



**INSTITUTO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DA PARAÍBA
DIRETORIA DE DESENVOLVIMENTO DE ENSINO
COORDENAÇÃO DO CURSO DE ENGENHARIA DE COMPUTAÇÃO**

AUTORES:

ÍTALO SANTOS NEVES

JOSEHYLTON RICARDO FERREIRA NEVES

SISTEMA DE AGENDAMENTO DE HORÁRIO PARA COORDENAÇÃO - SAHC

**CAMPINA GRANDE - PB
2022**

AUTORES:

ÍTALO SANTOS NEVES

JOSEHYLTON RICARDO FERREIRA NEVES

SISTEMA DE AGENDAMENTO DE HORÁRIO PARA COORDENAÇÃO - SAHC

Trabalho de Conclusão de Curso apresentado ao Curso de bacharelado em ENGENHARIA DE COMPUTAÇÃO, do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – Campus Campina Grande, em cumprimento às exigências parciais para a obtenção do título de bacharel.

ORIENTADOR (A):

CESAR ROCHA VASCONCELOS

**CAMPINA GRANDE - PB
2022**

N518s Neves, Ítalo Santos.

Sistema de agendamento de horário para coordenação - SAHC / Ítalo Santos Neves, Josehylton Ricardo Ferreira Neves. - Campina Grande, 2022.

50 f. : il.

Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) - Instituto Federal da Paraíba, 2023.

Orientador: Prof. Cesar Rocha Vasconcelos.

1. Engenharia da Computação - Desenvolvimento de sistema 2. Spring framework 3. Java - Javascript I. Josehylton Ricardo Ferreira Neves II. Vasconcelos, Cesar Rocha. III. Título.

CDU 004.65

AUTORES:

ÍTALO SANTOS NEVES

JOSEHYLTON RICARDO FERREIRA NEVES

SISTEMA DE AGENDAMENTO DE HORÁRIO PARA COORDENAÇÃO - SAHC

Trabalho de Conclusão de Curso apresentado ao Curso de bacharelado em ENGENHARIA DE COMPUTAÇÃO, do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – Campus Campina Grande, em cumprimento às exigências parciais para a obtenção do título de bacharel.

Aprovada em ____ / ____ / _____

Banca Examinadora

Prof. Dr. Cesar Rocha Vasconcelos
Orientador (IFPB)

Prof.
Examinador

Prof.
Examinador

À Deus. À meus pais e familiares, por todo apoio e carinho!

Dedico!

AGRADECIMENTOS

Agradecimentos - Ítalo Santos Neves.

É com imensa gratidão que dedico este momento especial, começando por agradecer a Deus por conceder-me saúde, sabedoria e perseverança ao longo dessa jornada. Agradeço também ao meu pai, Antônio, e à minha mãe, Edileuza, que sempre estiveram ao meu lado, apoiando-me incondicionalmente em todas as etapas da minha vida. À minha esposa, Iasmyn, agradeço pelo seu apoio que me motivou a continuar perseverando, mesmo diante dos desafios. Também gostaria de expressar minha gratidão ao meu colega de TCC, Josehilton. Sua dedicação e comprometimento foram essenciais para o desenvolvimento deste trabalho. Ao meu professor e orientador, César, o meu profundo agradecimento por sua orientação, suporte e dedicação ao longo deste trabalho. Seu conhecimento, experiência e orientações foram fundamentais para o sucesso deste projeto. Por fim, agradeço a todos os professores que contribuíram para o meu crescimento profissional ao longo do curso e aos colegas que fizeram parte dessa jornada. Obrigado!

Agradecimentos - Josehylton Ricardo Ferreira Neves.

Em primeiro lugar, agradeço a Deus pela vida e por me dar forças todos os dias para superar os obstáculos que a vida impõe. Agradeço também a minha mãe Risalva por acreditar em mim, estar comigo nos momentos bons e ruins, me apoiar e por todo os esforço e dedicação para me criar e educar da melhor forma possível. Agradeço também a minha esposa Stephany que sempre esteve ao meu lado, me apoiando e dando força nos momentos mais difíceis da minha vida, por estar compartilhando comigo momentos bons e momentos ruins, por me trazer paz, felicidade e alegria. Agradeço também a meus irmãos Josehyres e Stefanya que também estão sempre apoiando minhas decisões, estão sempre comigo nos momentos mais difíceis e por todos os momentos tristes e felizes que passamos. Agradeço também a Italo, o meu companheiro de TCC, pois, a ajuda e parceria dele foi fundamental para conseguirmos concluir esta jornada. Agradeço também ao nosso orientador Cesar por seu conhecimento, dedicação e ajuda que foram valiosas para a conclusão desta fase da minha vida. Agradeço também a todos os meus amigos e professores que contribuíram para que eu conseguisse alcançar este objetivo.

“Nada resiste ao trabalho.”

Autor desconhecido

RESUMO

A comunicação efetiva entre estudantes e coordenação de cursos é importante para o bom funcionamento das instituições de ensino. Embora os meios tradicionais, como e-mails, sejam amplamente utilizados, há espaço para melhorias na interação e no fluxo de informações. Neste trabalho, foi desenvolvido um sistema de agendamento de horários exclusivo para a coordenação, comparando-o com soluções existentes, como o Google Calendar e o Shifts do Teams. O sistema proposto permite um controle de permissões diferenciado para alunos, coordenadores e administradores, além de oferecer facilidade no agendamento e uma abordagem dedicada a esse propósito. Utilizando tecnologias como o Spring Framework e seus submódulos (Spring Data, Spring Security, Spring Boot), Java, JavaScript, Vue.js, FullCalendar, Bootstrap, entre outras, a solução implementada visa preencher a lacuna na comunicação entre estudantes e coordenação. Com o sistema de agendamento, os estudantes terão maior facilidade em marcar reuniões com os responsáveis pela coordenação de seus cursos de forma simples e eficiente. Essa abordagem trará benefícios significativos para a comunicação dentro da universidade, permitindo que os estudantes obtenham suporte e esclareçam dúvidas com mais facilidade, resultando em uma experiência acadêmica mais satisfatória. Além disso, a coordenação poderá gerenciar os atendimentos de forma mais eficaz, otimizando recursos e garantindo um fluxo de comunicação eficiente. Assim, o sistema proposto complementa os meios de comunicação existentes e potencializa a interação entre coordenação e discentes, contribuindo para a melhoria da comunicação institucional e o aprimoramento do ambiente acadêmico.

Palavras-chave: Spring Framework; Sistema de Agendamento; Vue.js; Java; Javascript;

ABSTRACT

Effective communication between students and course coordination is important for the proper functioning of educational institutions. Although traditional means such as e-mails are widely used, there is room for improvement in interaction and information flow. In this work, a system for scheduling exclusive appointments for coordination was developed, comparing it with existing solutions, such as Google Calendar and Shifts from Teams. The proposed system allows a control of differentiated permissions for students, coordinators and administrators, in addition to offering ease in scheduling and a dedicated approach to this purpose. Using technologies such as Spring Framework and its submodules (Spring Data, Spring Security, Spring Boot), Java, JavaScript, Vue.js, FullCalendar, Bootstrap, among others, the integrated solution aims to fill the gap in communication between students and coordination. With the scheduling system, students will find it easier to arrange meetings with those responsible for coordinating their courses in a simple and efficient way. This approach has inspired benefits for communication within the university, allowing students to get support and clarify doubts more easily, resulting in a more satisfying academic experience. In addition, coordination will be able to manage services more effectively, optimizing resources and ensuring an efficient communication flow. Thus, the proposed system complements the existing means of communication and enhances the interaction between coordination and students, ensuring the improvement of institutional communication and the improvement of the academic environment.

Keywords: Spring Framework, Scheduling System, Vue.js, Java, Javascript.

LISTA DE ILUSTRAÇÕES

Figura 1 : Ciclo de vida do Scrum	20
Figura 2 : Diagrama de casos de uso (Autoria própria).....	25
Figura 3 : “Arquitetura do sistema SAHC” (Imagem: Autoria própria).....	27
Figura 4 : Tela de <i>login</i> (Imagem: Autoria própria).....	31
Figura 5 : Tela de cadastro (Imagem: Autoria própria).....	32
Figura 6 : Tela de recuperar senha (Imagem: Autoria própria).....	33
Figura 7 : Tela de boas-vindas (Imagem: Autoria própria).....	33
Figura 8 : Tela “Sobre” (Imagem: Autoria própria).....	34
Figura 9 : Tela de “Erro” (Imagem: Autoria própria).....	34
Figura 10 : Tela “Meus Agendamentos” (Imagem: Autoria própria).....	36
Figura 11 : Tela “Disponibilizar horários” área do coordenador (Imagem: Autoria própria).....	37
Figura 12 : Tela “Agendar horário” área do aluno (Imagem: Autoria própria).....	37
Figura 13 : Tela “Agendar horário” com modal de confirmação presente após usuário selecionar data (Imagem: Autoria própria).....	38
Figura 14 : Tela “Agendar horário” após aluno confirmar agendamento (Imagem: Autoria própria).....	38
Figura 15 : Tela “Redefinir Senha” área do aluno e coordenador(Imagem: Autoria própria).....	39
Figura 16 : Tela “Gerenciar Usuários” (Imagem: Autoria própria).....	40
Figura 17 : Tela “Gerenciar Coordenações” (Imagem: Autoria própria).....	41
Figura 18 : Tela “Cadastrar Coordenação” (Imagem: Autoria própria).....	41
Figura 19 : Tela “Editar Coordenação” (Imagem: Autoria própria).....	42
Figura 21 : Estrutura de Pacotes do back-end (Imagem: Autoria própria).....	43

LISTA DE QUADROS

Quadro 1 : Princípios do Manifesto ágil.....	19
Quadro 2 : Diferenciação de recursos entre sistemas	22
Quadro 3 : Requisitos Funcionais	23
Quadro 4 : Requisitos não funcionais	26
Quadro 5 : Cronograma das atividades	29

LISTA DE ABREVIATURA E SIGLAS

IFPB	Instituto Federal da Paraíba
HTTPS	Hypertext Transfer Protocol Secure
JSON	JavaScript Object Notation
J2EE	Java 2 Platform Enterprise Edition
XML	Extensible Markup Language
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
SAHC	Sistema de Agendamento de Horário para Coordenação
API	Application Programming Interface
SUAP	Sistema Unificado de Administração Pública
JWT	JSON Web Token
SPA	Single Page Application
NPM	Node Package Manager

SUMÁRIO

LISTA DE ILUSTRAÇÕES	9
LISTA DE QUADROS	10
LISTA DE ABREVIATURA E SIGLAS	11
1 INTRODUÇÃO	14
1.1 Objetivos	14
1.1.1 Objetivo Geral	14
1.1.2 Objetivos Específicos	14
1.2 Relevância	15
1.3 Contribuições	15
2 REFERENCIAL TEÓRICO	16
2.1 Tecnologias para Construção de Sistemas <i>Web</i>	16
2.1.1 Tecnologia Servidor	16
2.1.1.1 <i>Spring Boot</i>	17
2.1.1.2 <i>Spring Data</i>	17
2.1.1.3 <i>Spring Security</i>	17
2.1.2 Tecnologia Cliente	18
2.2 Metodologias Ágeis	19
2.2.1 Scrum	20
2.3 Padrão Arquitetural	21
2.4 Sistemas Relacionados	21
3 METODOLOGIA	23
3.1 Análise e Definição dos Requisitos	23
3.1.1 Requisitos Funcionais	23
3.1.2 Requisitos Não-funcionais	25
3.2 Arquitetura Cliente-Servidor	26
3.3 Escolha das Tecnologias	27
3.3.1 Tecnologias <i>Back-end</i>	27
3.3.1.1 Token de Acesso	28
3.3.2 Tecnologias <i>Front-end</i>	28
3.4 Ferramentas Auxiliares	29
3.5 Processo de Desenvolvimento	29
3.6 Cronograma das Atividades	29

4 RESULTADOS E DISCUSSÕES	30
4.1 Listagem de Telas por Perfil de Usuário	30
4.1.1 Telas Comuns Para Todos os Usuários	30
4.1.1.1 Página de <i>Login</i>	30
4.1.1.2 Página de Cadastro de Usuário	31
4.1.1.3 Tela de “Recuperar Senha”	32
4.1.1.4 Página de Boas Vindas	33
4.1.1.5 Tela “Sobre”	33
4.1.1.6 Tela de “Erro”	34
4.1.2 Telas dos Alunos e Coordenadores	35
4.1.2.1 Tela “Meus Agendamentos”	35
4.1.2.2 Telas “Disponibilizar Horários” e “Agendar Horários”	36
4.1.2.3 Tela de “Redefinir Senha”	38
4.1.3 Telas do Administrador	39
4.1.3.1 Tela “Gerenciar Usuários”	39
4.1.3.2 Tela “Gerenciar Coordenações”	40
4.1.3.3 Tela “Cadastrar Coordenação”	41
4.1.3.4 Tela “Editar Coordenador”	42
4.2 Projeto Arquitetural do Sistema	42
5 CONCLUSÃO	45
5.1 Trabalhos Futuros	46
Apêndice A : Guia de instalação do SAHC	47
A.1 Instalação do Back-end	47
A.2 Instalação do Front-end	48
A.3 Acessando o Swagger	48
REFERÊNCIAS	49

1 INTRODUÇÃO

A comunicação com a coordenação do curso do qual o discente está matriculado ainda é um dos problemas enfrentados pelos discentes do IFPB. Em alguns casos, muitos alunos desconhecem o horário da coordenação e, por vezes, necessitam realizar várias tentativas sem sucesso para encontrar o coordenador em sua sala de atendimento.

