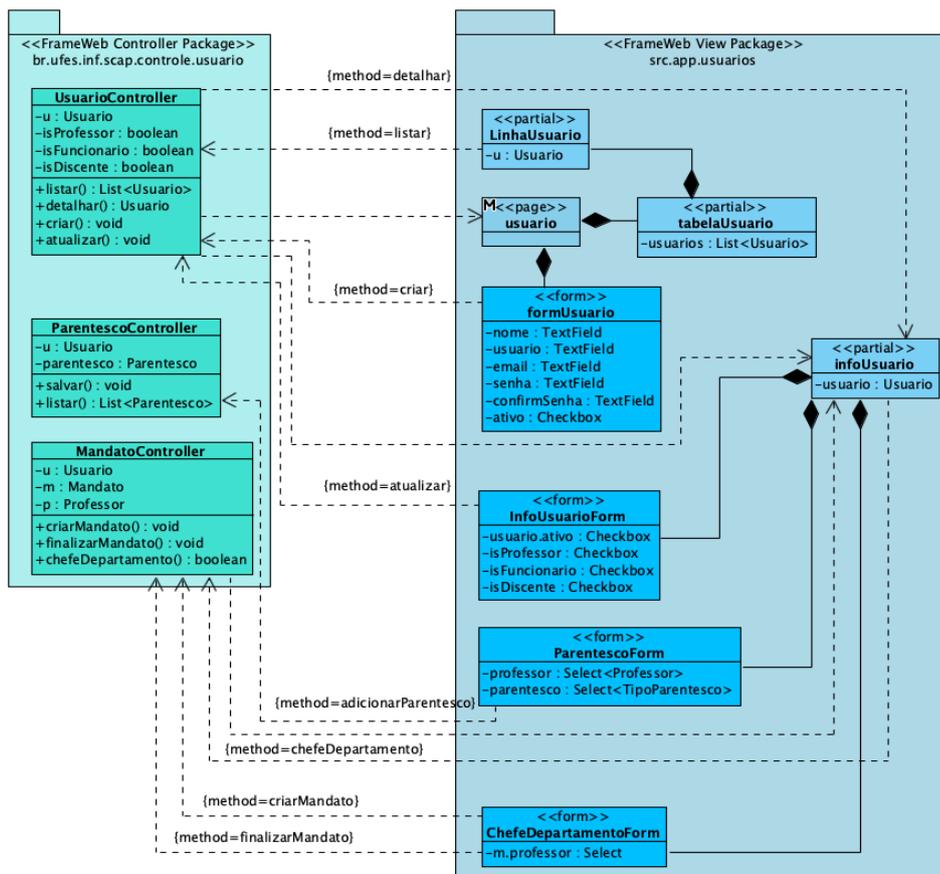


Figura 13 – Modelo de Navegação do Subsistema Usuário

dados necessários. Da mesma forma, os outros formulários, como o *infoUsuárioForm* e o *ChefeDepartamentoForm*, podem ser preenchidos e enviados conforme as necessidades do usuário. Além disso, a página também contém um formulário separado, destinado à criação de novos usuários, permitindo o cadastro de novos dados diretamente no sistema.

O modelo descreve também que os componentes de formulário e componentes de exibição de dados se relacionam diretamente com os controladores responsáveis por gerenciar as respectivas ações e dados. Por exemplo, o *UsuarioController* lida com ações relacionadas ao cadastro e edição de informações do usuário, enquanto o *ParentescoController* gerencia as interações associadas ao *ParentescoForm*. Da mesma forma, o *MandatoController* é acionado em interações com o *ChefeDepartamentoForm*. Dependendo da ação executada, seja o envio de um formulário ou a busca de dados, o modelo descreve as chamadas apropriadas para os controladores correspondentes, garantindo a integração entre o pacote de visão e o pacote de controle.

Similar ao modelo de navegação do Subsistema Usuário, o modelo de navegação do Subsistema Pedido de Afastamento, descrito na Figura 14, articula o comportamento dos componentes e controladores que guiam o usuário através do processo de abertura, acompanhamento e gestão de pedidos de afastamento. Desde a solicitação inicial até a

adição de documentos e recebimento de pareceres, este modelo assegura que os usuários possam navegar eficientemente por todas as etapas necessárias para gerenciar seus pedidos de afastamento.

O modelo de navegação ilustrado na Figura 14 descreve uma página semelhante ao modelo de navegação do Subsistema Usuário, onde há uma tabela de pedidos de afastamento. Essa tabela permite ao usuário acessar informações detalhadas sobre cada pedido. Ao selecionar um pedido na tabela, o sistema exibe um componente de detalhes que contém três formulários: *RelatorForm*, *ParecerForm* e *DocumentoForm*, permitindo ao usuário interagir de forma mais detalhada com os dados do pedido de afastamento específico.

Por exemplo, se o usuário deseja adicionar um parecer a um pedido, ele deve acessar as informações detalhadas do pedido e submeter o *ParecerForm* com as informações relevantes. Da mesma forma, os outros formulários, como o *RelatorForm* e o *DocumentoForm*, podem ser preenchidos e enviados conforme necessário para complementar o pedido de afastamento com informações adicionais. Além disso, a página contém um formulário separado, chamado *PedidoAfastamentoForm*, que permite a criação de novos pedidos de afastamento diretamente no sistema.

O modelo também descreve que esses componentes de formulário e componentes de exibição de dados se relacionam diretamente com os controladores que gerenciam as respectivas ações e dados. Por exemplo, o *RelatorController* é responsável por gerenciar as interações associadas ao *RelatorForm*, enquanto o *ParecerController* lida com as ações relacionadas ao *ParecerForm*. O *DocumentoController* é acionado em interações com o *DocumentoForm*.

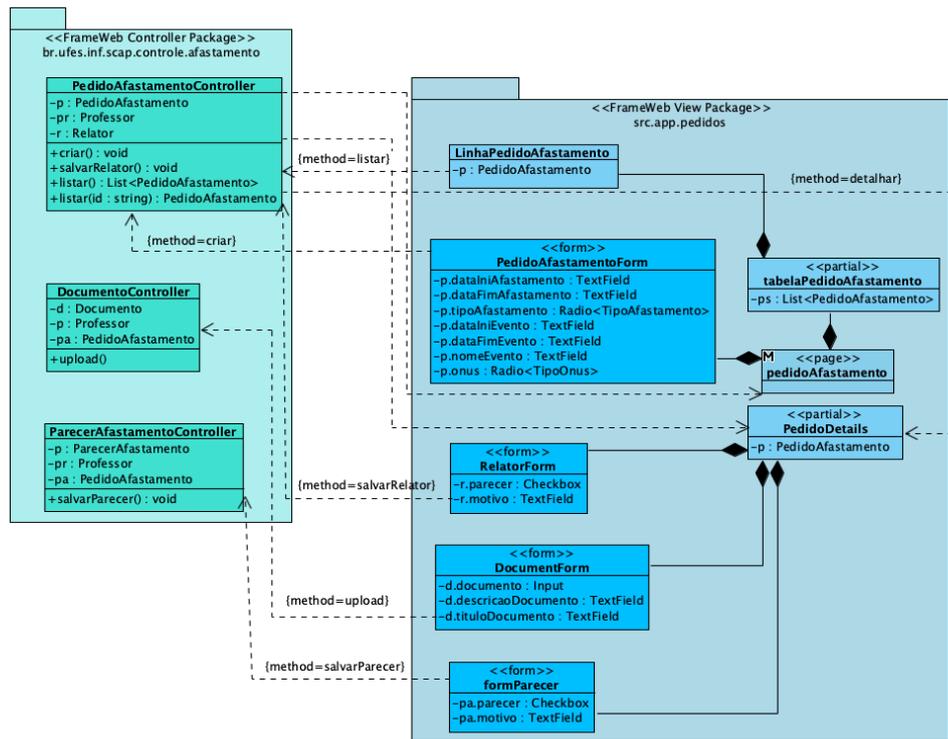


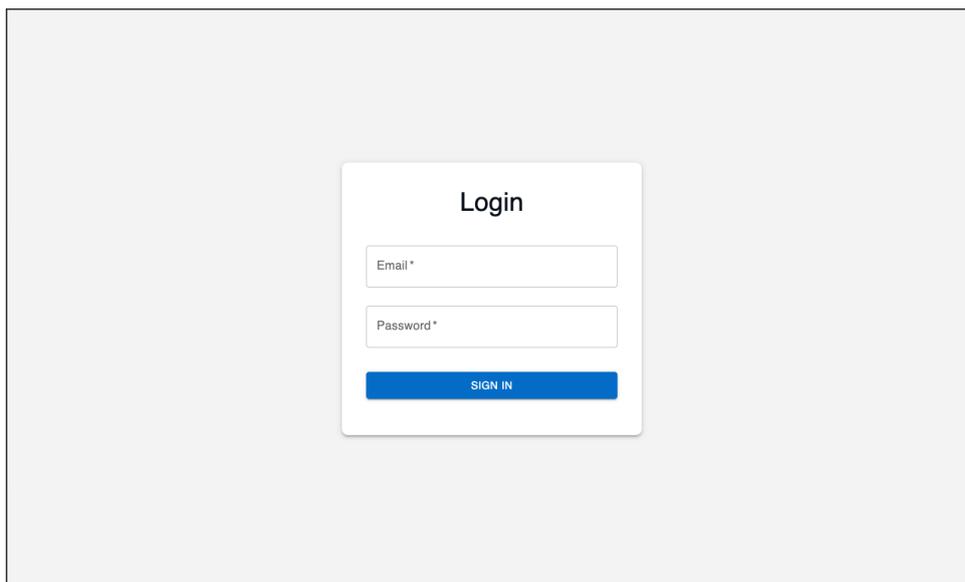
Figura 14 – Modelo de Navegação do Subsistema Pedido de Afastamento

3.4 Telas do Sistema

Nesta seção, são apresentadas as principais telas do Sistema de Controle de Afastamento de Professores (SCAP), desenvolvido como parte deste trabalho. As interfaces foram projetadas com foco na usabilidade e simplicidade, utilizando o *Next.js* para o *frontend*, garantindo uma experiência de usuário eficiente e responsiva.

As telas ilustram as funcionalidades-chave do sistema, desde a solicitação de afastamentos até o acompanhamento e aprovação. Cada tela foi cuidadosamente desenvolvida para atender aos requisitos levantados e garantir uma navegação intuitiva. A seguir, detalhamos as principais interfaces do sistema, destacando seus componentes e a interação entre as diferentes funcionalidades.

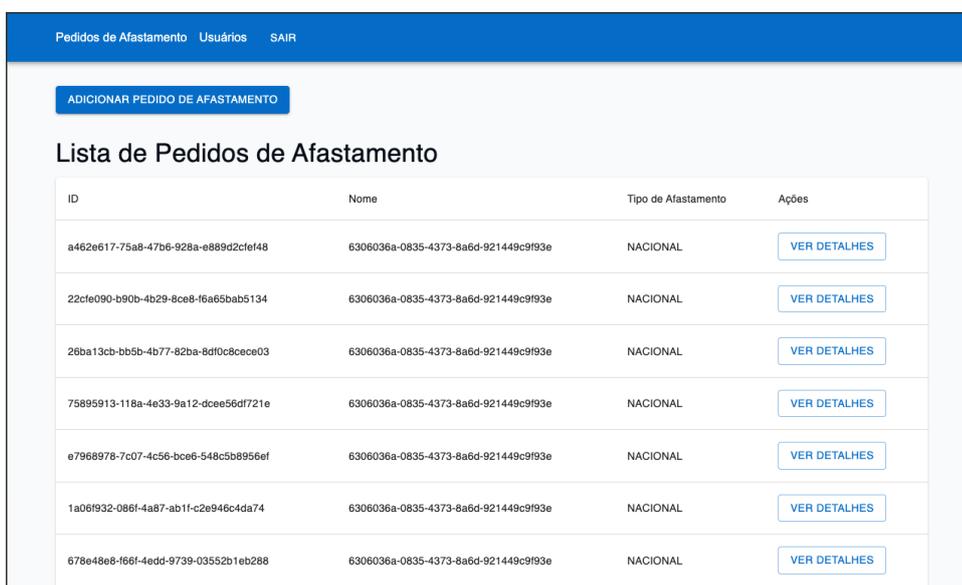
A tela de login, ilustrada na Figura 15, mantém o usuário nela até que as credenciais sejam inseridas corretamente e enviadas com sucesso. Somente após a validação das credenciais, o usuário é redirecionado para outras partes do sistema, assim como descrito na Figura 12.



