

FACULDADE SENAC FLORIANÓPOLIS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

DANILO DE JESUS SANTOS
GABRIEL EDUARDO DOS SANTOS GARBIN
GUSTAVO CARDOSO DA COSTA
LEONARDO GONÇALVES DUARTE DA SILVA

STM:
SISTEMA DE TRANSFERÊNCIA DE MATERIAIS

Florianópolis

2023

DANILO DE JESUS SANTOS
GABRIEL EDUARDO DOS SANTOS GARBIN
GUSTAVO CARDOSO DA COSTA
LEONARDO GONÇALVES DUARTE DA SILVA

STM:
SISTEMA DE TRANSFERÊNCIA DE MATERIAIS

Trabalho apresentado à Faculdade Senac Florianópolis
como requisito parcial para a obtenção do título de
Tecnólogo em Análise e Desenvolvimento de Sistema.

Orientador(a): Msc Thayse Christiane da Silva.

Florianópolis

2023

DANILO DE JESUS SANTOS
GABRIEL EDUARDO DOS SANTOS GARBIN
GUSTAVO CARDOSO DA COSTA
LEONARDO GONÇALVES DUARTE DA SILVA

STM:
SISTEMA DE TRANSFERÊNCIA DE MATERIAIS

Trabalho apresentado à Faculdade Senac Florianópolis
como requisito parcial para a obtenção do título de
Tecnólogo em Análise e Desenvolvimento de Sistema.

Thayse Cristine da Silva (Orientadora) – Senac

Luciano José Kogut - Senac

Rafael Queiroz Gonçalves - Senac

Wallace Eduardo Mota – Senac

Florianópolis, 07 de julho de 2023

RESUMO

A gestão eficiente das transferências de materiais entre sedes de uma mesma empresa requer uma abordagem logística bem estruturada e uma comunicação fluída entre o setor responsável pela entrega e os solicitantes dos materiais. A falta de uma gestão adequada dos pedidos e do agendamento das entregas pode resultar em deslocamentos desnecessários, gastos excessivos e atrasos no transporte, prejudicando a empresa como um todo. Diante desse cenário, buscamos coletar informações detalhadas sobre o problema e, por meio de pesquisas e questionários, reunimos sugestões valiosas para encontrar uma solução. Com base nesse conhecimento adquirido, desenvolvemos um sistema web projetado especificamente para gerenciar as transferências de materiais. Esse sistema abrange funções como agendamento de viagens, cadastro de sedes, motoristas, veículos e materiais, além de notificações via e-mail e geração de relatórios para os motoristas. O objetivo primordial é aprimorar a gestão e a comunicação dessas viagens para benefício das empresas e de seus usuários.

Palavras-chave: Transferências de materiais; Gestão Logística; Comunicação.

ABSTRACT

Efficient management of material transfers between branches of the same company requires a well-structured logistic approach and smooth communication between the delivery department and material requestors. Inadequate management of orders and delivery scheduling can lead to unnecessary travel, excessive expenses, and transportation delays, negatively impacting the entire company. Given this scenario, we strive to gather detailed information about the problem and, through surveys and questionnaires, collect valuable suggestions to find a solution. Based on this acquired knowledge, we have developed a web-based system specifically designed to manage material transfers. This system encompasses features such as trip scheduling, branch registration, driver and vehicle management, material tracking, email notifications, and generating reports for drivers. The primary goal is to enhance the management and communication of these trips for the benefit of companies and their users.

Keywords: Material Transfers; Logistic Management; Communication.

LISTA DE FIGURAS

Figura 1 - Tabela Comparativa.....	15
Figura 2 - Visão geral do funcionamento do Docker	25
Figura 3 - Gráfico dos meios de comunicação	26
Figura 4 - Gráfico da opinião dos entrevistados sobre a solução apresentada	27
Figura 5 - Caso de Uso do ADMIN.....	31
Figura 6 - Caso de Uso do USER.....	32
Figura 7 - Caso de Uso do Sistema	32
Figura 8 - Diagrama de Sequência / Viagem.....	33
Figura 9 - Diagrama de Atividade / Viagem	34
Figura 10 - Diagrama de Entidade-Relacionamento (DER).....	35
Figura 11 - Diagrama de Classes.....	35
Figura 12 - Diagrama de Arquitetura	36
Figura 13 - Ferramentas Utilizadas	39
Figura 14 - Swagger UI Estrutura Geral.....	40
Figura 15 - Swagger UI endpoints <i>Destino</i>	41
Figura 16 - Estrutura das pastas no Backend.....	42
Figura 17 - Estrutura das pastas no Frontend	44
Figura 18 - Tela de Acesso ao Sistema.....	45
Figura 19 - Tela Inicial do Sistema (Home).....	46
Figura 20 - Tela Materiais	46
Figura 21 - Cadastro de Material.....	47
Figura 22 - Cadastro de Usuário.....	48
Figura 23 – Tela de Motoristas.....	49
Figura 24 - Edição de Motorista.....	49
Figura 25 - Confirmação de exclusão de Motorista	50
Figura 26 - Tela de Sedes	50
Figura 27 - Edição de Sede.....	51
Figura 28 - Confirmação de exclusão de Sede	51
Figura 29 – Inscrever-se / Desinscrever-se na lista de Interessados da Sede.....	52
Figura 30 - Tela de Veículos	52
Figura 31 - Edição de Veículo.....	53