Um dos principais métodos de comunicação institucional utilizados atualmente é o *e-mail*. Porém, como uma forma alternativa de comunicação foi pensado em um sistema que possa agendar horários com a coordenação para otimizar o processo de comunicação. É nesse sentido que o sistema proposto neste trabalho pode complementar essa comunicação e, conseqüentemente, potencializar a interação entre coordenações e discentes.

A partir de uma breve pesquisa percebeu-se que os sistemas de agendamento existentes lidam com agendamentos de forma geral e o organizador da agenda não consegue compartilhar sua agenda para que as pessoas reservem horários na mesma. Diante disso, neste trabalho, buscou-se implementar uma solução alternativa para atender essa demanda, que poderá facilitar e melhorar a comunicação entre as partes.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo deste projeto é desenvolver um sistema de agendamento funcional e seguro para uso dos discentes e das coordenações dos cursos para agiliza o processo de agendamento de horários de atendimento entre aluno-coordenador.

1.1.2 Objetivos Específicos

- Elaborar o projeto arquitetural do sistema, descrevendo seus principais componentes e funcionalidades;
- Desenvolver o projeto de um *sistema* na *Web* para os alunos e coordenações de cursos poderem realizar o cadastro, modificações e visualização de horários agendados de forma interativa, através de uma interface amigável;
- Projetar um banco de dados que possa suportar a alocação e alteração de agendamentos de forma remota, concorrente, segura;
- Realizar uma avaliação confiável dos componentes principais da aplicação, utilizando metodologias de desenvolvimento orientada a testes de unidade e de integração adequados;

1.2 Relevância

Após a conclusão do projeto, os coordenadores terão a possibilidade de melhor administrar seus horários de atendimento em relação às suas atividades na coordenação, podendo concluí-las com mais facilidade e organização. Com isso, os alunos serão diretamente beneficiados, pois terão uma forma de acesso alternativa a coordenação do curso.

O sistema de agendamento irá promover às coordenações dos cursos praticidade para cadastrar seus horários de atendimento e uma interface organizada e intuitiva para facilitar a visualização e organização dos mesmos.

Os alunos terão fácil acesso aos horários disponíveis da coordenação, a partir de uma interface amigável, podendo assim realizar agendamentos com o coordenador de forma rápida e simples.

Sendo assim, o uso do sistema proposto viabilizará uma maior e efetiva interação entre discente e coordenador, trazendo, dessa forma, benefícios para ambas as partes.

1.3 Contribuições

O sistema proposto contribuirá para o agendamento, visualização e organização de horários de atendimento da coordenação do curso. Este sistema será uma abordagem complementar e alternativa a o método tradicional usado atualmente, agregando valor de forma significativa na comunicação entre discente-coordenação. Os coordenadores disponibilizarão seus horários de atendimento de forma simplificada e organizada, enquanto os discentes irão conseguir visualizar e agendar um horário com a coordenação. Ao final deste trabalho será entregue o projeto completo, funcional e documentado com os repositórios utilizados no *back-end* e no *front-end* e toda sua codificação pronta para ser implantada.

2 REFERENCIAL TEÓRICO

Neste tópico são introduzidos os principais conceitos necessários ao entendimento desse projeto. Inicialmente, vamos apresentar de forma breve quais tecnologias foram necessárias para a construção do nosso sistema Web (Seção 2.1).

2.1 Tecnologias para Construção de Sistemas *Web*

O funcionamento típico de um aplicativo web depende exclusivamente do servidor web e de um navegador. O usuário, através de um navegador, acessa o front-end da aplicação e faz uma solicitação, geralmente através do protocolo *HTTPS* (*Hyper Text Transfer Protocol Secure*), ao servidor. Este por sua vez devolve uma resposta para o usuário, que pode ser um arquivo *html* já com as informações renderizadas na tela ou até mesmo um arquivo mais simples, como o *JSON* (*JavaScript Object Notation*), com as informações necessárias. Esse tipo de resposta vai depender da arquitetura de software utilizada na implementação do projeto. Adiante, vamos falar sobre a arquitetura do nosso projeto (Seção 2.3).

2.1.1 Tecnologia Servidor

As tecnologias servidor são as responsáveis por receber, gerenciar e tratar as requisições solicitadas pelo cliente. Grande parte dos sistemas web utilizam alguma linguagem de programação ou um *framework web* para desenvolver seus sites e ter seus conteúdos exibidos de forma dinâmica (MOZILLA, 2022).

Java é uma linguagem reconhecida por sua confiabilidade, segurança e capacidade de funcionar em diversas plataformas. Sendo orientada a objetos, oferece versatilidade para o desenvolvimento de aplicações em diferentes contextos, como *desktop*, *web* e *android*. Por isso, tem sido amplamente adotada pelos desenvolvedores ao longo de quase três décadas. Uma de suas principais vantagens é sua adequação para construção de aplicações de grande porte e robustas (SCHILDT, 2018).

Todo desenvolvedor que utiliza *J2EE* (*Java Platform Enterprise Edition*) sabe o quão trabalhoso é realizar a configuração de uma aplicação, sendo necessário a criação de pastas e arquivos extensos no formato *XML* (*Extensible Markup Language*) e um pequeno erro impede o funcionamento correto da aplicação, o que resulta em muito trabalho para a identificação do problema. Então, devido a essa complexidade, o especialista em computação Rod Johnson criou o *Spring Framework*. O *Spring Framework* foca na criação de aplicativos corporativos, tirando os vínculos desnecessários com configurações de ambiente e instalação de

dependências, deixando mais tempo para que as equipes de desenvolvimento foquem nas regras de negócios do aplicativo. (VMWARE, 2022). No que segue, vamos discorrer sobre os principais módulos disponíveis no ecossistema Spring.

2.1.1.1 *Spring Boot*

O Spring Boot é um módulo especial erigido como uma extensão do framework Spring. (MICROSOFT, 2022).

O Spring Boot trouxe agilidade, possibilitando ao desenvolvedor focar no desenvolvimento da aplicação propriamente dito. Com pré-configurações incorporadas no Spring Boot (ele não gera código, apenas analisa e configura o projeto automaticamente) basicamente o programador só necessita “rodar a aplicação” para que todo o seu sistema esteja em pleno funcionamento, mas isto não quer dizer que não seja possível customizar suas configurações.

2.1.1.2 *Spring Data*

A missão do *Spring Data* é fornecer um modelo de programação familiar e consistente baseado em *Spring* para acesso a dados, mantendo as características especiais do armazenamento de dados subjacentes. Ele facilita o uso de tecnologias e a integração com fontes de dados, bancos de dados relacionais e não relacionais, estruturas de redução de mapas e serviços de dados baseados em nuvem. Este é um projeto que contém muitos sub projetos específicos para um determinado banco de dados (VMWARE, 2022). O sub projeto do *Spring Data* que será trabalhado é o *Spring Data JPA*.

2.1.1.3 *Spring Security*

Segundo o Governo Federal (2022), em 2021 o número de domicílios com acesso à internet no Brasil chegou a 90,0%, segundo dados da Pesquisa Nacional por Amostra de Domicílios. Em termos absolutos, são 65,6 milhões de domicílios conectados, 5,8 milhões a mais do que em 2019.

A partir deste contexto podemos perceber que, cada dia que passa a população está mais conectada e com isso todos nós estamos mais suscetível a invasões e ataques cibernéticos. Isso é bastante preocupante, principalmente no que se refere a acesso a dados pessoais. Um dado interessante que Ingrid Oliveira (2022) nos mostra é que o Brasil registrou no primeiro semestre de 2022, 31,5 bilhões de tentativas de ataques cibernéticos a empresas. O número é 94% superior na comparação com o primeiro semestre do ano passado, quando

foram 16,2 bilhões de registros. Por este motivo, vemos cada vez mais empresas investindo pesado em segurança e com programas de recompensas como o Bug Bounty.

O Spring Security é uma estrutura de autenticação e controle de acesso poderosa e altamente personalizável. É o padrão de fato para proteger aplicativos baseados em Spring. Ele tem uma configuração que se concentra em fornecer autenticação e autorização para aplicativos Java. Como todos os projetos Spring, o poder real do Spring Security é encontrado na facilidade com que ele pode ser estendido para atender aos requisitos personalizados (VMWARE, 2022).

2.1.2 Tecnologia Cliente

As tecnologias cliente consistem no desenvolvimento da interface do usuário, onde a maior parte do código é executada em resposta à sua solicitação. (NEVES; CARVALHO; DA HORA, 2020).

Segundo Haverbeke (2018), *JavaScript* é uma linguagem de programação *web* que a maioria dos sites navegadores modernos usam, isso é possível através de um tradutor de *JavaScript* presente nos navegadores, tornando-o a linguagem de programação mais usada na história (HAVERBEKE, 2018). O *JavaScript* faz parte das três principais tecnologias que todo desenvolvedor *web* deve conhecer que são: *CSS*, para definir a aparência dessas páginas; *HTML*, para especificar o conteúdo das páginas *web* e *JavaScript* para especificar o comportamento.

A forma de desenvolvimento de sistemas *front-end* passou por transformações significativas, com o surgimento de diversos *frameworks* modernos que oferecem benefícios como reutilização de código e padronização. No contexto específico deste trabalho, optou-se pelo *Vue.js* em vez de outras opções, como o *React* e *Angular*, devido a uma série de razões. O *Vue.js* é conhecido por sua flexibilidade, desempenho e sintaxe clara e limpa, o que facilita a compreensão e a manutenção do código. Além disso, o *Vue.js* possui uma curva de aprendizado suave, o que permite aos desenvolvedores uma rápida adaptação e produtividade. Embora outros *frameworks*, como o *React* e *Angular*, também utilizem *JavaScript*, a escolha pelo *Vue.js* foi motivada por sua combinação de recursos, simplicidade e eficácia para as necessidades específicas deste projeto. Esses atributos foram considerados relevantes para alcançar os objetivos de desenvolvimento do sistema *front-end* proposto (INCAU, 2017).

2.2 Metodologias Ágeis

As metodologias ágeis rapidamente se tornaram uma abordagem de desenvolvimento significativa para a engenharia de software. Essas são uma série de técnicas que ajudam as equipes a desenvolver produtos de software com mais eficiência. Isso porque promove flexibilidade, colaboração e capacidade de resolução de problemas entre a equipe. Essencialmente, as metodologias ágeis são benéficas em ambientes pessoais e profissionais.

Embora esses métodos ágeis sejam todos baseados na noção de desenvolvimento e entrega incremental, eles propõem diferentes processos para alcançar tal objetivo. No entanto, compartilham um conjunto de princípios, com base no manifesto ágil, e por isso têm muito em comum (SOMMERVILLE, 2019).

O Quadro 1 a seguir elenca os princípios do manifesto ágil.

Quadro 1: Princípios do Manifesto ágil.

Princípios	Descrição
Geração de valor	A maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.
Flexibilidade	Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
Frequência	Entregar o software em funcionamento com frequência, seja na escala de semanas ou meses, dando preferência a períodos mais curtos.
União	Tanto pessoas relacionadas a negócios como desenvolvedores devem trabalhar em conjunto, diariamente, durante todo o curso do projeto.
Motivação	Construir projetos em torno de indivíduos motivados, dando a eles o ambiente e o suporte necessário e confiando neles para fazer o trabalho.
Comunicação	O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é por meio de conversa face a face
Funcionalidade	Software funcionando é a medida primária de progresso.
Sustentabilidade	Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
Revisão	Atenção contínua à excelência técnica e bom design, aumenta a agilidade.
Simplicidade	A arte de maximizar a quantidade de trabalho que não precisou ser feito.
Organização	As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis.

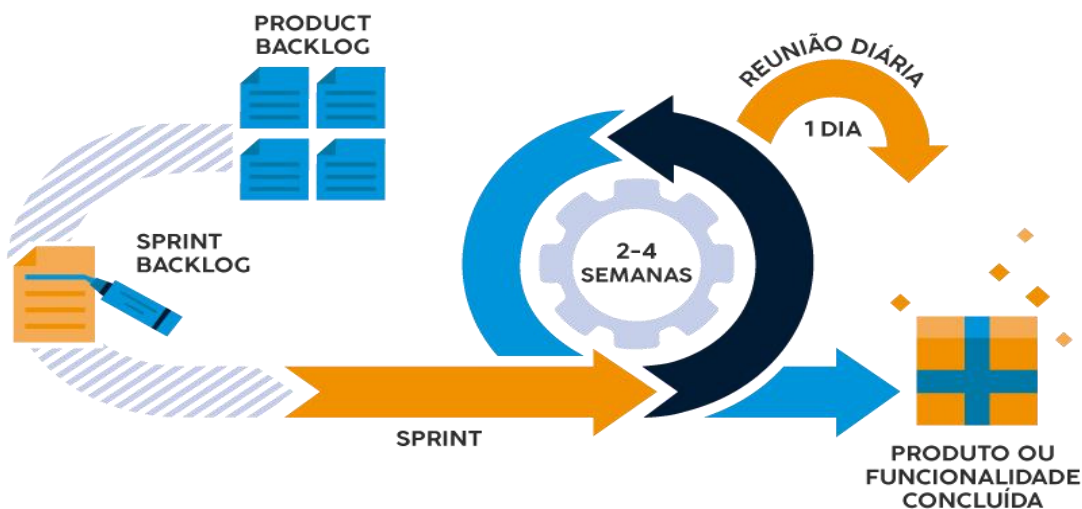
Autoavaliação	Em intervalos regulares, a equipe reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo
---------------	--

Fonte: CALAIS, 2021 (Adaptado).

2.2.1 Scrum

O *Scrum* é uma estrutura ágil de gestão de projetos que ajuda as equipes a estruturar e gerenciar o trabalho por meio do conjunto de valores, princípios e práticas. No *Scrum*, existem três fases. A primeira é uma fase de planejamento geral, em que se estabelecem os objetivos gerais do projeto e da arquitetura do software. Em seguida, ocorre uma série de ciclos de sprint, sendo que cada ciclo desenvolve um incremento do sistema. Finalmente, a última fase do projeto encerra o projeto, completa a documentação exigida, como quadros de ajuda do sistema e manuais do usuário, e avalia as lições aprendidas com o projeto (SOMMERVILLE, 2019).