Um formulário de login centralizado em uma tela cinza. O formulário tem o título "Login" no topo. Abaixo dele, há dois campos de entrada: "Email *" e "Password *". Abaixo dos campos, há um botão azul com o texto "SIGN IN".

Figura 15 – Tela de login

O modelo de navegação de pedido de afastamento, ilustrado na Figura 14, descreve como os componentes interagem no processo de gestão de pedidos de afastamento. Abaixo, descrevemos cada uma dessas telas:



Uma interface de usuário para a gestão de pedidos de afastamento. No topo, há uma barra azul com o texto "Pedidos de Afastamento", "Usuários" e "SAIR". Abaixo, há um botão azul "ADICIONAR PEDIDO DE AFASTAMENTO". O título principal é "Lista de Pedidos de Afastamento". Abaixo dele, há uma tabela com as seguintes colunas: ID, Nome, Tipo de Afastamento e Ações. Cada linha da tabela contém um ID único, um nome de usuário, o tipo de afastamento (NACIONAL) e um botão "VER DETALHES".

ID	Nome	Tipo de Afastamento	Ações
a462e617-75a8-47b6-928a-e889d2cfe48	6306036a-0835-4373-8a6d-921449c9f93e	NACIONAL	VER DETALHES
22cfe090-b90b-4b29-8ce8-f6a65bab5134	6306036a-0835-4373-8a6d-921449c9f93e	NACIONAL	VER DETALHES
26ba13cb-bb5b-4b77-82ba-8df0c8cece03	6306036a-0835-4373-8a6d-921449c9f93e	NACIONAL	VER DETALHES
75895913-116a-4e33-9a12-dcee56df721e	6306036a-0835-4373-8a6d-921449c9f93e	NACIONAL	VER DETALHES
e7968978-7c07-4c56-bce6-548c5b8956ef	6306036a-0835-4373-8a6d-921449c9f93e	NACIONAL	VER DETALHES
1a06f932-086f-4a87-ab1f-c2e946c4da74	6306036a-0835-4373-8a6d-921449c9f93e	NACIONAL	VER DETALHES
678e48e8-f661-4edd-9739-03552b1eb288	6306036a-0835-4373-8a6d-921449c9f93e	NACIONAL	VER DETALHES

Figura 16 – Tela de lista de pedidos de afastamento

A Figura 16 mostra a *tela de lista de pedidos de afastamento*, onde o usuário pode visualizar todos os pedidos já cadastrados no sistema. Nessa tela, é possível selecionar um pedido específico para visualizar seus detalhes ou iniciar um novo pedido através do botão de criação.

A Figura 17 representa a *tela de formulário de solicitação de um pedido de afastamento*, onde o usuário pode preencher os dados necessários para solicitar um novo pedido.

Pedidos de Afastamento Usúários SAIR

ESCONDER FORMULÁRIO

Adicionar Pedido de Afastamento

Data Início do Afastamento
dd/mm/yyyy, --:--

Data Fim do Afastamento
dd/mm/yyyy, --:--

Tipo de Afastamento

Data Início do Evento
dd/mm/yyyy, --:--

Data Fim do Evento
dd/mm/yyyy, --:--

Nome do Evento

Ônus

ADICIONAR PEDIDO

Lista de Pedidos de Afastamento

ID	Nome	Tipo de Afastamento	Ações
a462e617-75a8-47b6-928a-e889d2cfe48	6306036a-0835-4373-8a6d-921449c9f93e	NACIONAL	VER DETALHES
22cfe090-b90b-4b29-8ce8-f6a65bab5134	6306036a-0835-4373-8a6d-921449c9f93e	NACIONAL	VER DETALHES
26ba13cb-bb5b-4b77-82ba-8df0c8cece03	6306036a-0835-4373-8a6d-921449c9f93e	NACIONAL	VER DETALHES

Figura 17 – Tela de formulário de solicitação de um pedido de afastamento

Esse formulário captura informações importantes, como as datas de início e término do afastamento, e outras informações pertinentes ao processo.

Na Figura 18, temos a *tela de detalhes do pedido*, que exibe informações detalhadas sobre um pedido de afastamento específico. Nessa tela, o usuário pode visualizar dados previamente preenchidos e, se necessário, adicionar informações adicionais ao pedido.

A Figura 19 apresenta a *tela de formulário para designação de relator*, onde o usuário pode indicar um relator responsável pelo pedido de afastamento. Esse formulário permite a designação de um membro para revisar um pedido de afastamento internacional e emitir um parecer sobre ele.

A Figura 20 ilustra a *tela de formulário para emissão de parecer*, onde qualquer professor (desde que não haja relação de parentesco), pode registrar seu parecer sobre o pedido de afastamento. Essa funcionalidade é essencial para o processo de avaliação e aprovação do pedido.

Por fim, a Figura 21 apresenta a *tela de upload de documentos relacionados ao pedido de afastamento*, onde o usuário pode adicionar documentos complementares ao pedido.

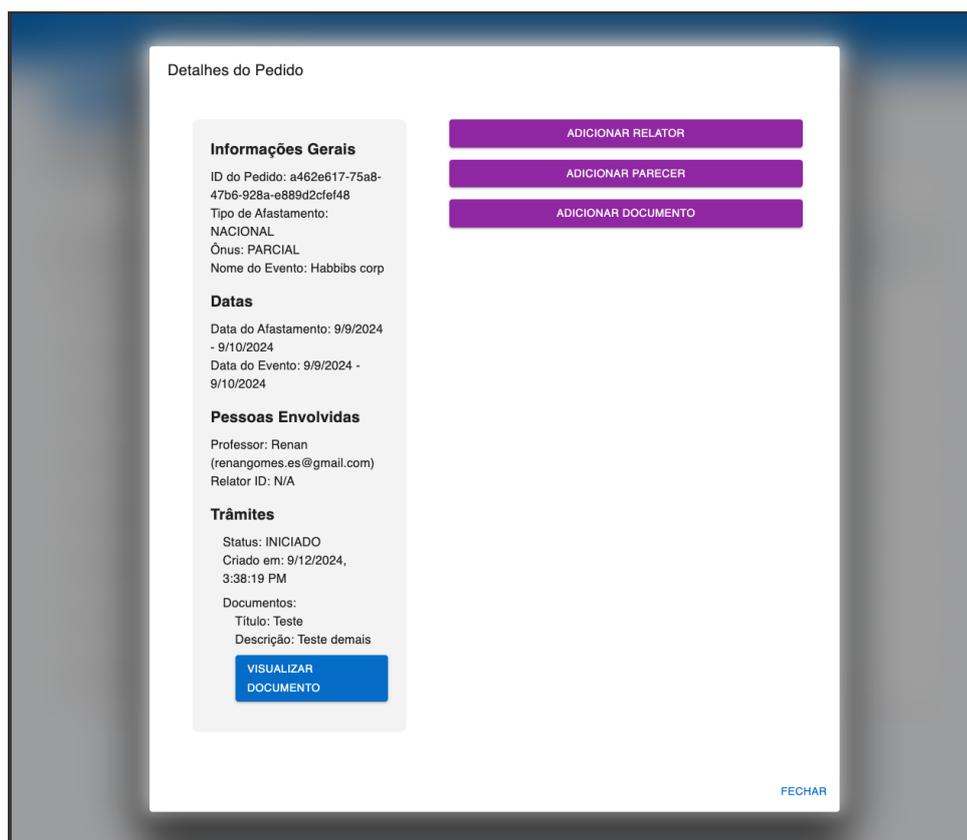


Figura 18 – Tela de detalhes do pedido

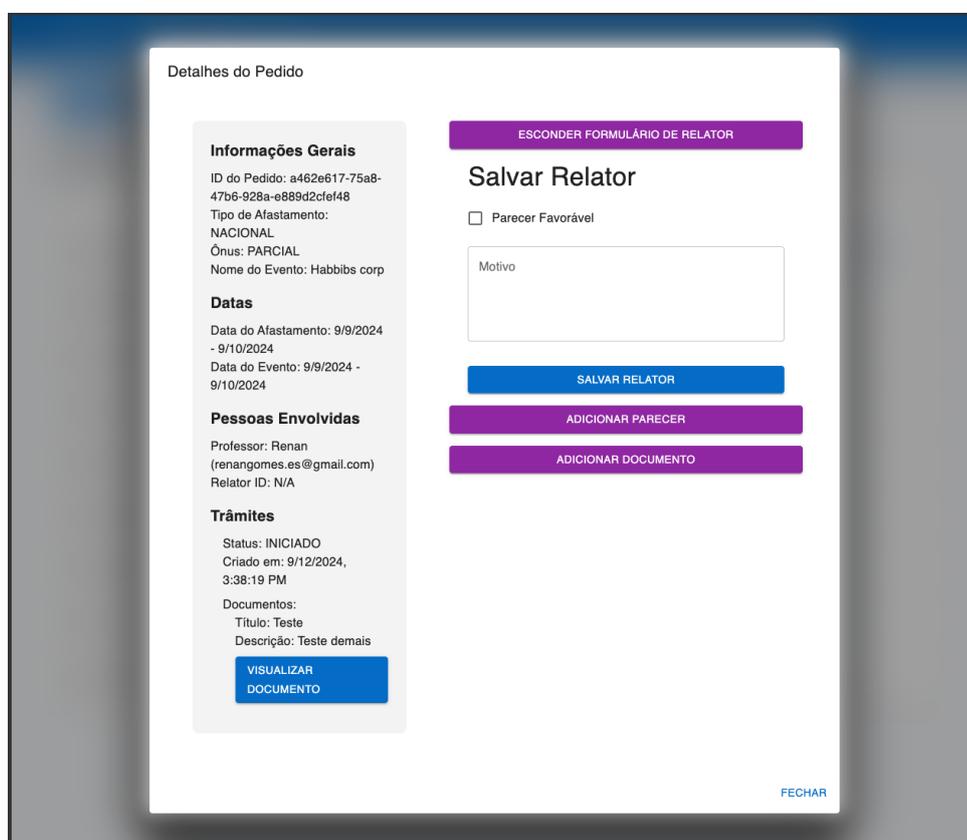


Figura 19 – Tela de formulário para designação de relator

Detalhes do Pedido

Informações Gerais
 ID do Pedido: a462e617-75a8-47b6-928a-e889d2cfe48
 Tipo de Afastamento: NACIONAL
 Ônus: PARCIAL
 Nome do Evento: Habbibs corp

Datas
 Data do Afastamento: 9/9/2024 - 9/10/2024
 Data do Evento: 9/9/2024 - 9/10/2024

Pessoas Envolvidas
 Professor: Renan (renangomes.es@gmail.com)
 Relator ID: N/A

Trâmites
 Status: INICIADO
 Criado em: 9/12/2024, 3:38:19 PM
 Documentos:
 Título: Teste
 Descrição: Teste demais

ADICIONAR RELATOR
ESCONDER FORMULÁRIO DE PARECER

Salvar Parecer
 Parecer Favorável
 Motivo
SALVAR PARECER
ADICIONAR DOCUMENTO
VISUALIZAR DOCUMENTO
 FECHAR