Figura 32 - Confirmação de exclusão de Veículo	53
Figura 33 - Cadastramento de Veículo	54
Figura 34 - Tela de Viagens	54
Figura 35 - Viagem Confirmada.....	55
Figura 36 - Edição da Viagem.....	55
Figura 37 – Relatório da Viagem (PDF)	56
Figura 38 - Cadastro de Viagem.....	57
Figura 39 - Notificação por E-mail.....	57
Figura 40 - Cobertura de Testes Unitários	58
Figura 41 - Gráfico da notificação aos setores	63

LISTA DE TABELAS

Tabela 1 - Requisitos funcionais do sistema STM	28
Tabela 2 - Requisitos não funcionais do sistema STM	30

LISTA DE ABREVIATURAS E SIGLAS

ACID - Atomic, Consistent, Isolated and Durable;
ADMIN – Administrador;
API - Application Programming Interface;
CEP - Código de Endereçamento Postal;
CSS - Cascading Style Sheets;
DDD - Domain Driven Design;
DER - Diagrama Entidade-Relacionamento;
EE - Enterprise Edition;
ES6 - ECMAScript 6;
GPS - Global Position System;
HTML - HyperText Markup Language;
ID - Identifier ou Identificador;
IDE - Integrated Development Environment;
JPA - Jakarta Persistence API;
JS – JavaScript;
JSF - JavaServer Faces;
JSON - JavaScript Object Notation;
JVM - Java Virtual Machine;
JWT - JSON Web Token;
MVC - Model, View Controller;
OAS - OpenAPI Specification;
ORM - Object-Relational Mapping;
PDF - Portable Document Format;
PGE SC - Procuradoria Geral do Estado de Santa Catarina;
RENAVAM - Registro Nacional de Veículos Automotores;
REST - Representational State Transfer;
SAAS - Software as a Service ;
SE - Standard Edition;
SMTP - Simple Mail Transfer Protocol;
SPA - Single Page Application;

SQL - Structured Query Language;

STM - Sistema de Transferência de Material;

TMS - Transportation Management System;

TS – TypeScript;

UF - Unidade Federativa;

UI - User Interface;

UML - Unified Modeling Language;

USER - User ou Usuário;

YAML - Yet Another Markup Language;

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	14
1.2	OBJETIVO GERAL	14
1.3	OBJETIVOS ESPECÍFICOS	14
1.4	JUSTIFICATIVA	14
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	LOGÍSTICA	17
2.2	TRANSPORTE	17
2.3	LEI DO TRANSPORTE DE CARGA PARA O MOTORISTA	19
2.4	TECNOLOGIAS	19
2.4.1	CSS	19
2.4.2	HTML	20
2.4.3	Javascript.....	20
2.4.4	Typescript.....	20
2.4.5	Angular	21
2.4.6	Java	21
2.4.7	Maven.....	22
2.4.8	JWT.....	23
2.4.9	Spring Framework.....	23
2.4.10	Open Api.....	23
2.4.11	Banco de dados.....	24
2.4.11.1	PostgreSQL.....	25
2.4.11.2	Docker.....	25
3	METODOLOGIA.....	26

3.1	Pesquisa	26
3.2	Requisitos	27
3.3	Diagramas	31
4	DESENVOLVIMENTO.....	37
4.1	Descrição do Projeto.....	37
4.2	DECISÕES do Projeto.....	38
4.3	CONSTRUÇÃO do Projeto.....	40
4.4	RESULTADOS E TESTES.....	57
5	CONCLUSÃO.....	59
6	TRABALHOS FUTUROS	60
	REFERÊNCIAS.....	61
	APÊNDICE A – Resultados das pesquisas.....	63

1 INTRODUÇÃO

A logística tem um papel importante nas empresas, independentemente de seu porte. Segundo Paura (2016), a logística não tem uma data específica de surgimento, mas seu conceito foi utilizado em diversos marcos importantes ao longo da história, como nas tropas de Alexandre, o Grande, a construção das pirâmides do antigo Egito e a derrota do exército de Napoleão. É possível identificar alguns acontecimentos em que se utilizaram estratégias de organização e logística na prática.