Figura 1: Ciclo de vida do Scrum



Fonte: TECNICON, 2019

A característica inovadora do Scrum é sua fase central, chamada ciclos de sprint. Um sprint do Scrum é uma unidade de planejamento na qual o trabalho a ser feito é avaliado, os recursos para o desenvolvimento são selecionados e o software é implementado. No fim de um sprint, a funcionalidade completa é entregue aos *stakeholders* (SOMMERVILLE, 2019).

Um grande benefício das metodologias ágeis é que podem ser trabalhadas e adaptadas para a realidade de cada projeto.

2.3 Padrão Arquitetural

A arquitetura de *software* tornou-se um campo de estudo reconhecido internacionalmente na última década. À medida que os programas se tornaram mais complexos, seu *design* tornou-se cada vez mais importante. Os *softwares* agora têm um impacto maior do que nunca, e é crucial que esses sistemas tenham uma estrutura bem projetada. O conceito de arquitetura de *software* existe há décadas, mas o campo ainda está em desenvolvimento. Este é um momento interessante para se envolver com arquitetura de *software*; com novas técnicas arquitetônicas, padrões e padrões de projeto surgindo diariamente.

Segundo SOMMERVILLE (2019), pode-se pensar em um padrão de arquitetura como sendo uma descrição abstrata, estilizada, de boas práticas experimentadas e testadas em diferentes sistemas e ambientes. Um padrão de arquitetura descreve uma organização de sistema bem-sucedida em sistemas anteriores. Deve incluir informações de quando o uso desse padrão é adequado, e seus pontos fortes e fracos.

2.4 Sistemas Relacionados

Foi levada a efeito uma pesquisa dos principais sistemas relacionados e observou-se suas características, vantagens e, principalmente, o que podia ser melhorado, para otimizar essas funcionalidades existentes.

O aplicativo *Shifts* (MICROSOFT, 2023) é um sistema de agendamento disponível no *Microsoft Teams*. Ele foi desenvolvido com o objetivo de gerenciar a comunicação entre equipes. Neste aplicativo, é possível que qualquer usuário do sistema marque uma reunião com outro usuário de sua organização ou com uma equipe. É um aplicativo que atende bem para o intuito em que foi feito, mas que não atende o objetivo deste trabalho, pois não possui a opção de disponibilizar agenda para os usuários reservarem um horário.

Outro sistema similar ao SAHC (Sistema de agendamento de horário para coordenação) proposto, é o Google Agenda, que tem como recursos principais: permitir aos usuários agendar reuniões, eventos, receber lembretes sobre as atividades agendadas em seu calendário e compartilhar agenda com outras pessoas da organização (um diferencial importante, quando comparado ao aplicativo *Shifts* da *Microsoft Teams*). No entanto, esse recurso só está disponível para os usuários que têm o pacote *G-suite*¹, e as pessoas que tiverem acesso a essa agenda poderão excluir, compartilhar com outras pessoas e editar

¹ O g-suite é um pacote de produtos google com recursos exclusivos para os assinantes.

eventos da agenda. Feita essa análise, serão expostas as vantagens do SAHC em relação aos seus concorrentes.

O SAHC vai ser um sistema de posse completa do IFPB. Assim a instituição não seria tão prejudicada caso a assinatura com o *G-suite* fosse cancelada ou até mesmo se o Google impusesse limitações restritivas ao acesso de recursos importantes da ferramenta. Logo, o sistema proposto neste trabalho estará sempre disponível e livre de restrições de negócios de empresas. Ademais, tem outras vantagens, como: Integridade dos horários disponíveis para agendamentos, pois apenas os coordenadores vão poder editar ou excluir seus horários disponíveis cadastrados, o acesso a agenda só vai ser permitido pelos usuários cadastrados no sistema dando assim uma maior confiabilidade às informações disponíveis na plataforma. Com a entrada ou saída de alunos da instituição o coordenador teria que adicionar manualmente cada novo *e-mail* no grupo de agendamento criado por ele (no caso do google calendar, o shift nem oferece essa opção) o que não seria legal pois seria mais uma atividade atribuída a coordenação, nosso objetivo é otimizar o tempo do coordenador que já tem diversas tarefas atribuídas no seu ofício, com o SAHC o próprio aluno vai fazer seu cadastro e vai ter o acesso a agenda do curso ao qual está cadastrado.

Quadro 2: Diferenciação de recursos entre sistemas

Recursos	Shift	Google Agenda	SAHC
Agendar horários	✓	✓	✓
Consultar horário agendado	✓	✓	✓
Disponibilização de horários		✓	✓
Posse completa do sistema			✓
Possibilidade de customização			✓

3 METODOLOGIA

Nesta seção, serão apresentadas as etapas e abordagens metodológicas adotadas durante o desenvolvimento do sistema SAHC. Serão abordados tópicos como a arquitetura cliente-servidor, a escolha das tecnologias, as ferramentas utilizadas e o processo de desenvolvimento.

3.1 Análise e Definição dos Requisitos

O levantamento de requisitos é uma etapa fundamental no processo de desenvolvimento de *software*. Refere-se ao processo de identificação, coleta e documentação de necessidades, expectativas e restrições do sistema que está sendo desenvolvido. O objetivo principal é compreender e definir claramente o que o sistema deve fazer, quais problemas ele deve resolver e quais são as exigências dos usuários.

Existem várias técnicas utilizadas no levantamento de requisitos. Algumas comumente utilizadas são: entrevistas com os *stakeholders*, questionários, observação direta dos usuários em seu ambiente de trabalho, prototipagem, workshops e análise de documentos existentes. A escolha das técnicas depende do projeto, do contexto e dos recursos disponíveis.

3.1.1 Requisitos Funcionais

Os requisitos funcionais têm como finalidade descrever as funcionalidades que um sistema ou *software* deve realizar. Em particular, definem as ações que o sistemas deve executar em resposta a determinados estímulos ou eventos. Abaixo, são mostrados os requisitos funcionais do sistema SAHC proposto:

Quadro 3: Requisitos Funcionais

Requisito Funcional	Requisitos Funcionais
RF01	<i>O sistema deve permitir a realização de cadastro de usuários (Discentes e Coordenador).</i>
RF02	<i>O sistema deve possibilitar que o usuário cadastrado modifique sua senha.</i>
RF03	<i>O sistema deve possibilitar que o usuário cadastrado recupere sua senha em caso de esquecimento da mesma que será enviada via para o e-mail cadastrado.</i>
RF04	<i>O sistema possibilita o discente visualizar todos os horários disponíveis para agendamento com a sua coordenação.</i>
RF05	<i>O sistema possibilita o discente visualizar todos os horários agendados da sua</i>

	<i>coordenação.</i>
RF06	<i>O sistema deve permitir que o discente possa agendar um horário com a coordenação do seu curso e após o agendamento receberá um e-mail de agendamento confirmado.</i>
RF07	<i>O sistema deve permitir que o discente possa desagendar o horário previamente agendado.</i>
RF08	<i>O sistema possibilita o discente de visualizar todos os horários agendados pelo mesmo.</i>
RF09	<i>O sistema possibilita que o coordenador possa cadastrar horários para serem agendados por discentes.</i>
RF10	<i>O sistema possibilita que o coordenador possa descadastrar horários previamente cadastrados pelo mesmo.</i>
RF11	<i>O sistema possibilita que o administrador cadastre coordenações de cursos.</i>
RF12	<i>O sistema possibilita que o administrador atualize informações das coordenações existentes.</i>
RF13	<i>O sistema possibilita que o administrador exclua coordenações cadastradas.</i>
RF14	<i>O sistema possibilita que o administrador exclua usuários.</i>
RF15	<i>O sistema possibilita que o administrador altere o privilegio de usuários (aluno ou coordenador).</i>

Fonte: Autoria própria.

Diagrama de casos de uso

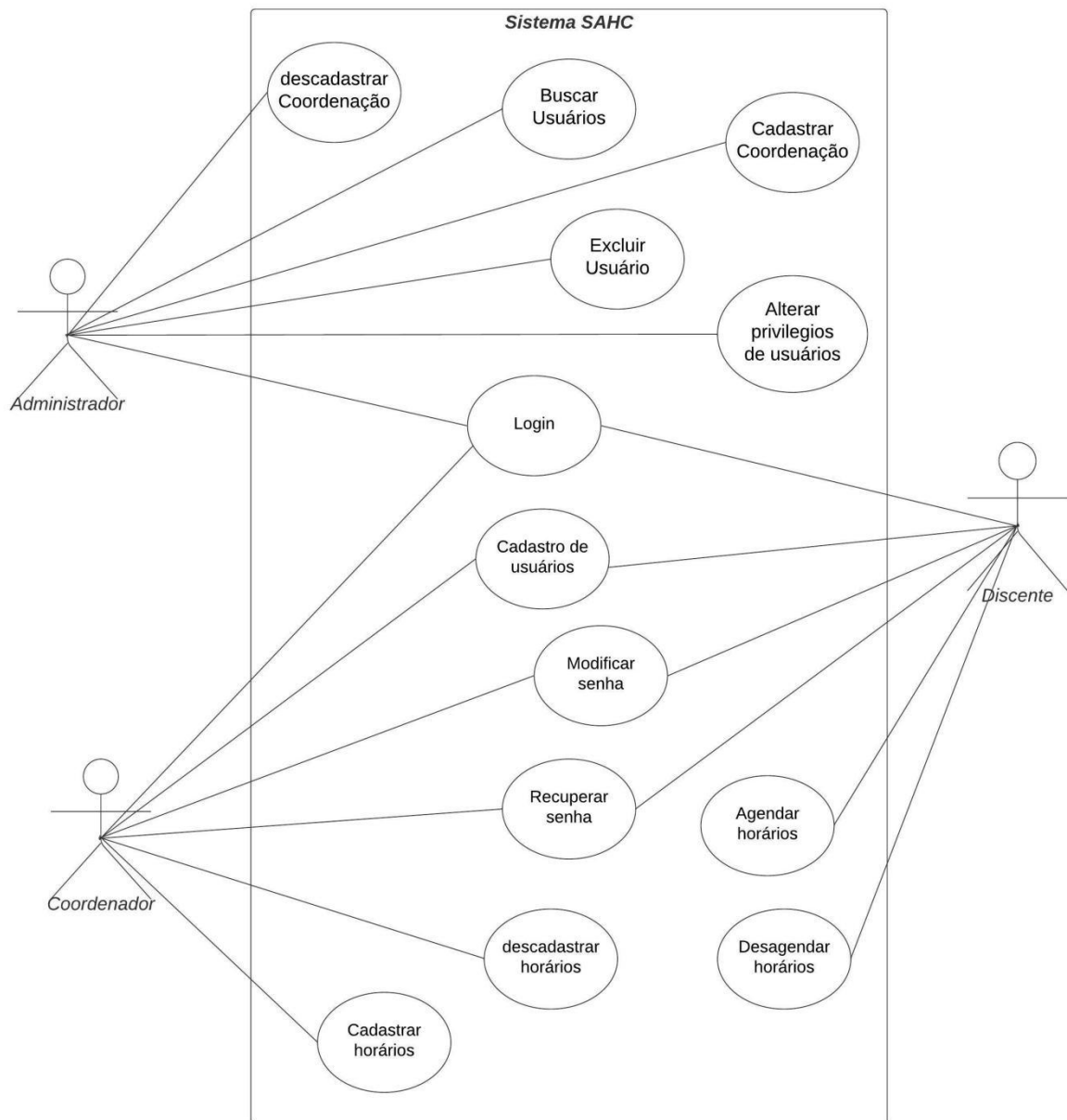


Figura 2: Diagrama de casos de uso (Autoria própria).

3.1.2 Requisitos Não-funcionais

Os requisitos não funcionais são responsáveis por descrever características e restrições que não estão diretamente relacionadas às funcionalidades específicas do sistema, mas que têm impacto na sua qualidade, desempenho, segurança e outros aspectos. Temos alguns requisitos não funcionais abaixo:

Quadro 4: Requisitos não funcionais

Requisito Não Funcional	Requisitos Não Funcionais
RNF01	<i>Extensibilidade: O sistema deve possuir a facilidade de expansão e modificação do sistema para atender novos requisitos futuros e permitir a incorporação de novas funcionalidades.</i>
RNF02	<i>Testabilidade: O sistema deve possuir a capacidade de ser testado facilmente, por meio de testes de integração e de unidade.</i>
RNF03	<i>Segurança: O acesso deve ser realizado através de usuário e senha previamente cadastrados no sistema, além da utilização de um token de acesso JWT garantindo que somente usuários cadastrados e que efetuaram login poderá ter acesso as funcionalidades. O sistema também garante que existirá diferentes privilégios de usuários permitindo o acesso a diferentes recursos a depender do usuário. O sistema também garante que existirá diferentes privilégios de usuários permitindo o acesso a diferentes recursos a depender do usuário.</i>

Fonte: Autoria própria.

3.2 Arquitetura Cliente-Servidor

Arquitetura de *software* é a estrutura fundamental de um sistema. Ela envolve a organização dos componentes do sistema, suas interações, os princípios de *design* e as decisões tomadas para atingir os objetivos desejados. A arquitetura de *software* fornece uma visão abstrata e de alto nível do sistema, delineando sua estrutura, comportamento e características principais.

A escolha da arquitetura para embasar o sistema proposto foi a arquitetura Cliente-Servidor. Essa arquitetura é uma arquitetura amplamente utilizada em sistemas de computação, redes e desenvolvimento de software. Nela, a interação entre os computadores ocorre por meio de uma divisão clara de papéis: um computador, chamado de servidor, fornece serviços e/ou recursos, enquanto outros computadores, chamados de clientes, solicitam e recebem esses serviços.