Figura 20 – Tela de formulário para emissão de parecer

Detalhes do Pedido

Informações Gerais
 ID do Pedido: a462e617-75a8-47b6-928a-e889d2cfe48
 Tipo de Afastamento: NACIONAL
 Ônus: PARCIAL
 Nome do Evento: Habbibs corp

Datas
 Data do Afastamento: 9/9/2024 - 9/10/2024
 Data do Evento: 9/9/2024 - 9/10/2024

Pessoas Envolvidas
 Professor: Renan (renangomes.es@gmail.com)
 Relator ID: N/A

Trâmites
 Status: INICIADO
 Criado em: 9/12/2024, 3:38:19 PM
 Documentos:
 Título: Teste
 Descrição: Teste demais

ADICIONAR RELATOR
ADICIONAR PARECER
ESCONDER FORMULÁRIO DE DOCUMENTO

Upload de Documento
 Título do Documento
 Descrição do Documento
SELECIONE O ARQUIVO
ENVIAR DOCUMENTO
VISUALIZAR DOCUMENTO
 FECHAR

Figura 21 – Tela de upload de documentos relacionados ao pedido de afastamento

O modelo de navegação de usuários, ilustrado na Figura 13, descreve como os componentes interagem no processo de gerenciamento e cadastro de usuários. A seguir, descrevemos cada uma das telas relacionadas a esse modelo de navegação:

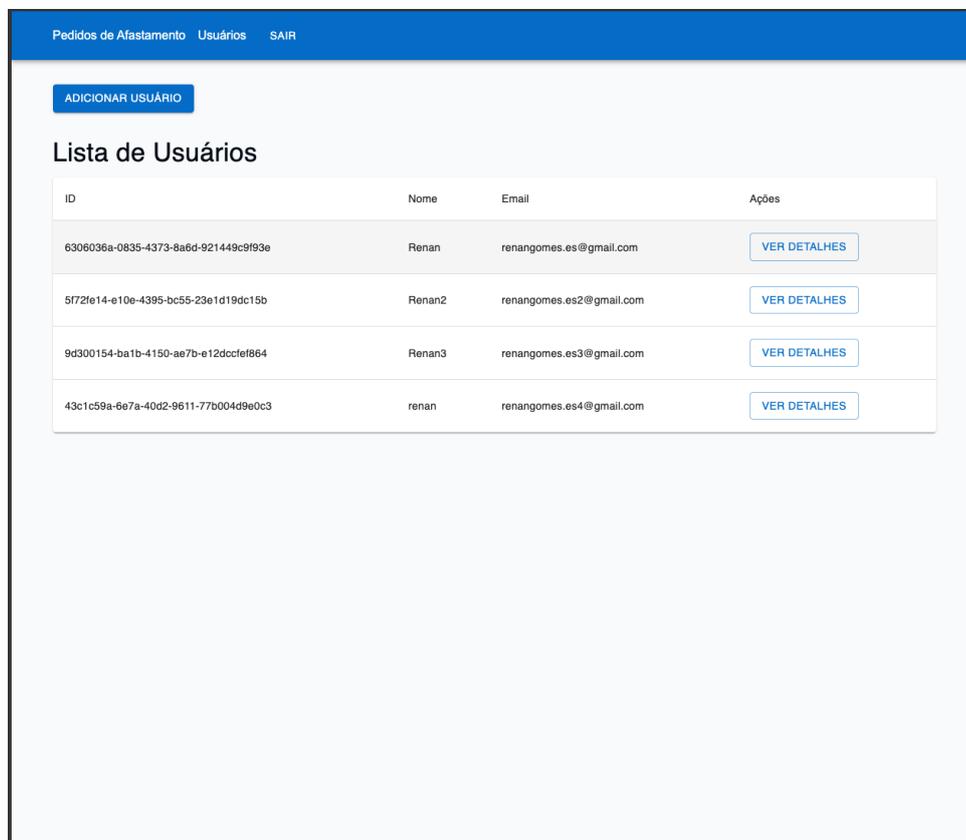


Figura 22 – Tela de lista de usuários

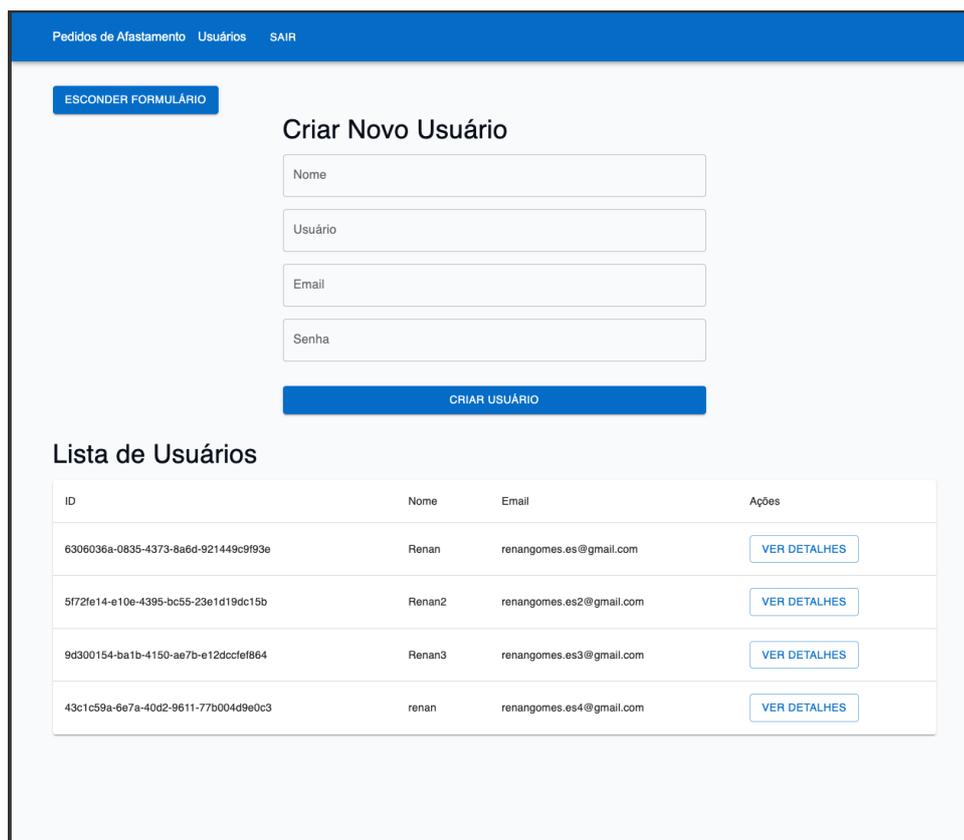
A Figura 22 apresenta a *tela de lista de usuários*, onde o usuário pode visualizar todos os usuários cadastrados no sistema. Nessa tela, é possível selecionar um usuário para visualizar suas informações detalhadas ou iniciar o cadastro de um novo usuário por meio do botão de criação.

A Figura 23 mostra a *tela de formulário para cadastro de usuário*, onde é possível inserir dados como nome, email, e outros detalhes relevantes para registrar um novo usuário no sistema. Esse formulário é essencial para adicionar novos usuários à plataforma.

Na Figura 24, temos a *tela de informações do usuário*, que exibe detalhes específicos de um usuário já cadastrado. Essas informações podem incluir dados pessoais, relações de parentesco, cargo, e outros atributos importantes para o gerenciamento do usuário no sistema.

A Figura 25 apresenta a *tela de informações do usuário chefe*, onde são exibidos dados adicionais e detalhados sobre o usuário que ocupa o cargo de chefe de departamento.

A Figura 26 ilustra a *tela de edição de informações do usuário*, onde o administrador ou responsável pode modificar dados de um usuário existente.



The screenshot displays a web interface for user management. At the top, there is a navigation bar with links for 'Pedidos de Afastamento', 'Usuários', and 'SAIR'. Below the navigation bar, there is a button labeled 'ESCONDER FORMULÁRIO'. The main content area is titled 'Criar Novo Usuário' and contains a registration form with four input fields: 'Nome', 'Usuário', 'Email', and 'Senha'. Below the form is a blue button labeled 'CRIAR USUÁRIO'. Underneath the form, there is a section titled 'Lista de Usuários' which contains a table with four columns: 'ID', 'Nome', 'Email', and 'Ações'. The table lists four users with their respective IDs, names, and email addresses, each with a 'VER DETALHES' button in the 'Ações' column.

ID	Nome	Email	Ações
6306036a-0835-4373-8a6d-921449c9f93e	Renan	renangomes.es@gmail.com	VER DETALHES
5f72fe14-e10e-4395-bc55-23e1d19dc15b	Renan2	renangomes.es2@gmail.com	VER DETALHES
9d300154-ba1b-4150-ae7b-e12dccfe6864	Renan3	renangomes.es3@gmail.com	VER DETALHES
43c1c59a-6e7a-40d2-9611-77b004d9e0c3	renan	renangomes.es4@gmail.com	VER DETALHES

Figura 23 – Tela de formulário para cadastro de usuário

Por fim, a Figura 27 mostra a *tela de formulário para salvar parentesco entre usuários*, onde é possível registrar relações de parentesco entre dois usuários do sistema. Esse formulário é essencial em contextos onde é necessário indicar vínculos familiares ou associativos entre os usuários cadastrados.