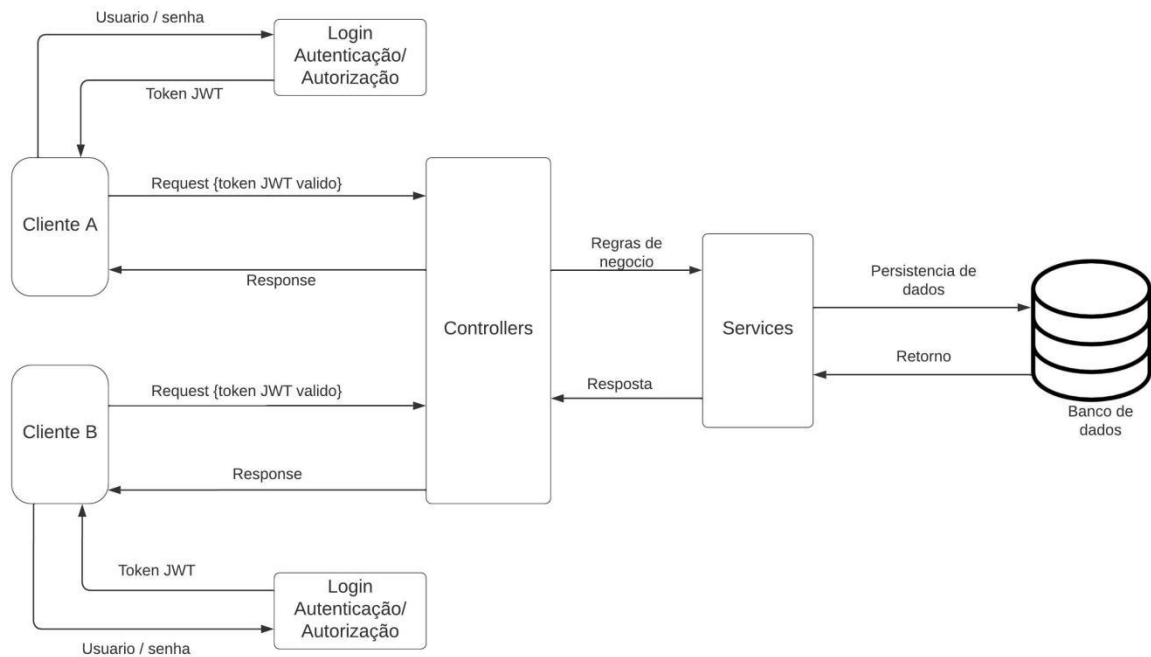


Figura 3: “Arquitetura do sistema SAHC” (Imagem: Autoria própria).

3.3 Escolha das Tecnologias

A escolha das tecnologias para o desenvolvimento de um *software* é uma etapa crucial no processo de criação de um sistema. As tecnologias escolhidas podem afetar a eficiência, a escalabilidade, a segurança e a facilidade de manutenção do software. Existem alguns pontos a serem considerados ao fazer esta escolha, alguns deles são:

- Experiência da equipe;
- Compatibilidade e integrações;
- Suporte a comunidade;
- Tendências e Longevidade;

3.3.1 Tecnologias *Back-end*

Para o desenvolvimento deste projeto, será escolhida a linguagem de programação *Java* e o *Spring Framework* pois, conforme citado na seção 2.1.1 o mesmo contém submódulos como por exemplo *Spring Data*, *Spring Boot*, *Spring Security*, *Spring Mail*, entre outros, que simplificam sobremaneira a configuração e o desenvolvimento da aplicação fornecendo agilidade, confiabilidade e segurança. Em conjunto com o *Java* utilizou-se o *Maven* (MAVEN, 2023), ferramenta de gerenciamento e compreensão de projetos de software desenvolvida pela *Apache*, que serve para o gerenciamento de dependências e automatização de projetos facilitando a utilização de bibliotecas como *Lombok*, para agilizar o trabalho e

reduzir várias linhas de códigos com suas anotações, *JUnit* para realização de testes da aplicação entre outras bibliotecas que auxiliarão no processo de desenvolvimento. No contexto do *back-end* utilizaremos o Sistema Gerenciador de Banco de Dados *MySQL* (ORACLE, 2023), O conjunto mais abrangente de recursos avançados, ferramentas de gerenciamento e suporte técnico para alcançar os mais altos níveis de escalabilidade, segurança, confiabilidade, para integração dos dados.

3.3.1.1 Token de Acesso

A segurança de um sistema é de extrema importância então com a utilização do *Spring Security*, decidiu-se por utilizar o *JSON Web Token (JWT)* como forma de autenticação/autorização para os usuários utilizarem o sistema com confiança e segurança.

Em particular, um *token JWT* consiste em três partes separadas por pontos: o cabeçalho (*header*), informações do token (*payload*) e a assinatura (*signature*). Cada parte é codificada em Base64 e juntas formam um token no formato "header.payload.signature". O cabeçalho contém informações sobre o algoritmo de criptografia utilizado para assinar o token. A carga útil (*payload*) contém as declarações (*claims*) que representam os dados associados ao *token*, como informações do usuário, permissões ou outros dados personalizados. A assinatura é usada para verificar a integridade do *token*.

A principal vantagem do *JWT* é que ele é auto-contido, o que significa que o servidor não precisa armazenar informações sobre o *token* do lado do servidor. Cada vez que o servidor recebe um *token JWT*, ele pode verificar a autenticidade do *token*, a validade e as declarações contidas nele apenas verificando a assinatura usando uma chave secreta, com isso podemos criar um servidor *Stateless* onde não existe a necessidade de guardar as sessões dos usuários.

3.3.2 Tecnologias Front-end

Para a criação do *front-end* do sistema foi escolhida a linguagem de programação *JavaScript* e a partir desta escolha utilizaremos o *framework Vue.js*, pois, fornece uma estrutura e um conjunto de ferramentas para simplificar o desenvolvimento de aplicativos *web* interativos em *Single-Page-Applications (SPA)*. O *Vue.js* permite a criação de componentes reutilizáveis, facilitando a organização e a manutenção do código. Também será utilizada a linguagem de marcação *HTML5* e a linguagem de estilização *CSS3* e para o gerenciamento das bibliotecas auxiliares utilizaremos o gerenciador de pacotes node (*NPM*) e através dele poderemos incluir bibliotecas como *Bootstrap5*, *Fullcalendar*, *jwt-decode*, *pinia*, entre outras.

3.4 Ferramentas Auxiliares

As ferramentas auxiliares no desenvolvimento de software desempenham um papel fundamental ao facilitar o processo de criação, edição e depuração de código. Elas ajudam a melhorar a produtividade, a eficiência e a qualidade do código, além de proporcionarem uma experiência de desenvolvimento mais agradável e organizada.

Priorizou-se a escolha de ferramentas gratuitas e de código aberto neste trabalho:

- *Eclipse*;
- *IntelliJ*;
- *Visual Studio Code*;
- *MySQL Workbench*;
- *Postman*;
- *Insomnia*;
- *Notepad++*;
- *TortoiseGIT*;
- *Github*;

3.5 Processo de Desenvolvimento

No desenvolvimento desse projeto vamos utilizar a metodologia ágil *Scrum* para obter ganhos de produtividade e performance. No entanto, vamos utilizá-la de forma adaptada para se adequar ao nosso time, no qual é composto por apenas dois desenvolvedores. A *Sprint Scrum* se dará de forma semanal e realizaremos *Dailys* de 10 minutos para alinhar o que foi feito, o que irá fazer e os impedimentos que tivemos. Ao final de cada sprint realizaremos a reunião de revisão para que possamos verificar como foi finalizada as *tasks*, e em seguida executar a reunião de planejamento da *sprint*.

3.6 Cronograma das Atividades

Quadro 5: Cronograma das atividades

Atividade	Prazo
Desenvolvimento do front-end	25/01/2023 - 03/03/2023
Desenvolvimento do back-end	02/02/2023 - 31/03/2023
Teste do front-end	03/04/2023 - 12/05/2023
Teste do back-end	03/04/2023 - 12/05/2023
Integração back-end com front-end	15/04/2023 - 12/05/2023

4 RESULTADOS E DISCUSSÕES

Os resultados e discussões deste trabalho são fundamentais para analisar a eficácia do sistema de agendamento de horários desenvolvido para as coordenações dos cursos. Nesta seção, serão apresentadas as telas do sistema, acompanhadas de uma descrição de suas funcionalidades e dos casos de uso correspondentes. Essas informações fornecerão uma visão abrangente das capacidades do sistema e permitirão uma análise do que foi construído durante este trabalho. Vamos agora observar os resultados obtidos e iniciar a discussão sobre eles, com o objetivo de compreender melhor as contribuições e os desafios encontrados durante o desenvolvimento do sistema de agendamento de horários.

4.1 Listagem de Telas por Perfil de Usuário

Essa seção tem como objetivo apresentar as telas do sistema organizadas de acordo com os diferentes perfis de usuário. Serão descritas as telas comuns a todos os usuários, bem como as telas específicas para os perfis de aluno, coordenador e administrador.

A listagem das telas por perfil de usuário proporciona uma visão abrangente das funcionalidades disponíveis no sistema, destacando as características e permissões específicas de cada tipo de usuário. Isso permite uma melhor compreensão do fluxo de interação entre os usuários e o sistema.

A seção será dividida em sub-seções para cada perfil de usuário, iniciando com as telas comuns que são compartilhadas por todos os usuários. Em seguida, serão apresentadas as telas destinadas aos alunos e coordenadores, destacando as principais funcionalidades e recursos disponíveis para esses perfis. O mesmo procedimento será adotado para as telas dos administradores.

4.1.1 Telas Comuns Para Todos os Usuários

4.1.1.1 Página de *Login*

A tela de *login* é a primeira interface que os usuários encontram ao acessar o sistema. Nessa tela, eles são solicitados a inserir suas credenciais de acesso, como nome de usuário e senha, a fim de autenticar-se no sistema. O principal objetivo dessa tela é garantir a segurança e a privacidade das informações, permitindo o acesso apenas aos usuários autorizados.

Além disso, na tela de *login*, é possível observar a presença de dois links adicionais. O primeiro *link*(cadastrar) direciona os usuários para a tela de cadastro, caso seja o primeiro

acesso deles ao sistema. Essa opção permite que novos usuários se registrem e obtenham suas próprias credenciais de acesso.

O segundo *link*(recuperar senha) é destinado à recuperação de senha. Caso um usuário já cadastrado no sistema tenha esquecido sua senha, ele pode clicar nesse link para iniciar o processo de recuperação, seguindo as etapas necessárias para redefinir sua senha de acesso.

Essas funcionalidades complementares na tela de *login* visam fornecer aos usuários opções adicionais e soluções para possíveis cenários, como novos usuários e usuários que tenham esquecido suas senhas, melhorando assim a experiência do usuário e facilitando o acesso ao sistema.

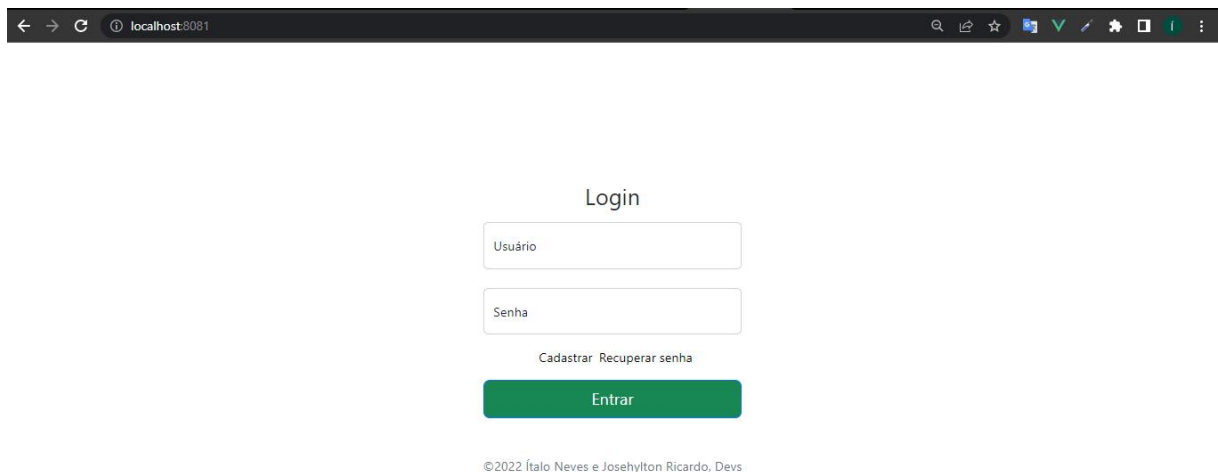


Figura 4: Tela de *login* (Imagem: Autoria própria).

4.1.1.2 Página de Cadastro de Usuário

A tela de cadastro é utilizada para registrar novos usuários no sistema. Ela apresenta campos nos quais os usuários podem inserir suas informações pessoais, como matrícula, e-mail, nome, sobrenome, senha, e um menu suspenso (*select*) que lista os cursos disponíveis para que o aluno possa selecionar o seu curso.

As informações inseridas pelos usuários nessa tela são armazenadas no banco de dados do sistema e são utilizadas posteriormente para fins de autenticação e controle de acesso. Após o registro bem-sucedido, os usuários poderão usar suas credenciais de acesso para fazer *login* no sistema e utilizar suas funcionalidades.

localhost:8081/cadastro

SISTEMA DE AGENDAMENTO DE HORÁRIOS PARA ATENDIMENTO COM A COORDENAÇÃO

Cadastro

Matrícula

Email

Nome

Sobrenome

Senha

Confirme a Senha

Curso
Selecione um curso

Cadastrar

© 2022 Ítalo Neves e Josehyllton Ricardo, Devs

Figura 5: Tela de cadastro (Imagem: Autoria própria).