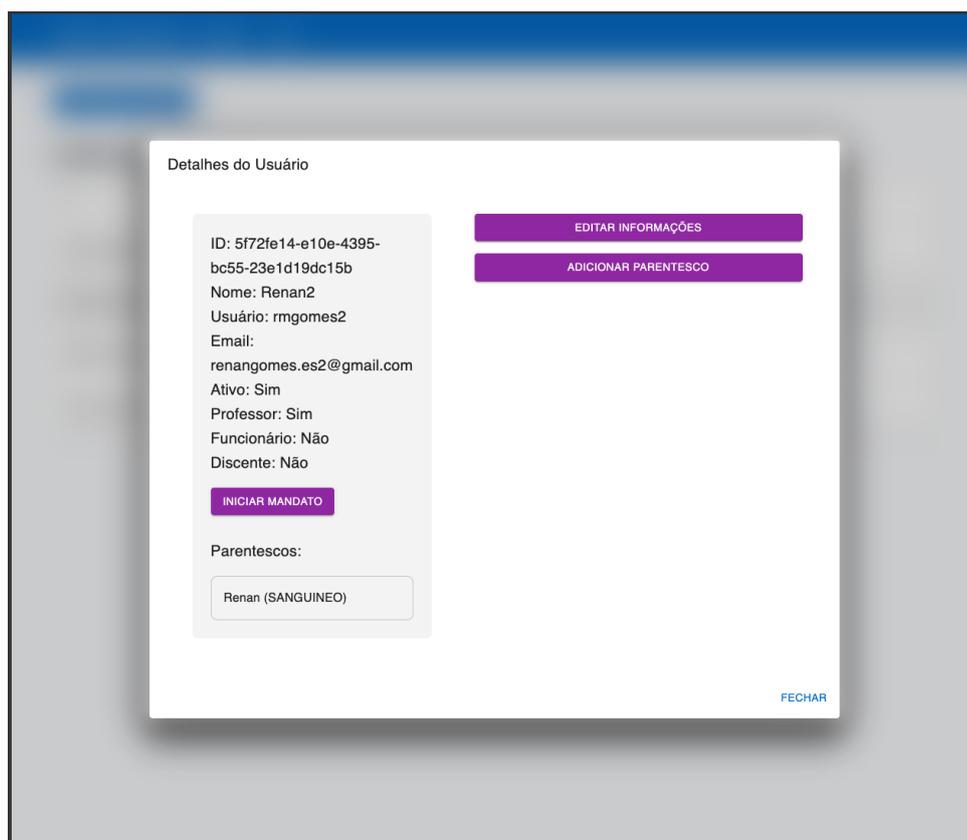


Figura 24 – Tela de informações do usuário

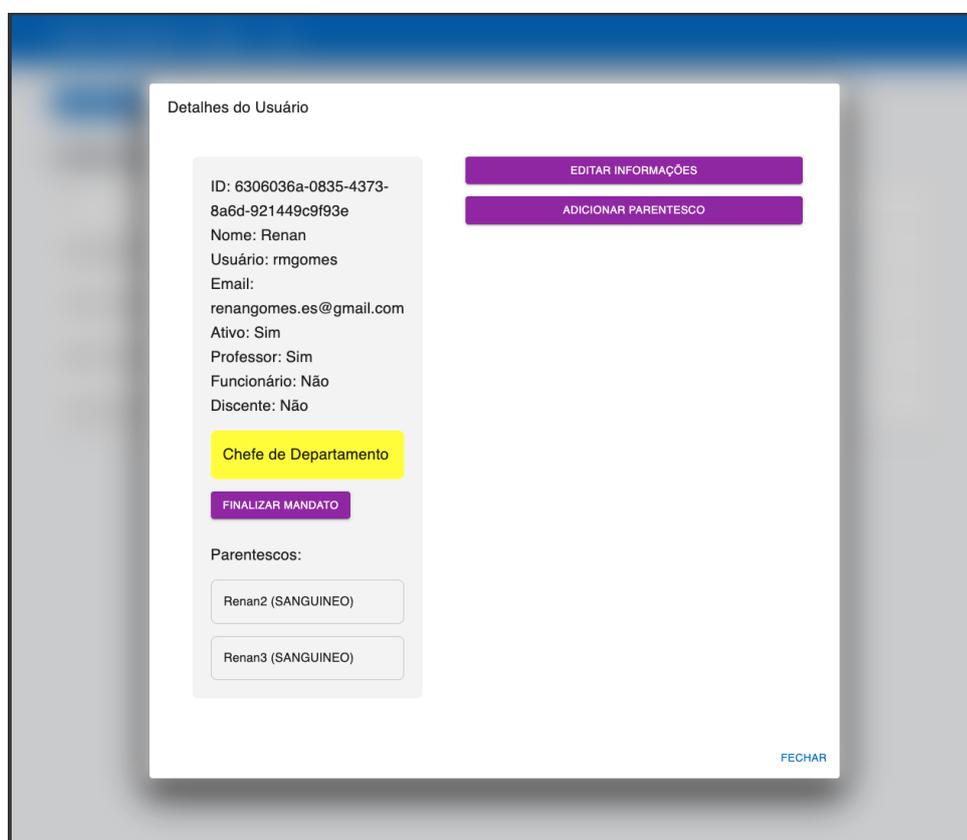


Figura 25 – Tela de informações do usuário chefe

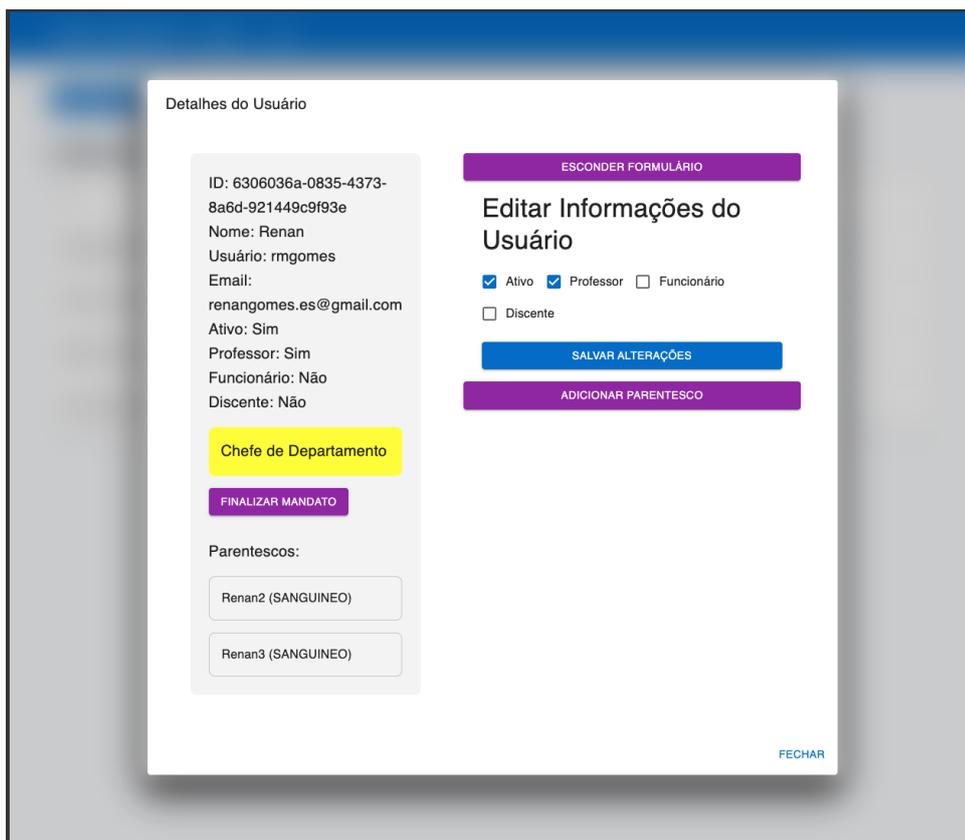


Figura 26 – Tela de edição de informações do usuário

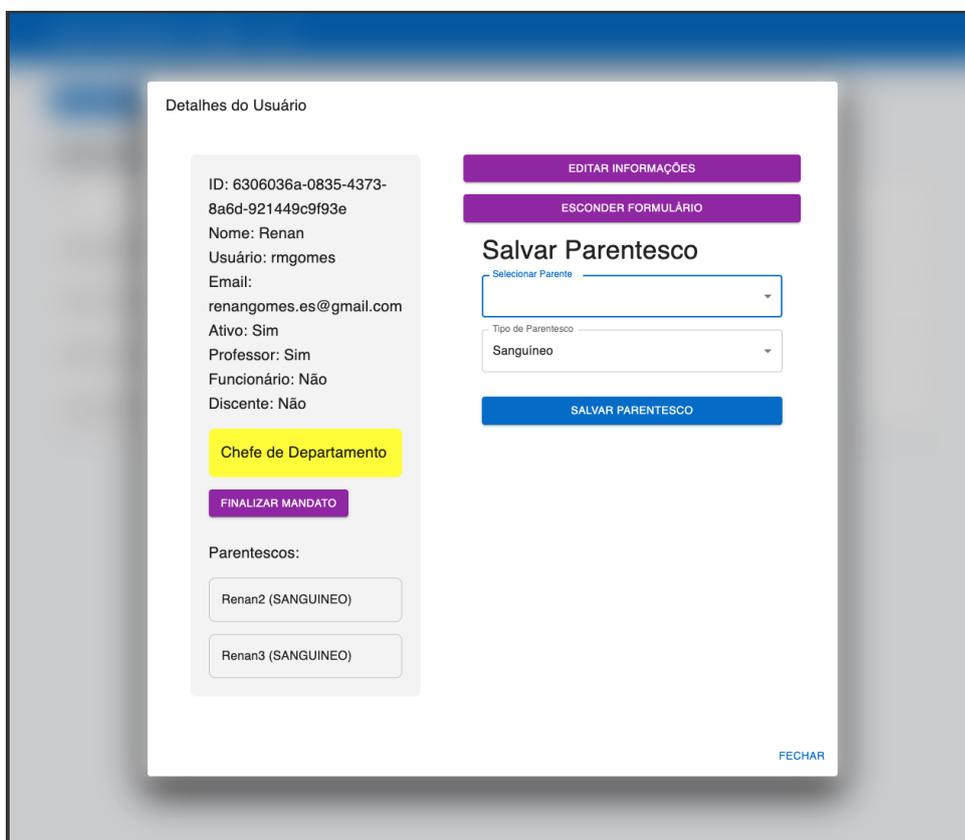


Figura 27 – Tela de formulário para salvar parentesco entre usuários

4 Conclusão

Neste trabalho, foi realizada a implementação de um sistema de controle de afastamento de professores (SCAP) utilizando o *Next.js* para o frontend e o *Spring Boot* para o backend, seguindo o método *FrameWeb* como referência principal para a organização e documentação arquitetural. Ao longo do desenvolvimento, o uso do *FrameWeb* mostrou-se consideravelmente vantajoso, pois forneceu uma base sólida de documentação arquitetural para guiar cada etapa do processo. Estando habituado a trabalhar com sistemas em que a documentação é limitada a ferramentas como o Miro ou cards refinados no Jira, pude notar como o *FrameWeb* proporciona uma clareza muito maior. Seus modelos arquiteturais permitem uma visão global e detalhada do sistema, facilitando a identificação e o isolamento de cada parte a ser desenvolvida.

Uma das principais contribuições deste trabalho foi a oportunidade de utilizar um método que ajuda a pensar de maneira estruturada e isolada sobre cada camada do sistema. A segmentação clara das partes do sistema proposta pelo *FrameWeb* — como a camada de apresentação, a lógica de negócio e a persistência — não apenas simplificou o desenvolvimento, mas também melhorou a comunicação entre as diferentes áreas do projeto, reduzindo ambiguidades e aumentando a consistência no processo de desenvolvimento.

Todos os objetivos estabelecidos foram totalmente alcançados. O primeiro objetivo específico, que consistia em compreender e analisar o método *FrameWeb* (SOUZA, 2007; SOUZA, 2020a), foi atingido com uma avaliação detalhada de suas principais características e da forma como ele organiza o desenvolvimento de projetos de sistemas Web. O segundo objetivo, que era projetar e implementar o sistema SCAP integrando os *frameworks* *Spring Boot* e *Next.js* utilizando o *FrameWeb*, também foi cumprido, permitindo a construção eficaz do sistema. Por fim, o terceiro objetivo, que visava contribuir para elaboração da percepção no uso do método *FrameWeb* utilizando um conjunto de *frameworks* diferente dos já aplicados, foi alcançado com sucesso, fornecendo uma análise prática do método em um novo contexto tecnológico.

4.1 Percepções do Desenvolvedor acerca do FrameWeb

O desenvolvimento deste trabalho trouxe valiosas lições e experiência no uso de metodologias estruturadas para o desenvolvimento de sistemas Web. Uma importante lição aprendida foi perceber que, mesmo com uma abordagem sólida como a proposta pelo *FrameWeb*, é consideravelmente desafiador criar uma arquitetura que cubra 100% do desenvolvimento sem ajustes. Houve situações em que precisei voltar ao modelo arquitetural para corrigir ou ajustar algo que havia sido feito incorretamente, evidenciando que o

desenvolvimento de software é um processo iterativo, mesmo com uma documentação bem estruturada.

Além disso, trabalhar com vários modelos simultaneamente pode, em certos momentos, causar confusão, já que é necessário manter a coerência entre as diferentes camadas e aspectos do sistema, o que pode se tornar um desafio ao longo do projeto. No entanto, a clareza proporcionada pelos modelos arquiteturais do *FrameWeb* ajudou a mitigar essas dificuldades, permitindo uma visão mais clara das interações entre os componentes do sistema.

No entanto, uma das limitações que enfrentei foi relacionada ao uso de editores de modelos arquiteturais. Atualmente, esses editores ainda apresentam dificuldades, especialmente por não existir um plugin dedicado e totalmente otimizado para o *FrameWeb*. Essa falta de suporte especializado torna o processo de modelagem mais complicado e, em algumas situações, menos intuitivo. Embora o método forneça um excelente direcionamento, o ferramental para modelar a arquitetura de forma eficiente ainda está aquém do esperado.

Em suma, a aplicação do *FrameWeb* no desenvolvimento deste sistema trouxe ganhos em termos de organização e clareza arquitetural, proporcionando uma base forte para o desenvolvimento de sistemas de informação. Apesar das limitações em relação ao ferramental e à necessidade de ajustes pontuais na arquitetura, o método se mostrou uma valiosa adição ao processo de desenvolvimento. Como próximas etapas, seria interessante explorar melhorias nas ferramentas de modelagem e continuar testando o *FrameWeb* em outros tipos de sistemas e arquiteturas, com o objetivo de expandir o uso dessa metodologia em cenários de desenvolvimento com tecnologias modernas.

4.2 Trabalhos Futuros

Com base nas dificuldades enfrentadas ao utilizar o *FrameWeb*, especialmente no que diz respeito aos editores de modelos, uma possível linha de trabalho futuro seria o desenvolvimento de uma versão dos modelos do *FrameWeb* em formato de código fonte. Esse conceito seria semelhante ao que o *Terraform* faz para a infraestrutura como código, simplificando a criação e modificação dos componentes arquiteturais. Um sistema de modelagem baseado em código teria várias vantagens, incluindo a capacidade de versionar, compartilhar e colaborar em grupo, facilitando o trabalho em partes isoladas e permitindo maior modularização.

Essa abordagem permitiria que os projetistas gerassem artefatos versionáveis, além de melhorar a integração com sistemas de controle de versão, como o Git. Isso não apenas aumentaria a capacidade de colaboração em projetos complexos, mas também permitiria que os modelos fossem modularizados, com cada parte da arquitetura sendo trabalhada de forma independente e posteriormente combinada em um sistema coeso.

Outro benefício seria a possibilidade de converter os modelos *FrameWeb* mais facilmente em código executável, facilitando o desenvolvimento e a integração com linguagens de programação. Além disso, essa abordagem abriria caminho para o desenvolvimento de ferramentas ou plugins exclusivos para o *FrameWeb*, proporcionando um suporte técnico mais robusto.

Além das melhorias no suporte ao *FrameWeb*, outra linha de pesquisa e desenvolvimento futuro seria a continuidade dos estudos que já exploram a aplicação do *FrameWeb* com diferentes *frameworks*. Trabalhos anteriores já demonstraram sua aplicabilidade em conjunto com *frameworks* como *Spring Boot*, *Angular* e *VRaptor*. A continuidade desses estudos seria valiosa para investigar ainda mais o potencial do *FrameWeb* em ambientes modernos, especialmente com a evolução contínua dos *frameworks* e ferramentas de desenvolvimento Web.

Trabalhos futuros poderiam se concentrar em expandir esses experimentos para incluir tecnologias de infraestrutura moderna, como *Lambda Functions*, serviços de mensageria (como *Kafka* ou *RabbitMQ*), e outras soluções nativas da nuvem amplamente adotadas em sistemas Web contemporâneos. A exploração dessas tecnologias no contexto do *FrameWeb* ajudaria a validar sua eficácia em sistemas de grande escala, que demandam escalabilidade, modularidade e alta disponibilidade, contribuindo para consolidar o *FrameWeb* como uma metodologia robusta para o desenvolvimento de arquiteturas orientadas a eventos e sistemas nativos em nuvem.

Além das direções acadêmicas e de pesquisa, o *FrameWeb* também tem grande potencial para ser explorado comercialmente. Seu enfoque em modularidade, reutilização de componentes e clareza arquitetural pode acelerar significativamente o processo de desenvolvimento de software, reduzindo o tempo necessário para o design e implementação de sistemas confiáveis. Empresas que buscam agilidade no desenvolvimento de software e que precisam de sistemas bem estruturados poderiam adotar o *FrameWeb* como uma metodologia padrão, aproveitando sua capacidade de facilitar a organização e documentação dos projetos. Isso não apenas ajudaria na entrega mais rápida de produtos, mas também garantiria a escalabilidade e manutenibilidade dos sistemas a longo prazo, tornando-o uma solução atrativa para o mercado corporativo.

Referências

- ALUR, D.; CRUPI, J.; MALKS, D. *Core J2EE Patterns*. [S.l.]: S.l.: s.n., 2003. ISBN 0131422464. Citado na página 24.
- CONALLEN, J. *Building Web Applications with UML*. 2. ed. [S.l.]: Addison-Wesley, 2002. ISBN 0785342730388. Citado na página 24.
- DUARTE, B. B. Aplicação do método frameweb no desenvolvimento de um sistema de informação na plataforma java ee 7. Trabalho de Conclusão de Curso. 2014. Citado 3 vezes nas páginas 13, 14 e 30.
- FALBO, R. d. A. *Engenharia de Software*. 2014. 178 p. [s.n.]. Disponível em: <http://www.inf.ufes.br/~jssalamon/wp-content/uploads/disciplinas/engsoft/Notas_Aula_Engenharia_Software_Falbo_2014.pdf>. Citado 5 vezes nas páginas 8, 19, 20, 21 e 22.
- FALBO, R. d. A. *Engenharia de Requisitos*. 2017. 178 p. [s.n.]. Disponível em: <http://www.inf.ufes.br/~vitorsouza/falbo/Notas_Aula_Engenharia_Requisitos_Falbo_2017.pdf>. Citado 3 vezes nas páginas 8, 17 e 18.
- FALBO, R. d. 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 19.
- FOWLER, M. *Patterns of Enterprise Application Architecture*. [S.l.]: Addison-Wesley, 2002. ISBN 0321127420. Citado 3 vezes nas páginas 8, 24 e 33.
- KOCH, N. e. a. 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.
- KOSCIANSKI, A.; SOARES, M. S. *Qualidade de Software*. [S.l.]: Editora Novatec, 2006. Citado na página 20.
- MCGREGOR, J. D.; SYKES, D. A. *A Practical Guide to Testing Object-Oriented Software*. [S.l.]: Addison-Wesley, 2001. Citado na página 20.
- MURUGESAN, S. et al. Web engineering: A new discipline for development of web-based systems. In: *Web Engineering*. [s.n.], 2001. v. 2016, p. 3–13. Disponível em: <https://link.springer.com/chapter/10.1007/3-540-45144-7_2>. Citado na página 20.
- 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: <https://link.springer.com/chapter/10.1007/3-540-45144-7_26>. Citado na página 20.
- PANDOLFI, D.; MELOTTI, E. *Ciclo de Desenvolvimento de Projetos Web*. 2006. Citado na página 16. Citado na página 21.
- PRADO, R. C. d. Aplicação do método frameweb no desenvolvimento de um sistema de

informação utilizando o framework vraptor 4. Trabalho de Conclusão de Curso. 2015. Citado 2 vezes nas páginas 14 e 30.

PRESSMAN, R. S. *Software Engineering: A Practitioner's Approach*. 6. ed. [S.l.]: McGraw Hill, 2005. ISBN 007301933X. Citado na página 21.

PRESSMAN, R. S. *Engenharia de Software - Uma Abordagem Profissional*. 7. ed. S.l.: McGraw-Hill, 2011. ISBN 9788563308337. Citado 2 vezes nas páginas 17 e 20.

PRESSMAN, R. S.; LOWE, D. *Engenharia Web. Um Enfoque Profissional*. 1. ed. [S.l.]: LTC, 2009. ISBN 8521616961. Citado na página 21.

SHANNON, B. *Java™ 2 Platform Enterprise Edition Specification*. [S.l.]: Sun Microsystems, 2003. v. 1.4. Citado na página 25.

SOMMERVILLE, I. *Engenharia de Software*. 9th. ed. [S.l.]: Pearson Universidades, 2011. 568 p. ISBN 978-85-7936-108-1. Citado 3 vezes nas páginas 17, 18 e 19.

SOUZA, V. E. S. *FrameWeb: um Método baseado em Frameworks para o Projeto de Sistemas de Informação Web*. Dissertação (Mestrado) — <https://nemo.inf.ufes.br/wp-content/papercite-data/pdf/souza-masterthesis07.pdf>, 2007. Citado 9 vezes nas páginas 8, 13, 14, 22, 23, 24, 25, 26 e 51.

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*. Vitória, ES, Brazil: NEMO, 2020. p. 100–124. ISBN 9781393963035. Disponível em: <<http://purl.org/nemo/celebratingfalbo>>. Citado 6 vezes nas páginas 13, 14, 22, 26, 33 e 51.

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 na página 33.

Apêndices



Documento de Projeto de Sistema

**Aplicação do método FrameWeb no
desenvolvimento de um sistema de informação
baseado na Web utilizando os *frameworks*
Spring Boot e NextJS**

Vitória, ES

2024

Registro de Alterações:

Versão	Responsável	Data	Alterações
1.0	Renan Moreira Gomes	04/10/2023	Versão inicial.
1.1	Renan Moreira Gomes	01/02/2024	Plataforma de Desenvolvimento.
1.2	Renan Moreira Gomes	02/02/2024	Plataforma de Desenvolvimento (versões).
2.0	Renan Moreira Gomes	03/02/2024	Requisitos não funcionais.
3.0	Renan Moreira Gomes	15/02/2024	Arquitetura de Software.
4.0	Renan Moreira Gomes	03/02/2024	Início da criação dos modelos do FrameWeb.
4.1	Renan Moreira Gomes	10/02/2024	Modelos de Domínio.
4.2	Renan Moreira Gomes	02/03/2024	Modelos de Aplicação.
4.3	Renan Moreira Gomes	15/03/2024	Modelos de Persistência.
4.4	Renan Moreira Gomes	20/04/2024	Modelos de Navegação.
4.5	Renan Moreira Gomes	07/05/2024	Textos para explicar os modelos do FrameWeb.
5.0	Renan Moreira Gomes	02/06/2024	Ajustes nos modelos do Frameweb usando Framework Editor.
5.1	Renan Moreira Gomes	10/06/2024	Finalização dos ajustes nos modelos do Frameweb

1 Introdução

Este documento apresenta o projeto (*design*) do sistema SCAP, desenvolvido com base nos *frameworks* Spring Boot e NextJS.

O Sistema de Controle de Afastamentos de Professores (SCAP) nasceu com o objetivo de facilitar o processo de solicitação e acompanhamento dos afastamentos para professores e secretários, simplificando desde a criação até a análise e armazenamento dos pedidos.

No Departamento de Informática da Universidade Federal do Espírito Santo (UFES), os pedidos de licença para participação em eventos, tanto nacionais quanto internacionais, seguem um processo específico de avaliação e aprovação. As solicitações são inicialmente examinadas pelos professores do departamento e, dependendo do caso, também pelo Conselho Departamental do Centro Tecnológico e pela Pró-reitoria de Pesquisa e Pós-Graduação. Uma vez que o pedido recebe aprovação em todas as instâncias necessárias, o afastamento é oficialmente autorizado. A Especificação de Requisitos completa do SCAP pode ser vista em (PRADO, 2015).

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
Java	17	Linguagem de programação orientada a objetos e independente de plataforma.	Escrita do código-fonte das classes que compõem o sistema.
Spring Boot	3.1.4	Framework para facilitar o desenvolvimento de aplicações Spring.	Redução da complexidade do desenvolvimento, implantação e gerenciamento de aplicações Web.

Tecnologia	Versão	Descrição	Propósito
Spring Data JPA	3.1.4	API para persistência de dados por meio de mapeamento objeto/-relacional.	Persistência dos objetos de domínio sem necessidade de escrita dos comandos SQL.
Hibernate ORM	6.3	Framework para mapeamento objeto-relacional.	Mapeamento dos objetos Java para tabelas de banco de dados e vice-versa.
PostgreSQL	14.x	Sistema Gerenciador de Banco de Dados Relacional de código aberto.	Armazenamento dos dados manipulados pela ferramenta.
Next.js	12.x	Framework de geração de sites estáticos para aplicativos da Web baseados em React.	Criação do <i>frontend</i> da aplicação e gerenciamento de rotas.
React	18.x	Biblioteca JavaScript para construção de interfaces de usuário.	Desenvolvimento da interface do usuário no <i>frontend</i> .
TypeScript	5.x	Linguagem de programação que estende JavaScript adicionando tipos estáticos.	Escrita do código-fonte do <i>frontend</i> , proporcionando um desenvolvimento mais seguro e eficiente.
Material-UI	5.x	Biblioteca de componentes React para um desenvolvimento ágil e fácil de interfaces de usuário.	Criação de interfaces de usuário consistentes e responsivas com base no <i>Material Design</i> .
Tailwind CSS	3.x	<i>Framework</i> de utilidades para construção de interfaces de usuário.	Desenvolvimento rápido e responsivo de interfaces de usuário com classes de utilidade.