4.1.1.3 Tela de “Recuperar Senha”

A tela de Recuperar Senha apresenta ao usuário a opção de recuperar sua senha caso a tenha esquecido. Nessa tela, o usuário encontra um campo *select* onde pode escolher entre duas opções para recuperar sua senha: por *e-mail* ou por matrícula. Após selecionar a opção desejada, o usuário deve clicar no botão Enviar. Em seguida, será enviado um *email* para o endereço de *e-mail* cadastrado no sistema, contendo sua senha de recuperação. Essa senha de recuperação permitirá que o usuário acesse o sistema novamente. Ao acessar o sistema com a senha de recuperação, o usuário pode redefinir senha por uma de sua escolha.

SISTEMA DE AGENDAMENTO DE HORÁRIOS PARA ATENDIMENTO COM A COORDENAÇÃO

Recuperar Senha

Escolha como deseja recuperar sua senha:

Email

Email:

Enviar

© 2022 Ítalo Neves e Josehyllton Ricardo, Devs

Figura 6: Tela de recuperar senha (Imagem: Autoria própria).

4.1.1.4 Página de Boas Vindas

Após o *login* bem-sucedido, os usuários são redirecionados para a tela *home*, que exibe uma mensagem de boas-vindas. Nessa tela, há um botão denominado "Começar a usar" que leva o usuário para a tela correspondente às suas permissões de acesso.

Para os usuários com perfil de aluno ou coordenador, ao clicar no botão "Começar a usar", eles serão direcionados para a tela onde podem visualizar seus agendamentos. No caso de um usuário com perfil de administrador, ao clicar no botão "Começar a usar", ele será redirecionado para uma tela que lista todos os usuários do Sistema.

Falarei mais detalhadamente sobre as telas dos alunos, coordenadores e administradores nos tópicos abaixo.



Figura 7: Tela de boas-vindas (Imagem: Autoria própria).

4.1.1.5 Tela “Sobre”

Nessa tela, são apresentados o propósito do sistema, os desenvolvedores responsáveis pelo projeto e os principais recursos e benefícios oferecidos aos usuários, fornecendo detalhes sobre a origem e as funcionalidades do sistema desenvolvido.

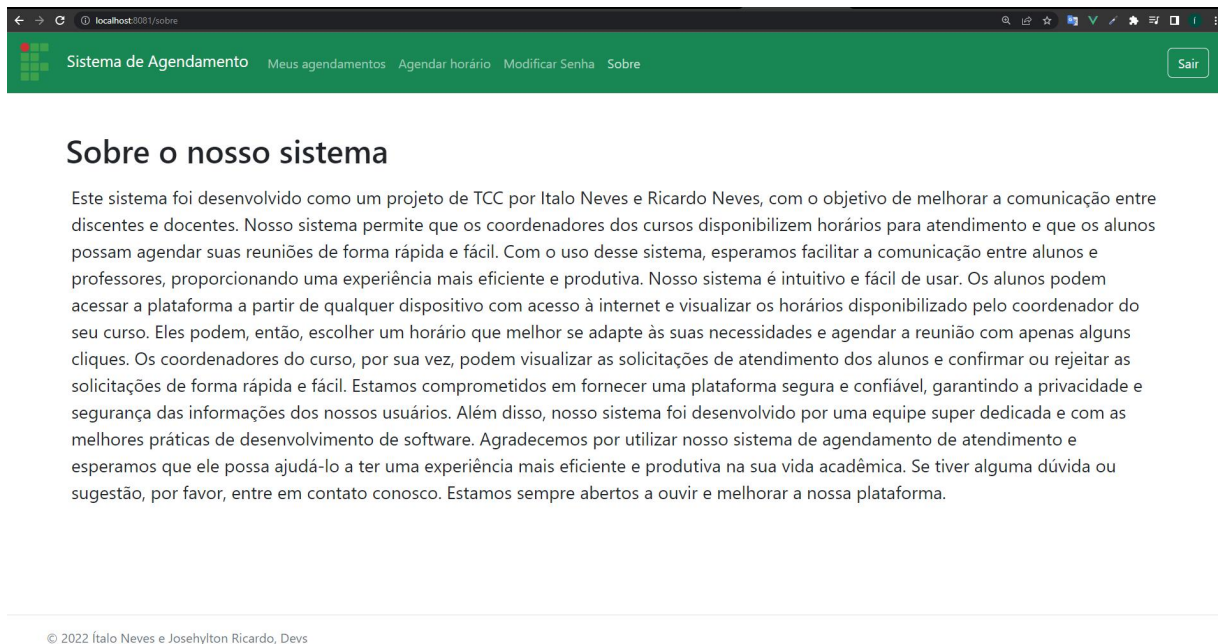


Figura 8: Tela “Sobre” (Imagem: Autoria própria).

4.1.1.6 Tela de “Erro”

A tela de erro é exibida quando o usuário acessa uma rota inexistente, não tem permissão para acessá-la ou quando ocorre um erro inesperado durante a interação com o sistema. O objetivo dessa tela é informar ao usuário que algo não aconteceu conforme o esperado e fornecer uma mensagem genérica de erro.

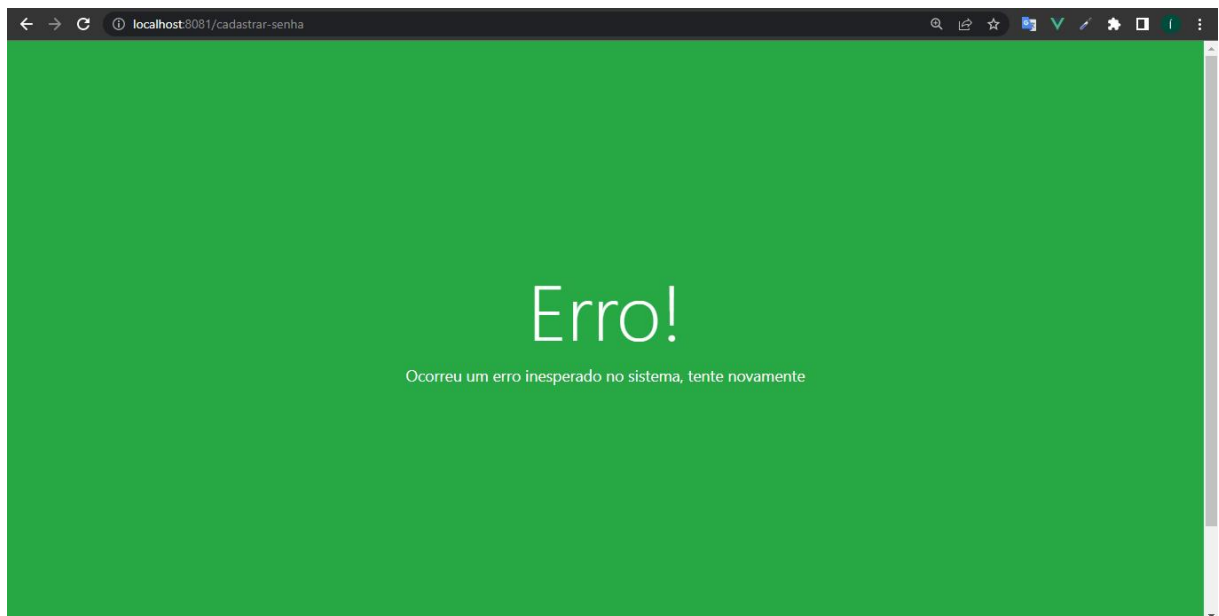


Figura 9: Tela de “Erro” (Imagem: Autoria própria).

4.1.2 Telas dos Alunos e Coordenadores

Esta seção, tem como objetivo apresentar as telas compartilhadas por esses dois perfis de usuários, uma vez que elas são as mesmas, diferindo apenas nas regras de negócio aplicadas. Aqui serão descritas as telas que englobam as funcionalidades e recursos disponíveis tanto para os alunos quanto para os coordenadores.

4.1.2.1 Tela “Meus Agendamentos”

A tela “Meus Agendamentos” apresenta um cabeçalho comum para alunos e coordenadores, diferenciando-se apenas pelo texto do *link* presente no cabeçalho. Para os alunos, o *link* é exibido como “Agendar Horário”, enquanto para os coordenadores, ele é apresentado como “Disponibilizar Horários”. Essa tela desempenha um papel essencial ao fornecer uma visão abrangente dos agendamentos realizados ou horários reservados pelos usuários.

Ao acessar a tela, os alunos encontrarão uma lista completa de todos os agendamentos que realizaram. Cada entrada na lista é acompanhada pela data do agendamento, bem como pelo horário inicial e final. Isso permite aos alunos visualizarem facilmente os detalhes de suas reuniões agendadas, permitindo um gerenciamento eficiente do tempo e um acompanhamento adequado de suas obrigações acadêmicas.

Por outro lado, quando um usuário com perfil de coordenador acessa a tela "Meus Agendamentos", são apresentados todos os horários que os alunos agendaram com ele. Assim como na lista de agendamentos dos alunos, cada entrada nessa lista de horários é acompanhada pela data e pelos horários iniciais e finais. Essa visualização facilita a organização e o planejamento do coordenador, permitindo que ele esteja preparado e disponível para atender aos alunos nos horários agendados e visualizar os atendimentos realizados.

Essa tela foi projetada para fornecer uma experiência clara e organizada para todos os usuários. Com informações essenciais, como data e horários, ela possibilita que os usuários acompanhem seus agendamentos de forma eficiente. Seja para os alunos visualizarem suas reuniões agendadas ou para os coordenadores gerenciarem os horários reservados pelos alunos, essa tela desempenha um papel fundamental na centralização e no acesso rápido às informações relevantes sobre os agendamentos no sistema.

#	Data	Hora Inicial	Hora Final
1	21/04/2023	15:00	15:30
2	21/04/2023	16:30	17:00
3	27/04/2023	07:00	07:30
4	27/04/2023	08:30	09:00

© 2022 Italo Neves e Josehyton Ricardo, Devs

Figura 10: Tela “Meus Agendamentos” (Imagem: Autoria própria).

4.1.2.2 Telas “Disponibilizar Horários” e “Agendar Horários”

A tela "Disponibilizar Horários" é acessada apenas pelo coordenador do curso e permite que ele selecione uma data específica ou um período para disponibilizar horários de atendimento. Visualmente, essa tela é semelhante à tela "Agendar Horários". No entanto, a diferença está na regra de negócio aplicada.

Por outro lado, tanto o coordenador quanto o aluno têm acesso a três tipos de visualização do calendário: semanal, mensal e diário. Essa configuração permite uma visão abrangente dos horários disponíveis e facilita o planejamento e agendamento das reuniões.

Quando um aluno acessa a tela por meio do link presente no cabeçalho, denominado "Agendar Horários", ele visualiza os horários disponibilizados pelo coordenador do seu curso. Os horários disponíveis são exibidos em verde na tela, e o aluno pode clicar em cima de um horário disponível para agendar uma reunião. Em seguida, um modal é aberto, apresentando os detalhes do horário selecionado, como data e horário, e perguntando ao aluno se ele deseja confirmar a reserva. O aluno tem a opção de confirmar ou cancelar o agendamento.

Caso o aluno decida cancelar um horário agendado, ele pode retornar à mesma tela, onde o horário previamente agendado será exibido em vermelho. Ao clicar nesse horário, o coordenador do curso será notificado por *e-mail* sobre o cancelamento do atendimento. Por outro lado, se o aluno concluir o agendamento, ele receberá um *e-mail* de confirmação com os detalhes do seu agendamento.

Dessa forma, as telas de "Disponibilizar Horários" e "Agendar Horários" possuem uma aparência visual semelhante, mas diferem nas funcionalidades e regras de negócios aplicadas.

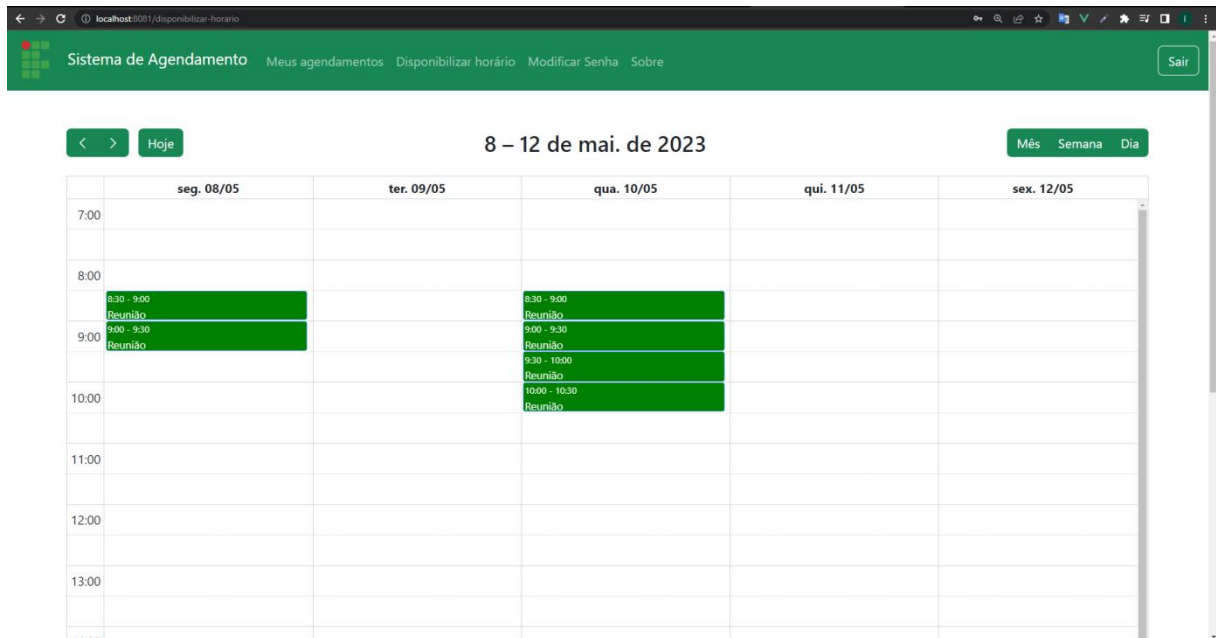


Figura 11: Tela “Disponibilizar horários” área do coordenador (Imagem: Autoria própria).