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
Visual Studio Code	latest	Editor de código-fonte leve, mas poderoso, que pode ser estendido com funcionalidades adicionais através de extensões.	Desenvolvimento e edição de código-fonte do projeto.
Overleaf	latest	Editor de \LaTeX online.	Escrita da documentação do projeto.
DBeaver	21.3	Ferramenta de administração de banco de dados com suporte a diversos SGBDs.	Visualização, gerenciamento e depuração do banco de dados.
Visual Paradigm Community Edition (com plugin FrameWeb)	17.1	Ferramenta com suporte ao método FrameWeb através de um <i>plugin</i> .	Criação dos modelos de Entidades, Aplicação, Persistência e Navegação.
Git	2.39	Sistema de controle de versão distribuído.	Gerenciamento e versionamento do código-fonte do projeto.
Postman	10.18	Ferramenta para teste e documentação de APIs.	Teste das APIs desenvolvidas no backend.

Tecnologia	Versão	Descrição	Propósito
Docker	24.0.5	Plataforma para desenvolvimento, envio e execução de aplicações em contêineres.	Isolamento, distribuição e execução do ambiente de desenvolvimento e produção.

3 Requisitos Não Funcionais

A Tabela 3 apresenta a especificação dos requisitos não funcionais identificados no Documento de Especificação de Requisitos, os quais foram considerados condutores da arquitetura.

Tabela 3 – Especificação de Requisitos Não Funcionais.

RNF-1 – O sistema deve apresentar uma interface de usuário intuitiva, garantindo eficiência mesmo sem uso constante.	
Categoria:	Usabilidade
Tática / Tratamento:	Utilizar design intuitivo e padronizado, com ícones claros e fluxo de trabalho lógico para facilitar a navegação e o uso.
Medida:	Conduzir um teste de usabilidade com um usuário novato, onde ele deve completar tarefas essenciais sem ajuda.
Critério de Aceitação:	O usuário novato consegue completar tarefas essenciais na ferramenta sem assistência em menos de 30 minutos.
RNF-2 – O sistema deve implementar um mecanismo de gerenciamento de acesso às suas funcionalidades.	
Categoria:	Segurança
Tática / Tratamento:	Implementar um sistema de gerenciamento de usuários com diferentes níveis de permissões, garantindo que cada usuário acesse apenas as funcionalidades adequadas ao seu nível.
Medida:	Realizar auditorias regulares no sistema de controle de acesso, verificando se os níveis de permissão estão sendo respeitados.
Critério de Aceitação:	Todos os usuários devem ter acesso apenas às funcionalidades permitidas pelo seu nível de permissão, sem nenhuma ocorrência de acesso não autorizado, conforme verificado nas auditorias.
RNF-3 – A ferramenta deve ser acessível como uma aplicação Web, compatível com os navegadores mais populares do mercado.	
Categoria:	Portabilidade
Tática / Tratamento:	Desenvolver a aplicação utilizando tecnologias Web padrão e responsivas, garantindo compatibilidade com os principais navegadores do mercado.
Medida:	Realizar testes de compatibilidade e desempenho em diferentes navegadores, incluindo suas versões mais recentes.

Critério de Aceitação:	A aplicação deve funcionar corretamente e de forma consistente nos últimos versões dos principais navegadores, como Chrome, Firefox, Safari e Edge, sem falhas de renderização ou funcionalidades comprometidas.
------------------------	--

RNF-4 – O sistema deve possuir uma estrutura que facilite a manutenção, permitindo a inclusão de novas funcionalidades ou adaptações para necessidades específicas.	
Categoria:	Manutenibilidade
Tática / Tratamento:	Utilizar padrões de código claros e documentação interna detalhada, seguindo as melhores práticas do <i>framework</i> escolhido, para facilitar a manutenção e expansão do sistema.
Medida:	Avaliar periodicamente o tempo e os recursos necessários para implementar novas funcionalidades ou adaptações, assegurando que estejam dentro de um limite aceitável.
Critério de Aceitação:	Novas funcionalidades ou adaptações específicas devem ser implementadas em um tempo médio pré-definido, com uma margem de erro mínima, sem comprometer a estabilidade do sistema.

RNF-5 – O desenvolvimento do sistema deve empregar os <i>frameworks</i> Spring Boot e Next.JS, aderindo ao método FrameWeb.	
Categoria:	-
Tática / Tratamento:	Implementar o sistema utilizando Spring Boot e Next.js, aplicando as diretrizes do método FrameWeb para estruturar o projeto de forma eficiente.
Medida:	Conduzir revisões de código e auditorias técnicas regulares para garantir a adesão consistente aos <i>frameworks</i> Spring Boot e Next.js, bem como à metodologia FrameWeb.
Critério de Aceitação:	O desenvolvimento do sistema deve ser completado com sucesso utilizando apenas Spring Boot e Next.js, e a documentação do projeto deve evidenciar a aplicação efetiva do método FrameWeb em todas as etapas do desenvolvimento.

RNF-6 – O sistema deve ser desenvolvido focando na reutilização de componentes, tanto na perspectiva de desenvolver utilizando componentes reutilizáveis quanto na criação de componentes para reuso futuro.	
Categoria:	Manutenibilidade
Tática / Tratamento:	Adotar uma abordagem modular no desenvolvimento com Spring Boot e Next.js, criando componentes reutilizáveis e mantendo um repositório de componentes para facilitar tanto o desenvolvimento com reuso quanto o desenvolvimento para reuso.
Medida:	Monitorar a frequência de reutilização de componentes em diversos projetos ou módulos, e realizar revisões para assegurar a modularidade e a qualidade dos componentes.
Critério de Aceitação:	Um percentual significativo do código (por exemplo, 60%) deve ser composto por componentes reutilizáveis, e estes componentes devem ser implementados em múltiplas partes do sistema ou em projetos futuros, demonstrando efetiva reutilização.

4 Arquitetura de Software

A Figura 1 mostra a arquitetura do sistema SCAP, desenvolvido com base nos *frameworks* Spring Boot e NextJS.

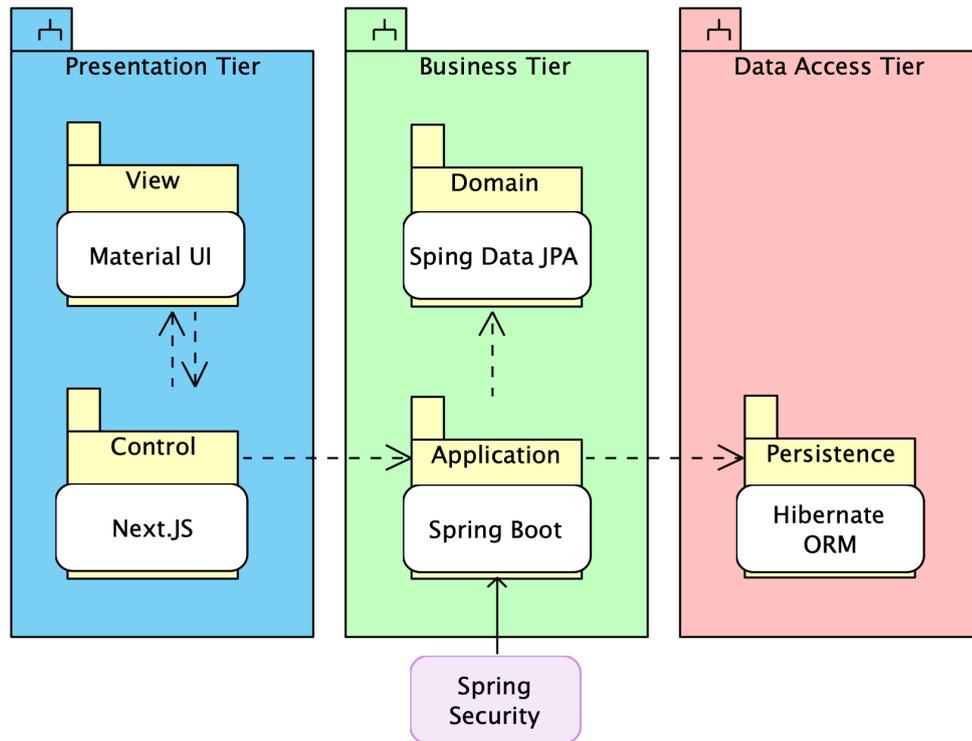


Figura 1 – Arquitetura de Software.

A arquitetura ilustrada na Figura 1 é estruturada em três camadas distintas: *Presentation Tier*, *Business Tier* e *Data Access Tier*. Esta configuração representa uma adaptação da arquitetura proposta pelo FrameWeb (SOUZA, 2020a), baseada no padrão *Service Layer* (FOWLER, 2002).

Camada de Apresentação: a interface do usuário é desenvolvida com *Material UI*, que oferece componentes visuais ricos e elegantes. *Next.js* é empregado como o *framework* de controle, facilitando a construção de aplicações Web do lado do servidor (*Server Side Rendering* – SSR) ou estáticas, além de gerenciar as interações do usuário e a comunicação com as camadas de negócio.

Camada de Negócios: *Spring Data JPA* é usado para a lógica de domínio e interação com o banco de dados, enquanto *Spring Boot* serve como a estrutura fundamental para o desenvolvimento da aplicação. Além disso, *Spring Security* é integrado para prover segurança robusta, cuidando de autenticação e proteção contra ameaças comuns.

Camada de Acesso a Dados: o *Hibernate* é utilizado para gerenciar a persistência dos dados, facilitando o mapeamento de objetos Java para tabelas de banco de dados e a execução eficiente das operações de banco de dados. Esta camada garante que os dados

sejam manipulados de forma segura e eficiente.

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, 2020b).

A Tabela 4 indica os *frameworks* presentes na arquitetura do sistema que se encaixam em cada uma das categorias de *frameworks* que 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	Next.JS
Injeção de Dependências	Spring Framework
Mapeamento Objeto/Relacional	Hibernate ORM
Segurança	Spring Security

5.1 Camada de Negócio

O cerne do modelo de domínio descrito na Figura 2 é a entidade **Usuário**, que se desdobra em três categorias principais: funcionário, representante discente e professor. Cada categoria serve a propósitos distintos dentro do sistema, refletindo os diferentes papéis na vida acadêmica.

Os professores, como uma das subclasses de **Usuário**, possuem atributos e relações particulares que destacam suas responsabilidades e conexões pessoais. Eles podem ter vínculos de parentesco com outros professores, categorizados em relações sanguíneas ou matrimoniais. Além disso, professores podem assumir o papel de **Chefe de Departamento**, com um mandato definido que tem início e fim claros, ressaltando a natureza temporária e rotativa da administração acadêmica.

A mobilidade acadêmica é representada através dos pedidos de afastamento, permitindo que professores se ausentem para participar de eventos, sejam eles nacionais ou internacionais. A análise desses pedidos, dependendo do tipo de afastamento, é feita por professores designados como relatores, que avaliam e emitem pareceres sobre a viabilidade e relevância dos afastamentos solicitados. Outros professores, por sua vez, podem ser chamados a oferecer seus pareceres sobre esses pedidos, criando uma camada de revisão e supervisão colegiada.