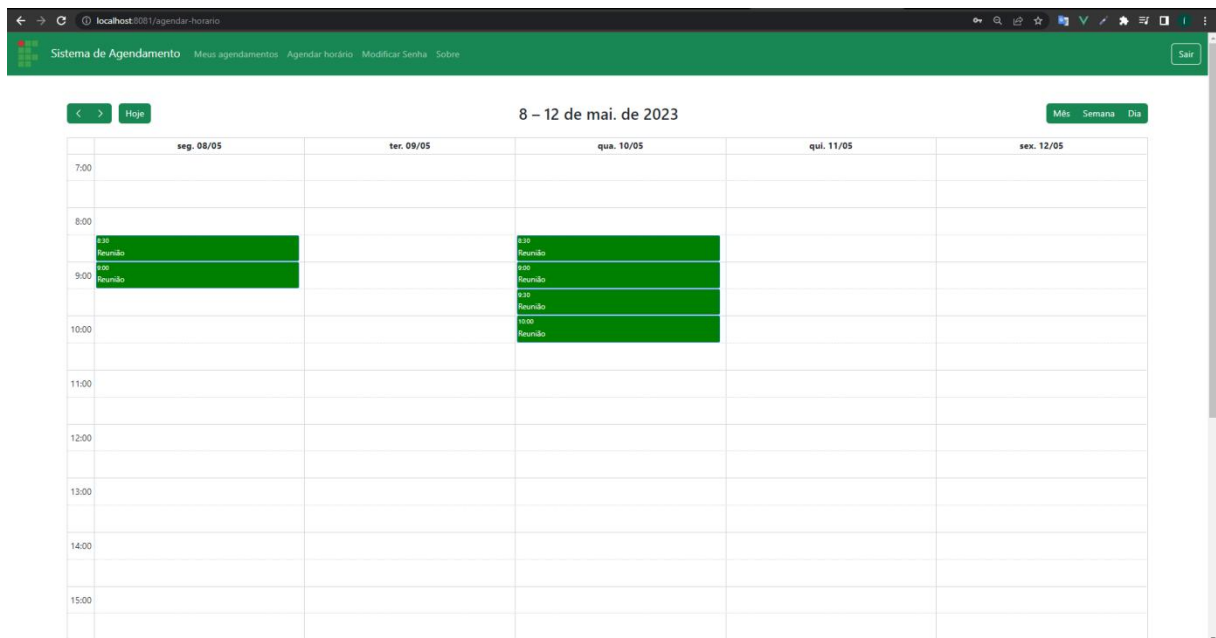


Figura 12: Tela “Agendar horário” área do aluno (Imagem: Autoria própria).

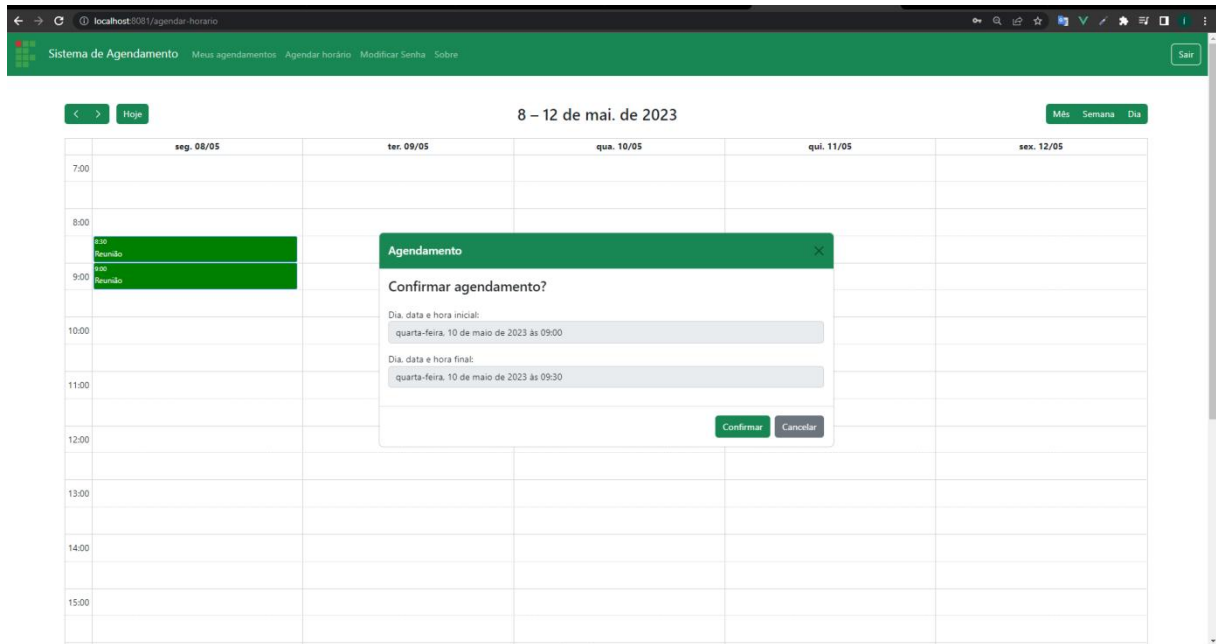


Figura 13: Tela “Agendar horário” com modal de confirmação presente após usuário selecionar data (Imagem: Autoria própria).

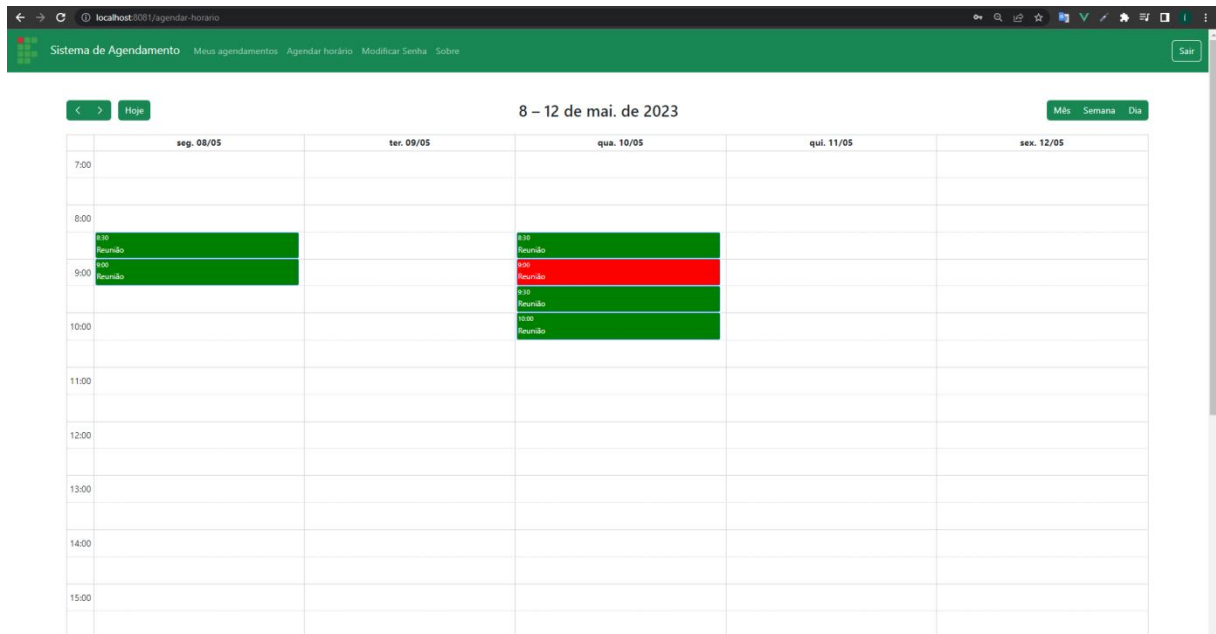


Figura 14: Tela “Agendar horário” após aluno confirmar agendamento (Imagem: Autoria própria).

4.1.2.3 Tela de “Redefinir Senha”

Nessa tela, o usuário encontra três campos: um para digitar a senha atual e dois campos para inserir a nova senha. Essa funcionalidade permite que o usuário atualize sua senha de acesso ao sistema de forma segura e fácil.

Ao preencher os campos, é realizada uma verificação para garantir que a senha atual informada seja válida e correspondente à senha cadastrada no sistema. Além disso, é feita uma comparação entre as duas novas senhas digitadas para garantir que coincidam, evitando erros de digitação ou inconsistências.

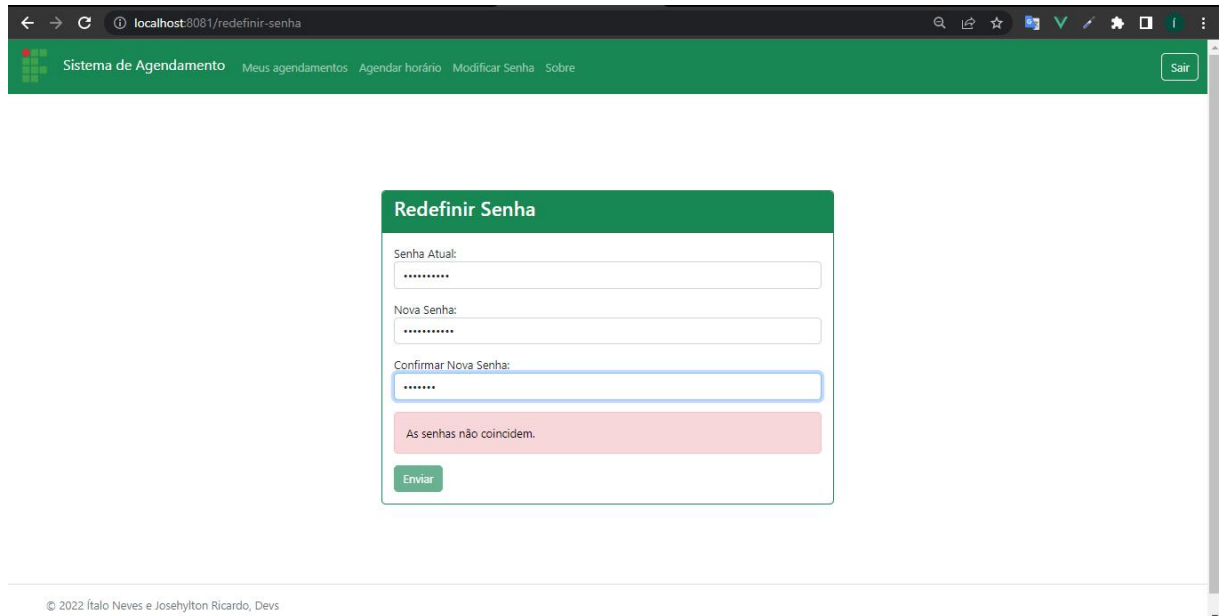


Figura 15: Tela “Redefinir Senha” área do aluno e coordenador (Imagem: Autoria própria).

4.1.3 Telas do Administrador

Nesta seção, exploraremos as telas projetadas exclusivamente para os administradores do sistema. Com funcionalidades e recursos específicos para gerenciamento e controle, essas telas oferecem aos administradores as ferramentas necessárias para supervisionar e administrar efetivamente o sistema. Acompanhe a descrição detalhada das telas disponíveis para os administradores, destacando suas principais características e funcionalidades.

4.1.3.1 Tela “Gerenciar Usuários”

A tela "Gerenciar Usuários" é a primeira tela que os administradores visualizam após a tela de boas-vindas. Nessa tela, os administradores têm acesso a uma lista completa de usuários do sistema, onde podem realizar diversas ações de gerenciamento. Os administradores podem filtrar os usuários por nome, perfil (coordenador ou aluno) e curso por meio dos filtros disponíveis na tela.

Cada usuário é apresentado com opções de interação, como um botão de exclusão, que permite que o administrador remova um usuário do sistema. Além disso, é possível alterar as permissões dos usuários por meio do campo *select*, permitindo que o administrador transforme um usuário de aluno para coordenador. Essa alteração de permissões é importante,

pois, inicialmente, ao se cadastrar no sistema, um usuário é registrado como aluno automaticamente.

Essa funcionalidade de alterar as permissões é restrita ao administrador, garantindo que apenas ele possa definir um usuário como coordenador, evitando que usuários comuns obtenham esse acesso privilegiado. Isso assegura a integridade e o controle adequado do sistema, mantendo a devida autoridade sobre as funções de coordenação.

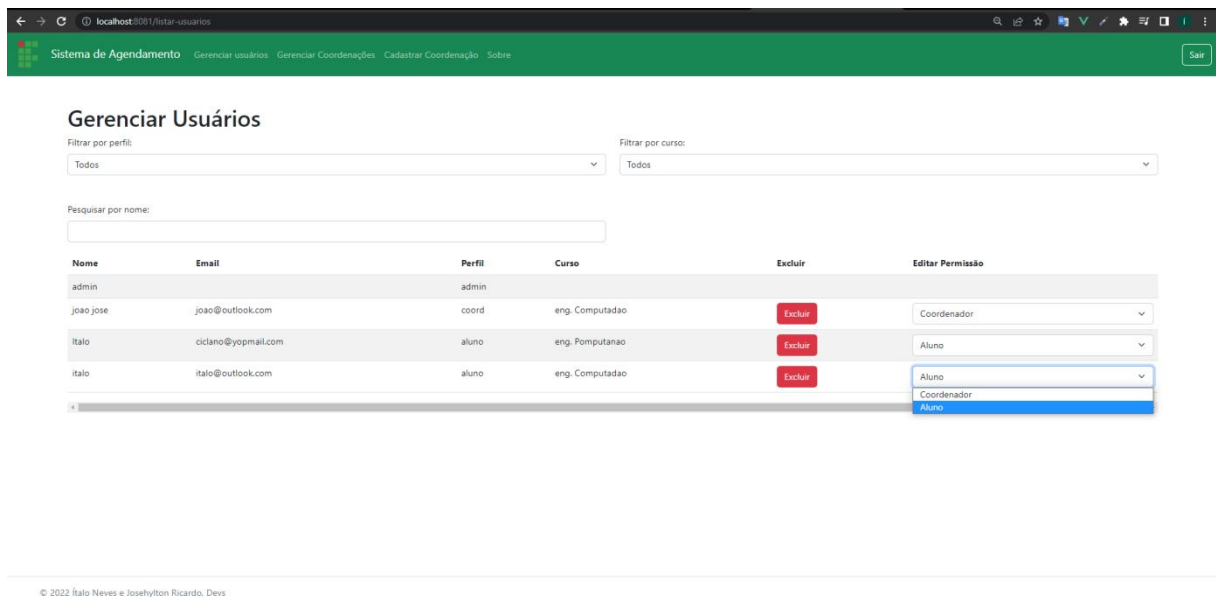


Figura 16: Tela “Gerenciar Usuários” (Imagem: Autoria própria).

4.1.3.2 Tela “Gerenciar Coordenações”

A tela "Gerenciar Coordenações" pode ser acessada pelo usuário ao clicar no *link* "Gerenciar Coordenações" presente no cabeçalho da página. Ao acessar essa tela, o usuário tem uma lista completa de todos os cursos da instituição, juntamente com as informações dos seus respectivos coordenadores, como *e-mail*, matrícula e nome. Cada coordenação é exibida com opções de ação, incluindo um botão para exclusão, que permite remover uma coordenação do sistema. Além disso, há também um botão de edição, que direciona o usuário para uma tela específica onde é possível alterar o coordenador do curso. Essa funcionalidade proporciona ao administrador a flexibilidade de gerenciar as coordenações de forma eficiente, permitindo a edição dos responsáveis por cada curso quando necessário.

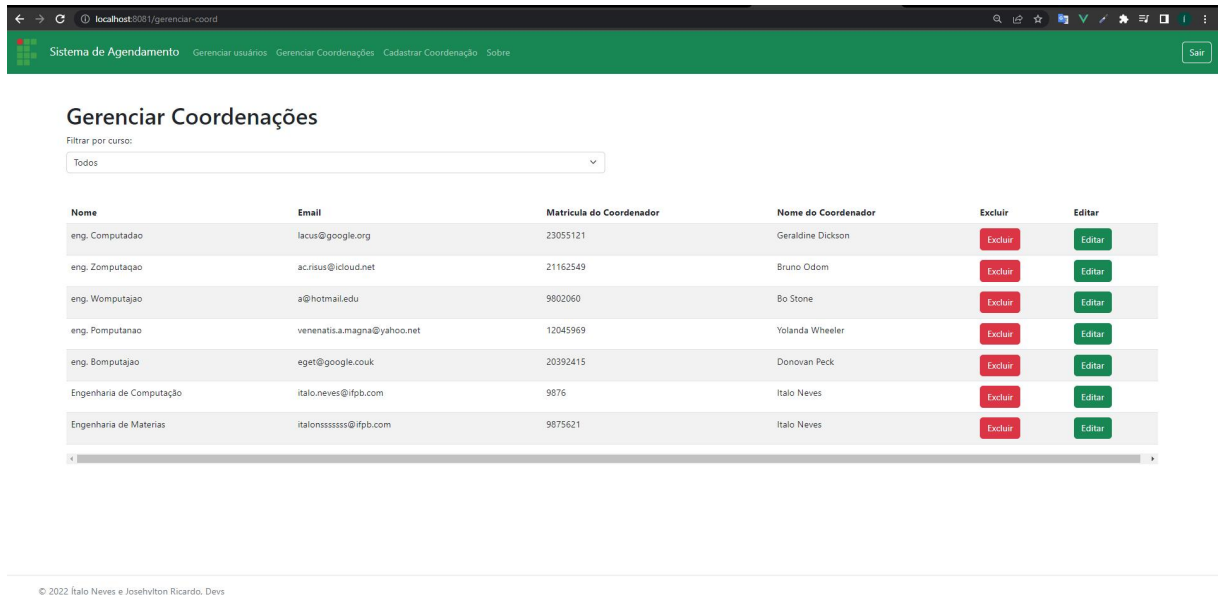


Figura 17: Tela “Gerenciar Coordenações” (Imagem: Autoria própria).

4.1.3.3 Tela “Cadastrar Coordenação”

A tela "Cadastrar Coordenação" oferece ao administrador a funcionalidade de cadastrar uma nova coordenação no sistema. Para isso, o administrador deve preencher as informações necessárias, que incluem o nome do curso e os dados do coordenador, como matrícula, nome e *e-mail* da coordenação. Essa tela permite ao administrador adicionar uma nova coordenação de forma simples e organizada, garantindo a correta associação entre o curso e o coordenador responsável.

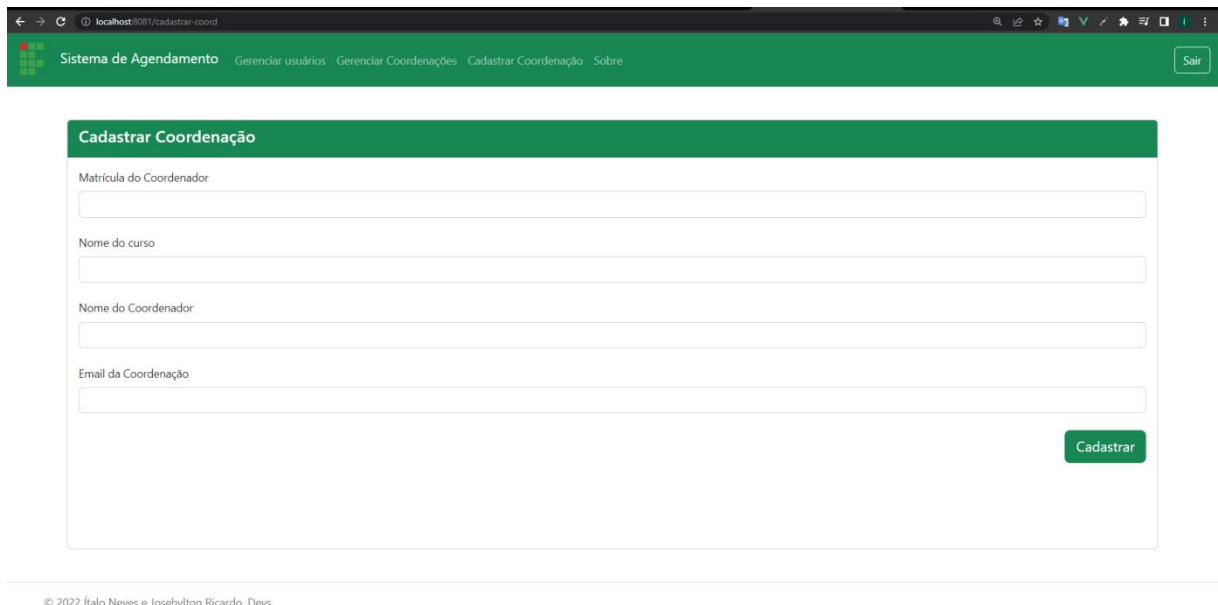


Figura 18: Tela “Cadastrar Coordenação” (Imagem: Autoria própria).

4.1.3.4 Tela “Editar Coordenador”

A tela "Editar Coordenador" é acessada por meio do botão "Editar" na tela de "Gerenciar Coordenações". Essa tela possui uma estrutura semelhante à tela de "Cadastrar Coordenação", com os campos preenchidos com o nome do curso e as informações do coordenador atual. Nessa tela, o administrador tem a possibilidade de editar o coordenador do curso, permitindo a atualização das informações do responsável pela coordenação. Essa funcionalidade oferece ao administrador a flexibilidade de gerenciar e manter atualizadas as informações dos coordenadores de cada curso.

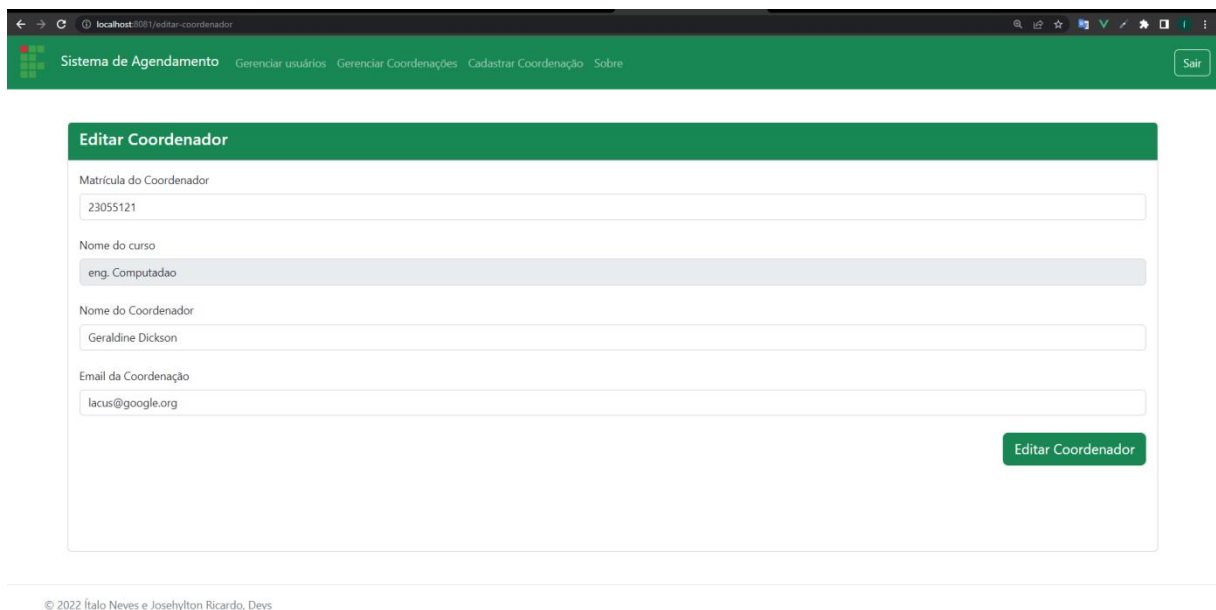


Figura 19: Tela “Editar Coordenação” (Imagem: Autoria própria).

4.2 Projeto Arquitetural do Sistema

Esta seção tem como objetivo fornecer uma visão geral do projeto arquitetural do sistema SAHC, com foco no funcionamento dos pacotes relacionados ao *back-end*. Serão apresentadas informações sobre a estrutura, organização e interações entre os componentes que compõem o sistema. Essa explanação detalhada permitirá uma melhor compreensão do funcionamento interno do sistema.

O SAHC é um sistema desenvolvido e projetado para atender às necessidades específicas da aplicação. O *back-end* desempenha um papel fundamental na manipulação dos dados, regras de negócio e integração com banco de dados. A imagem abaixo mostra a estrutura de pacotes do *back-end* do sistema SAHC.

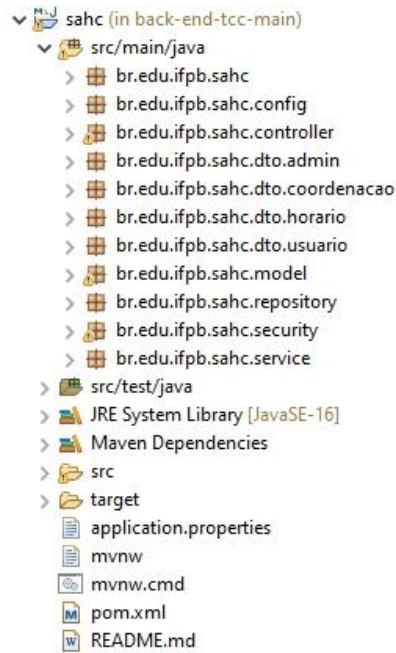


Figura 21: Estrutura de Pacotes do back-end (Imagem: Autoria própria).

Quando o usuário tenta realizar o *login* no sistema o *Spring Security* faz uma verificação para saber se o usuário em questão está cadastrado no sistema. O pacote responsável por esta verificação é o pacote “br.edu.ifpb.sahc.security”. Autenticado o usuário, então o sistema retorna um *token JWT* válido. De posse deste *token* o usuário pode enviar requisições informando o *token* no *header* da requisição.

Como o sistema não guarda a sessão do usuário, é necessário que um *token* válido seja enviado no *header* da requisição, o pacote responsável por validar este *token* é o “br.edu.ifpb.sahc.security”. autorizando o usuário ao acesso da funcionalidade desejada.

Quando uma requisição chega na aplicação SAHC e está devidamente autorizada pelo *Spring Security*, o pacote responsável por receber esta requisição é o “br.edu.ifpb.sahc.controller”, inicialmente o *Spring Validation* realiza uma validação dos campos informados na requisição. Validação OK, o pacote controller direciona a requisição para o pacote “br.edu.ifpb.sahc.service”.

O pacote “br.edu.ifpb.sahc.service”, é responsável por processar tais informações que chegam no sistema SAHC. Este pacote é a regra de negócio. Quando as informações são processadas corretamente, então é chegado o momento de persistir os dados. O pacote “br.edu.ifpb.sahc.repository” se faz necessário nesse processo, pois o pacote de serviços faz uso do pacote repository para efetivar a persistência dos dados.