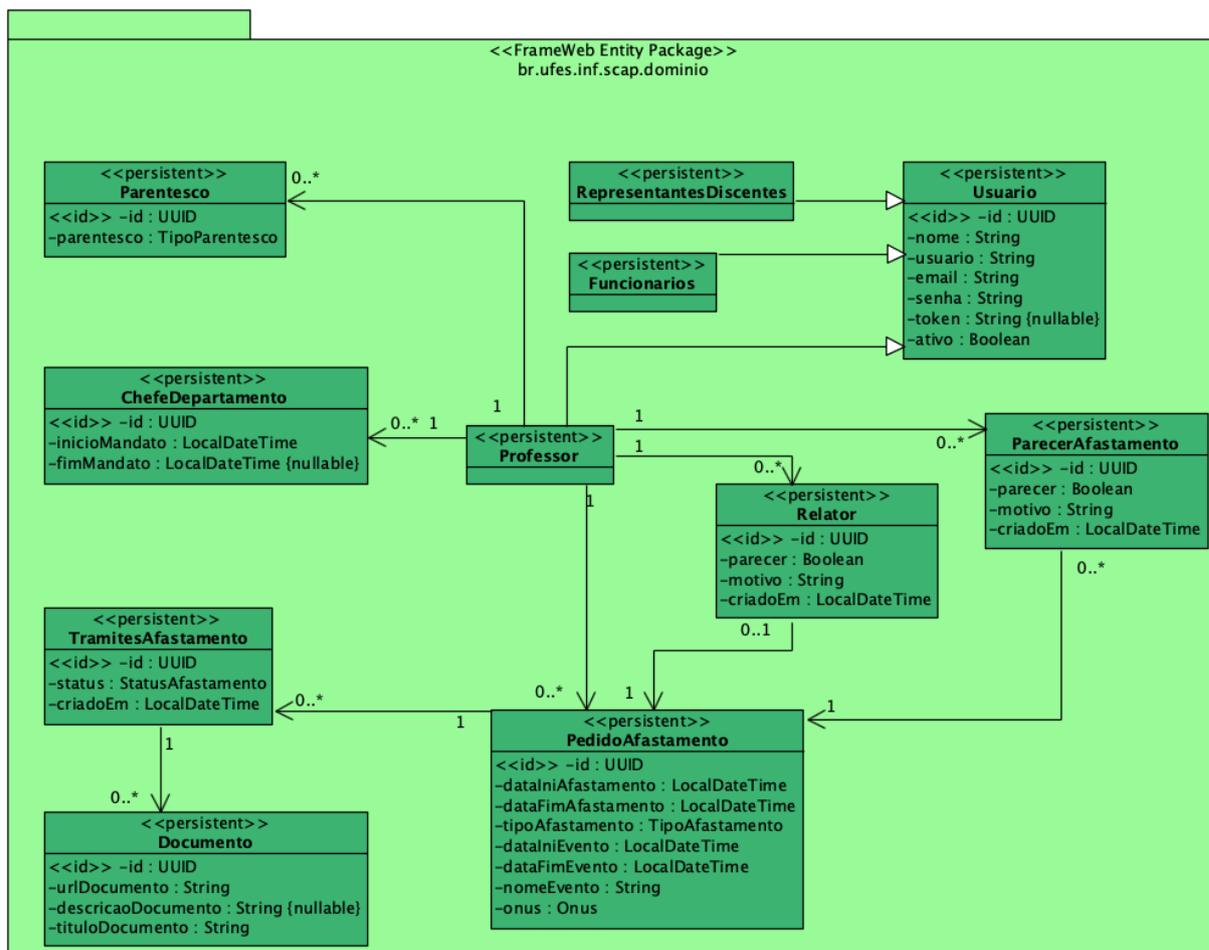


Figura 2 – Modelo de Domínio

O pedido de afastamento, como entidade, possui uma lista de trâmites que indicam o status atual do pedido, e cada status pode estar associado a uma lista de documentos requeridos. Esse mecanismo assegura que o processo de avaliação seja rigoroso e bem documentado, facilitando a gestão administrativa.

Importante destacar que os tipos enumeráveis do modelo, como os tipos de parentesco e os tipos de afastamento, estão claramente definidos na Figura 3. Este detalhamento garante que todas as nuances do modelo sejam facilmente acessíveis e compreensíveis, refletindo a complexidade e a dinâmica do ambiente acadêmico e garantindo que todos os aspectos da administração sejam meticulosamente considerados e geridos através do seu sistema.

O modelo de aplicação do subsistema de Usuário, descrito na Figura 4, é projetado para lidar com todas as operações relacionadas à criação e gerenciamento de perfis de usuários dentro da instituição. Isso inclui a capacidade de registrar novos usuários, seja como funcionários, representantes discentes ou professores, cada um com atributos e permissões específicas. Além disso, o subsistema facilita o registro de relações de parentesco entre professores e a gestão de mandatos para aqueles que ocupam posições de chefe de



Figura 3 – Modelo de Tipos do Domínio

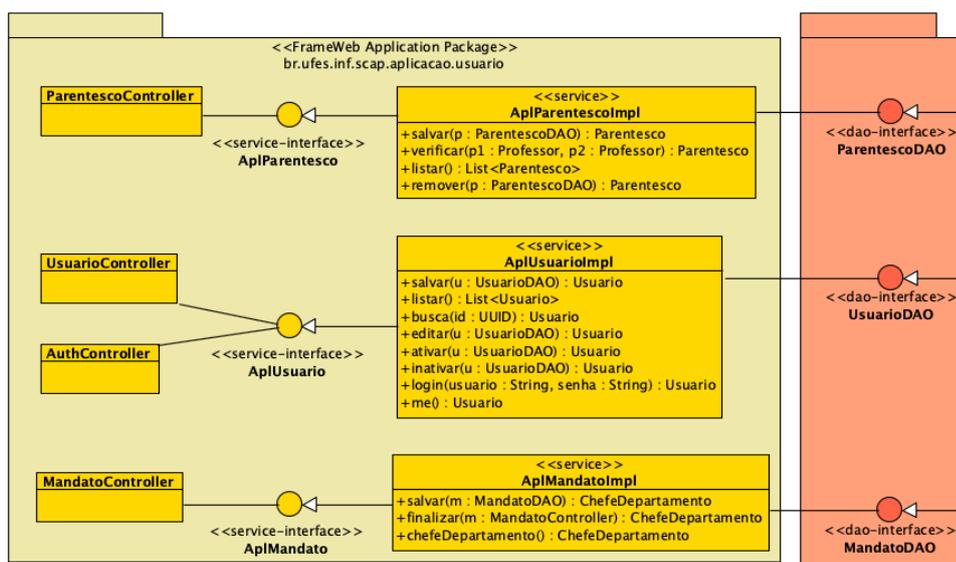


Figura 4 – Modelo de Aplicação do Subsistema Usuário

departamento.

Paralelamente, o modelo de aplicação do subsistema de Pedido de Afastamento, descrito na Figura 5 se concentra em automatizar e organizar o processo pelo qual os professores podem solicitar afastamentos. A abertura de novos pedidos, a adição de pareceres, tanto por parte de relatores quanto por outros professores autorizados a opinar sobre os pedidos e o registro metuculoso de documentos em cada etapa do processo são elementos chave deste subsistema. Esses processos garantem que cada pedido seja tratado com a devida atenção e rigor, assegurando a transparência e a eficácia na gestão dos afastamentos.

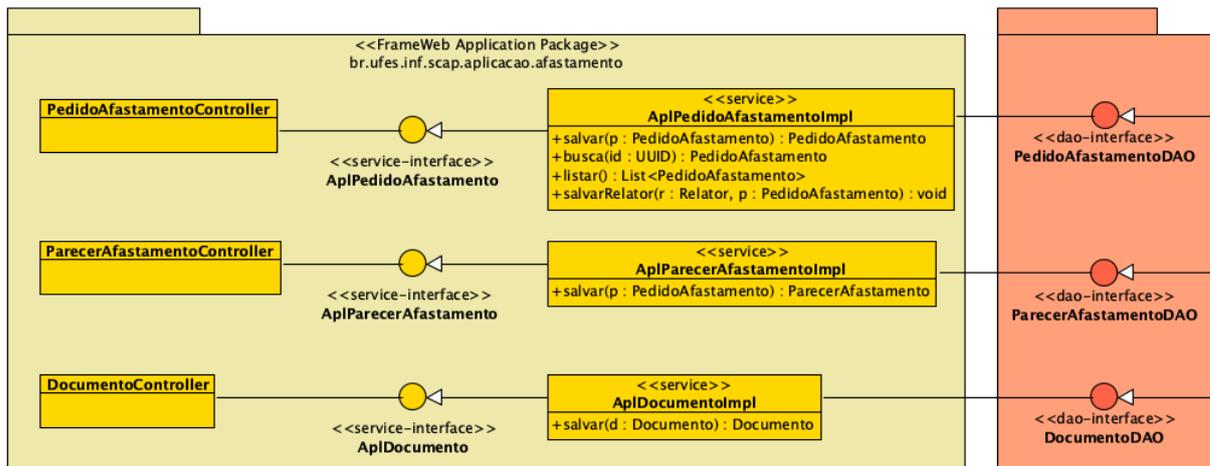


Figura 5 – Modelo de Aplicação do Subsistema Pedido de Afastamento

5.2 Camada de Acesso a Dados

Os modelos de persistência em ambos os subsistemas definem uma série de métodos especializados que são fundamentais para as operações diárias dentro da instituição. Por exemplo, no contexto do subsistema de usuários, descrito na Figura 6, métodos como a checagem de parentesco entre professores e a verificação de mandato de chefes de departamento são implementados para assegurar que as políticas internas e as normas de conformidade sejam rigorosamente seguidas. Esses métodos ajudam a manter a integridade das relações e posições dentro da instituição, evitando conflitos de interesse e garantindo a transparência.

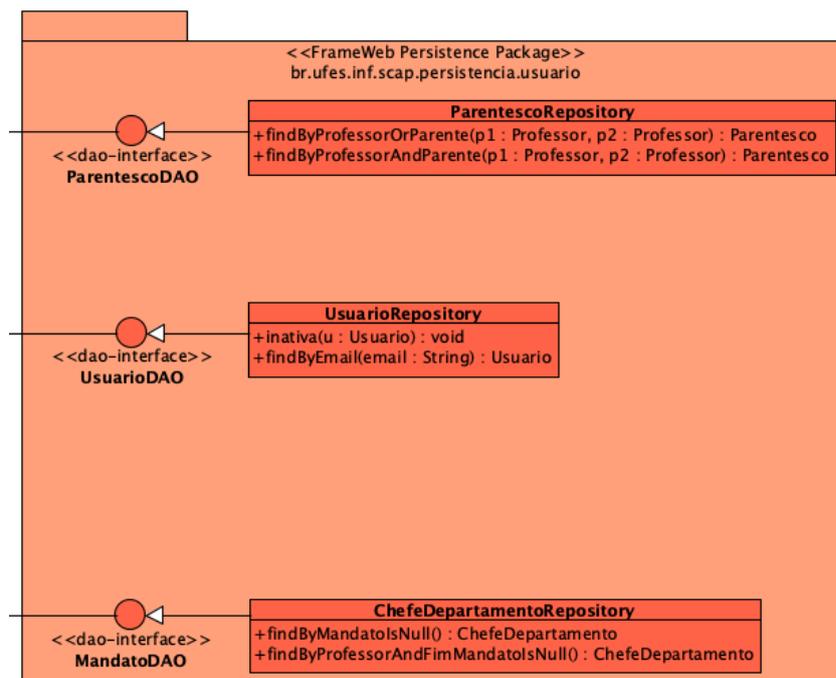


Figura 6 – Modelo de Persistência do Subsistema Usuário

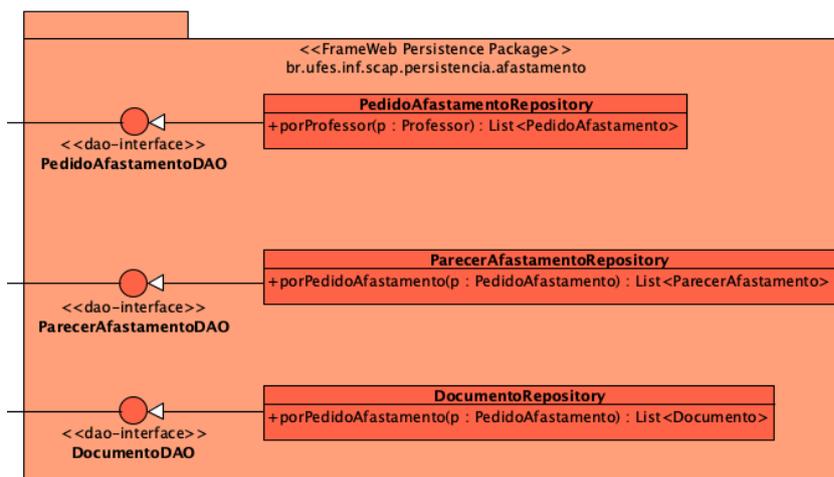


Figura 7 – Modelo de Persistência do Subsistema Pedido de Afastamento

Simultaneamente, no subsistema de pedido de afastamento, descrito na Figura 7, a funcionalidade é enriquecida com métodos para listar pedidos de afastamento por professor, permitindo uma visão clara do histórico e do status atual de solicitações de cada docente. Além disso, a listagem de pareceres por pedidos e o registro de trâmites são vitais para monitorar o progresso e as decisões relacionadas a cada pedido, enquanto a listagem de documentos por pedido assegura que todos os requisitos documentais sejam facilmente acessíveis.

Esses modelos de persistência herdam funcionalidades de *repository*, que incorpora as operações básicas de CRUD. Esta herança permite que os modelos de persistência mantenham uma base comum de operações de dados, ao mesmo tempo que expandem suas capacidades com métodos específicos para atender às necessidades únicas de cada subsistema.

5.3 Camada de Apresentação

Nos modelos de apresentação, há uma particularidade na nomeação dos pacotes. Esses pacotes são denominados de acordo com os diretórios do código-fonte do Next.js, que incorporam as convenções do React.

O modelo de navegação desautenticado, descrito na Figura 8 é fundamental para gerenciar o acesso inicial ao sistema. Este modelo controla o processo de login, verificando as credenciais do usuário e determinando se o acesso pode ser concedido. Caso as credenciais sejam inválidas ou o acesso seja negado, o usuário permanece na página de login. Se o login for bem-sucedido, o usuário é encaminhado para a área autenticada do sistema. Este modelo é a primeira camada de segurança e serve como um filtro para garantir que apenas usuários autorizados possam prosseguir para as funcionalidades internas do sistema.

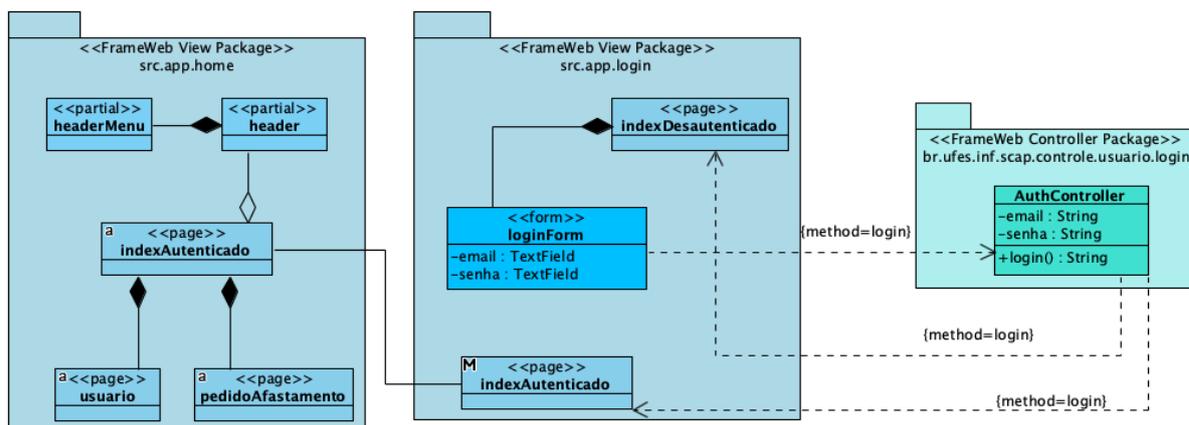


Figura 8 – Modelo de Navegação Desautenticado

Uma vez autenticado, o usuário ganha acesso a página inicial, que apresenta um menu que dá acesso às principais funcionalidades do sistema. Este modelo inclui links ou botões para as páginas de pedido de afastamento e a página de usuário, permitindo que os usuários naveguem facilmente entre essas áreas de acordo com suas necessidades e permissões. Este modelo de navegação é projetado para facilitar o acesso rápido e eficiente às funções que o usuário está autorizado a utilizar.

O modelo de navegação do subsistema Usuário, descrito na Figura 9, é específico para detalhar o comportamento dos componentes e controladores envolvidos na navegação dentro deste subsistema. Este modelo descreve como os usuários interagem com as páginas relacionadas ao gerenciamento de perfil, cadastro de novos usuários, e outras funcionalidades específicas do gerenciamento de usuários. Ele garante que as interações sejam intuitivas e que os usuários possam realizar suas tarefas relacionadas à administração de usuários sem dificuldades.

Similar ao modelo de navegação do Subsistema Usuário, o modelo de navegação do Subsistema Pedido de Afastamento, descrito na Figura 10, articula o comportamento dos componentes e controladores que guiam o usuário através do processo de abertura, acompanhamento e gestão de pedidos de afastamento. Desde a solicitação inicial até a adição de documentos e recebimento de pareceres, este modelo assegura que os usuários possam navegar eficientemente por todas as etapas necessárias para gerenciar seus pedidos de afastamento.

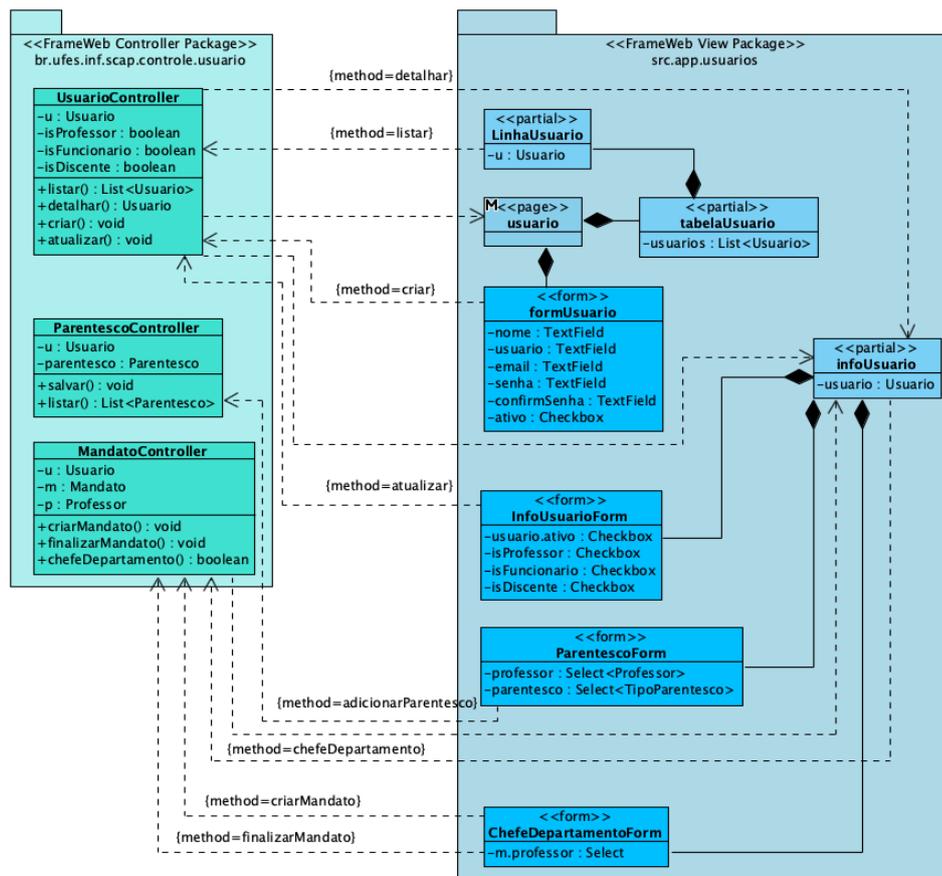


Figura 9 – Modelo de Navegação do Subsistema Usuário

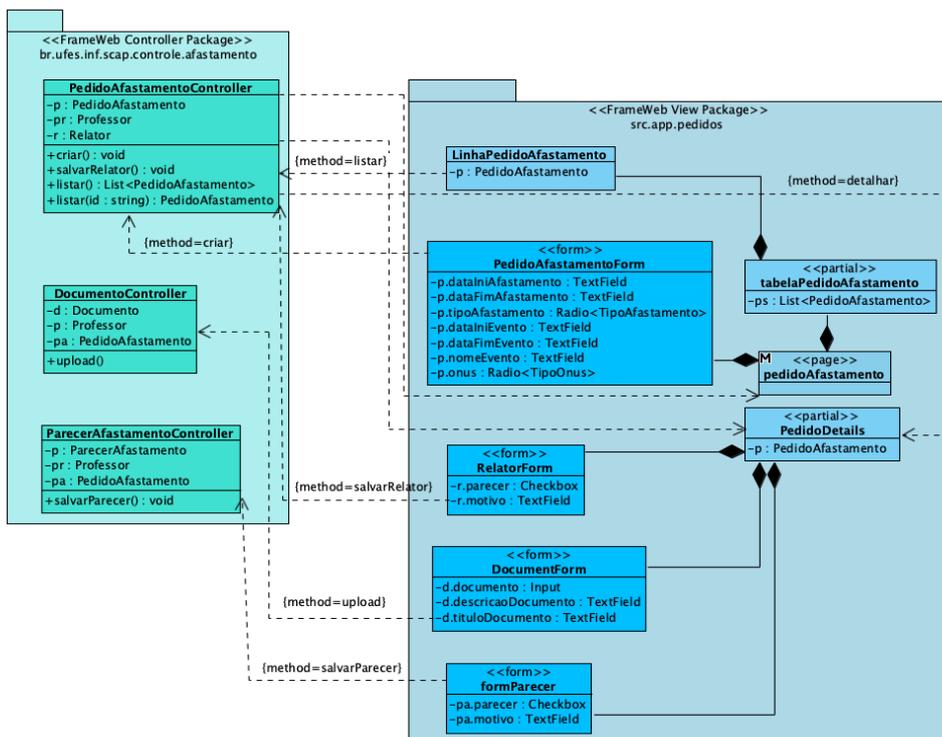


Figura 10 – Modelo de Navegação do Subsistema Pedido de Afastamento

Referências

- FOWLER, M. *Patterns of Enterprise Application Architecture*. [S.l.]: Addison-Wesley, 2002. ISBN 0321127420. Citado na página 6.
- PRADO, R. C. d. Aplicação do método frameweb no desenvolvimento de um sistema de informação utilizando o framework vraptor 4. Trabalho de Conclusão de Curso. 2015. Citado na página 2.
- 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*. Vitória, ES, Brazil: NEMO, 2020. p. 100–124. ISBN 9781393963035. Disponível em: <<http://purl.org/nemo/celebratingfalbo>>. Citado na página 6.
- 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 na página 7.