Outro pacote de extrema importância é o “br.edu.ifpb.sahc.model”. O pacote model é responsável pela modelagem do banco de dados, é onde encontramos as entidades do banco de dados.

No pacote “br.edu.ifpb.sahc.config” se encontra as classes de configuração do sistema, Classes com configuração do *swagger*, configuração de administrador, configuração de segurança, estão todas contidas neste pacote.

Por ultimo e não menos importante os pacotes “br.edu.ifpb.sahc.dto.coordenacao”, “br.edu.ifpb.sahc.dto.horario”, “br.edu.ifpb.sahc.dto.admin”, “br.edu.ifpb.sahc.dto.usuario”, fornecer uma estrutura simplificada e otimizada para transferir apenas os dados necessários entre as camadas ou componentes do sistema.

Na estrutura existe outros arquivos essenciais, são eles: o arquivo de propriedades “application.properties”, onde comporta propriedades necessárias ao sistema, como por exemplo, a conexão do banco de dados e o arquivo “pom.xml”, onde contém informações de bibliotecas e dependências usadas no projeto.

5 CONCLUSÃO

Neste trabalho, foi desenvolvido um Sistema de Agendamento de Horário para Coordenação (SAHC), que visa auxiliar na comunicação entre discentes e coordenações do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - IFPB Campus Campina Grande. O SAHC oferece uma alternativa ao método tradicional de comunicação via *e-mail*, permitindo que os alunos agendem horários de atendimento com as coordenações de forma mais fácil.

Durante o desenvolvimento do sistema, foi definido objetivo geral e específicos, incluindo a elaboração do projeto arquitetural, formulação do banco de dados, o desenvolvimento do sistema web e a realização de testes de integração. As principais tecnologias utilizadas no desenvolvimento incluíram o Spring Boot, Spring Data e Spring Security para o servidor, e o Vue.js para o cliente.

Os benefícios do SAHC incluem a praticidade na administração dos horários de atendimento da coordenação, a facilidade de acesso dos alunos e da coordenação e a simplificação da interação entre ambas as partes. O sistema também se destaca por utilizar Spring framework que normalmente apresenta um desempenho satisfatório diante de um grande número de requisições de usuários, sem apresentar lentidão significativa na emissão de respostas.

É importante ressaltar que não foram conduzidos testes com usuários reais e coordenações reais em um ambiente institucional real. Além disso, a coleta de requisitos foi realizada exclusivamente pelos autores deste trabalho, sem a participação direta da coordenação. Embora tenhamos nos esforçado para criar um ambiente que simule as condições reais e tenhamos realizado uma análise cuidadosa para identificar e listar todos os requisitos necessários, é necessário considerar que essa falta de interação com usuários reais e a coordenação pode representar um desafio para a validação do sistema.

A API da solução desenvolvida pode suportar acesso através de dispositivos móveis, visto que, a construção da sua arquitetura e as tecnologias utilizadas podem simplificar a integração com frameworks de dispositivos móveis.

Em suma, o SAHC representa uma contribuição significativa para o agendamento, visualização e organização de horários de atendimento da coordenação. O sistema desenvolvido neste trabalho está pronto para ser implantado, o que viabiliza a realização de testes em um ambiente institucional real, com a participação de usuários reais. Essa etapa permitirá identificar possíveis melhorias, exclusões ou criação de novos requisitos com base

no *feedback* dos usuários. Os repositórios utilizados no *back-end* e *front-end* serão entregues como parte integrante deste trabalho.

5.1 Trabalhos Futuros

Nesta seção serão apresentadas algumas sugestões de melhorias para o sistema e algumas ideias que foram pensadas no início do projeto, porém não foram incluídas no sistema atual e podem ser implementadas em trabalhos futuros.

- No *front-end* adicionar um modal de confirmação ao desalocar um horário agendado. Atualmente quando o aluno deseja desalocar um horário o mesmo é desagendado sem nenhuma confirmação da parte do aluno. O objetivo dessa melhoria é fornecer uma camada adicional de segurança e evitar possíveis desagendamentos acidentais;
- Validar o sistema em um ambiente real, com usuários reais;
- Criar um aplicativo *mobile* para dar mais flexibilidade para os usuários;
- Analisar a viabilidade de integração com o SUAP para realização da autenticação;
- Envio de lembrete via *e-mail* para as partes interessadas alguns minutos antes do horário agendado;
- Inclusão de duas colunas na tela de agendamentos, na área destinada ao coordenador. Estas colunas teriam o objetivo de apresentar informações relevantes sobre cada agendamento realizado pelos alunos, especificamente o nome do aluno e o assunto que será tratado;

Apêndice A : Guia de instalação do SAHC

Este apêndice fornece instruções detalhadas sobre como realizar a instalação do projeto, abrangendo tanto o back-end quanto o front-end do aplicativo. A instalação adequada do projeto é essencial para garantir o correto funcionamento do sistema.

Aqui você encontrará todas as etapas necessárias para configurar o ambiente de desenvolvimento e executar os comandos adequados para a instalação das dependências do projeto. As instruções fornecidas foram testadas e são compatíveis com as versões mais recentes do projeto.

A.1 Instalação do Back-end

O back-end foi construído utilizando o framework Spring Boot, na versão 2.5.1, e requer o Java versão 17. Além disso, ela utiliza o banco de dados MySQL para armazenar os dados.

Para realizar a instalação e execução do back-end, siga os passos abaixo:

1. Certifique-se de ter o JDK na versão 17 instalado em sua máquina. Você pode fazer o download e instalar a versão do Java Development Kit (JDK) no site oficial [java.com](https://www.java.com).
2. Faça o clone do repositório do back-end a partir do seguinte endereço:
<https://github.com/italoneves/back-end-tcc>

Você pode utilizar o seguinte comando no terminal ou utilizar alguma interface gráfica do Git.
`git clone https://github.com/italo/back-end-tcc`

3. Após clonar o repositório, abra o projeto na IDE de sua preferência, como Eclipse, IntelliJ, NetBeans, ou outra que você esteja habituado.
4. Verifique se você possui o MySQL instalado e configurado em sua máquina. Caso não tenha, faça o download e instalação do MySQL Server a partir do site oficial <https://www.mysql.com/>
5. No arquivo de configuração do projeto, chamado `application.properties`, estão as configurações do banco de dados MySQL. Lá você pode encontrar ou fornecer informações como endereço do servidor, porta, nome do banco de dados, nome de usuário e senha.
6. Execute o projeto em sua IDE, e ele será iniciado no endereço <http://localhost:8080>.

A.2 Instalação do Front-end

Depois de concluir a instalação do back-end, vamos prosseguir com a instalação do front-end.

Para executar o projeto do front-end, siga as etapas abaixo:

1. Faça um clone do repositório do front-end do TCC. Você pode encontrar o repositório em: <https://github.com/italoneves/front-end-tcc>
2. Certifique-se de ter o Node.js instalado em sua máquina. Você pode fazer o download e instalá-lo a partir do site oficial <https://nodejs.org>.
3. Abra um terminal e navegue até o diretório do projeto front-end clonado e execute o seguinte comando para instalar as dependências do projeto: `npm install`
4. Após a conclusão da instalação das dependências, você estará pronto para executar o projeto front-end. Utilize o seguinte comando: `npm run serve`

O projeto será compilado e o servidor de desenvolvimento será iniciado. Você poderá acessar o aplicativo no seu navegador através do endereço `http://localhost:8081`.

A.3 Acessando o Swagger

Para acessar o swagger é necessário que o back-end esteja funcionando. Caso o back-end esteja funcionando localmente o link de acesso para o swagger é `http://localhost:8080/swagger-ui/#/`, se não o link de acesso será o host mais `/swagger-ui/#/`.

REFERÊNCIAS

CALAIS, Raphael. **Descubra os Doze Princípios do Manifesto Ágil**. Performa_it. 2011. Disponível em: <https://performait.com/descubra-os-doze-principios-do-manifesto-agil/>. Acesso em: 3 dez. 2022.

GARCIA, Fabrício W. da S. *et al.* **Desenvolvimento e Avaliação de um Módulo Educacional para o Ensino da Língua Portuguesa**: Um Estudo Baseado em Variações Linguísticas. ResearchGate. Castanhal, 2013. 10 p. Disponível em: https://www.researchgate.net/publication/337991059_II_Congresso_Brasileiro_d_e_Recursos_Digitais_na_Educacao. Acesso em: 10 mai. 2023.

GOVERNO FEDERAL. **90% dos lares brasileiros já tem acesso à internet no Brasil, aponta pesquisa**. gov.br. 2022. Disponível em: <https://www.gov.br/casacivil/pt-br/assuntos/noticias/2022/setembro/90-dos-lares-brasileiros-ja-tem-acesso-a-internet-no-brasil-aponta-pesquisa>. Acesso em: 28 nov. 2022.

HAYERBEKE, Marijn. *Eloquent JavaScript: A Modern Introduction to Programming*. 3 ed. No Starch Press, 2018. 472 p.

INCAU, Caio. **Vue.js construa aplicações incríveis**. Ed. 1. São Paulo: Casa do Código 20 de Abril 2017.

MAVEN. **Welcome to Apache Maven**. Apache Maven Project. 2023. Disponível em: <https://maven.apache.org/>. Acesso em: 11 mai. 2023.

MICROSOFT. **O que é o Java Spring Boot?**: Uma introdução ao Spring Boot: a ferramenta popular baseada em Java para o desenvolvimento de aplicativos Web e microsserviços. Microsoft. 2022. Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-java-spring-boot/>. Acesso em: 25 nov. 2022.

MICROSOFT. **Manage the Shifts app for your organization in Microsoft Teams**. Microsoft Teams. 2023. Disponível em: <https://learn.microsoft.com/en-us/microsoftteams/expand-teams-across-your-org/shifts/manage-the-shifts-app-for-your-organization-in-teams>. Acesso em: 11 mai. 2023

MOZILLA. **Introdução ao lado servidor**: O que é programação de site do lado do servidor ?. Developer Mozilla. 2022. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/First_steps/Introduction. Acesso em: 29 nov. 2022.

NEVES, Lucas & CARVALHO, Átila & da HORA, Henrique. Baúru, 2020.
APLICAÇÕES WEB CLIENT-SIDE BASEADAS EM JAVASCRIPT: UMA ANÁLISE BIBLIOMETÉTRICA. XXVII Simpósio de Engenharia de Produção,
Disponível em:

https://www.researchgate.net/publication/347949078_APLICACOES_WEB_CLIENT-SIDE_BASEADAS_EM_JAVASCRIPT_UMA_ANALISE_BIBLIOMETETRICA

OLIVEIRA, Ingrid. **Levantamento mostra que ataques cibernéticos no Brasil cresceram 94%**: País é o 2º na América Latina com mais ataques cibernéticos em 2022. CNN. 2022. Disponível em: <https://www.cnnbrasil.com.br/tecnologia/levantamento-mostra-que-ataques-ciberneticos-no-brasil-cresceram-94/#:~:text=Levantamento%20mostra%20que%20ataques%20cibern%C3%A9ticos%20no%20Brasil%20cresceram%2094%25,-Pa%C3%ADs%20%C3%A9%20o&text=O%20Brasil%20registrou%20no%20primeiro,16%2C2%20bilh%C3%B5es%20de%20registros..> Acesso em: 28 nov. 2022.

ORACLE. **O banco de dados de código aberto mais popular do mundo**. MySQL. 2023. Disponível em: <https://www.mysql.com/>. Acesso em: 11 mai. 2023.

SCHILDT, Herbert. **Java: The Complete Reference**. 11 ed. New York, NY McGraw-Hill Education, 2018. 1248 p

SOMMERVILLE, IAN. **ENGENHARIA DE SOFTWARE**. 10 ed. São Paulo: Pearson Universidades, 2019. 768 p.

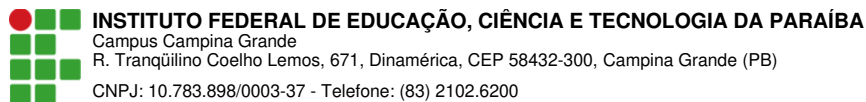
SOUZA, W. G. de; PINTO, G. S. **A UTILIZAÇÃO DO VUE.JS COM UMA API REST EM SPRING BOOT**. Revista Interface Tecnológica, [S. l.], v. 17, n. 2, p. 155–167, 2020. DOI: 10.31510/infa.v17i2.904. Disponível em: <https://revista.fatectq.edu.br/interfacetecnologica/article/view/904>. Acesso em: 3 dez. 2022.

TECNICON. **Metodologia Scrum para a gestão de processos ágeis na indústria**. 2019. Disponível em: https://www.tecnicon.com.br/blog/411-Metodologia_Scrum_para_a_gestao_de_processos_ageis_na_industria. Acesso em: 3 dez. 2022.

VMWARE. **Spring Framework**: Visão Geral. Spring. Spring.io, 2022. Disponível em: <https://spring.io/projects/spring-framework/>. Acesso em: 27 nov. 2022.

VMWARE. **Spring Security**: Visão Geral. Spring. Spring.io, 2022. Disponível em: <https://spring.io/projects/spring-security#overview>. Acesso em: 29 nov. 2022.

VMWARE. **Spring Data**: Visão Geral. Spring. Spring.io, 2022. Disponível em: <https://spring.io/projects/spring-data>. Acesso em: 1 dez. 2022.



Documento Digitalizado Ostensivo (Público)

Versão Final do TCC

Assunto: Versão Final do TCC
Assinado por: Italo Neves
Tipo do Documento: Anexo
Situação: Finalizado
Nível de Acesso: Ostensivo (Público)
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- **Ítalo Santos Neves, ALUNO (201621250013) DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO - CAMPINA GRANDE**, em 20/06/2023 14:17:39.

Este documento foi armazenado no SUAP em 20/06/2023. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 859196
Código de Autenticação: 1442e87d8b