O gerenciamento ou "processamento" de pedidos é essencial na logística, já que se relaciona com o ciclo do pedido, que vai desde o momento em que é solicitado até o momento em que é entregue ao cliente (CARVALHO, 2009). A partir disso, demonstra-se que uma boa logística é fundamental para diversas empresas.

Em diversas empresas, tanto públicas quanto privadas, que realizam transferência de materiais internamente em seu cotidiano, realizadas entre suas sedes e unidades que se encontram espalhadas pela cidade, estado ou até mesmo pelo país, podem enfrentar dificuldades quando não possuem uma logística adequada em suas entregas, ocasionando atrasos no envio, no transporte ou na distribuição dos materiais solicitados em questão. Assim como afirmado por Carvalho (2009), se o processamento for complexo ou ineficiente, ele pode retardar o tempo de ciclo do pedido e ainda aumentar os custos de transporte para a empresa.

Além disso, é essencial que a empresa conte com profissionais capacitados e ferramentas adequadas, como sistemas que auxiliam no gerenciamento da logística de forma eficiente, para minimizar atrasos no tempo de entrega do pedido.

A seleção do hardware e dos sistemas para o processamento de pedidos representa mais uma etapa dos elementos do projeto. São inúmeros os fatores com peso suficiente para acelerar ou retardar o tempo de processamento. Resultam eles de procedimentos operacionais, políticas de serviço ao cliente e práticas de transporte. (BALLOU, 2006, p.123).

Com base nesse contexto, este trabalho tem como objetivo apresentar o desenvolvimento do Sistema de Transferência de Material (STM), que visa gerenciar a transferência de materiais entre diferentes sedes, facilitando as entregas programadas e notificando os interessados sobre os destinos aceitos por meio de notificações por e-mail. Dessa forma, o sistema contribui para uma logística adequada, permitindo que a empresa apresente maior eficiência na entrega aos seus funcionários, com uma comunicação adequada e menor tempo e custo.

1.1 OBJETIVOS

Nas seções abaixo estão descritos o objetivo geral e os objetivos específicos deste TCS.

1.2 OBJETIVO GERAL

Desenvolver um sistema web para gerenciar e comunicar a transferência de materiais entre sedes de uma mesma empresa, com o objetivo de auxiliar na gestão das viagens.

1.3 OBJETIVOS ESPECÍFICOS

- Modelar um sistema web de apoio logístico.
- Desenvolver uma aplicação baseada nos diagramas e modelos criados.
- Realizar levantamento de requisitos para empresas que se encaixam no escopo estabelecido.
- Criar diagramas UML que representem o sistema, seus componentes e relações.
- Criar protótipos de telas.
- Seguir a arquitetura de projetos Maven.
- Realizar testes unitários no sistema desenvolvido.

1.4 JUSTIFICATIVA

A entrega de produtos ou materiais não é tão simples como parece, envolve mais do que apenas entregar a quantidade correta no lugar e hora determinados para o cliente final. Trata-se de ter o controle das entregas e a comunicação adequada. Empresas com sedes espalhadas por pequenas ou grandes áreas geográficas enfrentam dificuldades em suas transferências internas de materiais, quando não possuem uma boa logística e comunicação entre os responsáveis pela entrega e os solicitantes do material (CARVALHO, 2009).

No entanto, apenas a logística por si só não é suficiente para uma melhoria significativa na gestão das solicitações e entregas. É necessário uma comunicação básica e parceria dos vários setores que constituem uma empresa (PAURA, 2016). O trabalho apresentado tem o propósito de desenvolver um sistema para gerenciar a transferência de materiais entre as sedes de uma mesma empresa e a comunicação entre o setor responsável pela distribuição e os diversos setores solicitantes do material.

A implementação do STM (Sistema de Transferência de Material) permitirá o agendamento de transferência de diversos materiais para destinos específicos (sedes) e

solicitantes (setores), por meio do responsável pelo gerenciamento e apresentando informações necessárias para a realização da transferência. O STM servirá de auxílio para entidades públicas ou privadas que enfrentam dificuldades no mesmo quesito de transferência entre suas próprias sedes.

Com o objetivo de trazer melhores benefícios ao processo, o responsável pelas transferências terá o controle de cadastros, atualizações e remoção de viagens, destinos, materiais, motoristas, veículos e sedes no sistema para facilitar o processo de agendamento de viagens. O motorista poderá ter acesso apenas ao arquivo em PDF (Portable Document Format) ou em folhas impressas disponibilizadas pelo responsável pelas transferências e os demais usuários, que são os solicitantes dos materiais, como financeiro, recurso humanos, marketing e outros, terão a possibilidade de solicitar um novo destino e material na viagem.

A Figura 1 apresenta uma comparação entre três sistemas que possuem funcionalidades semelhantes ao nosso sistema STM em desenvolvimento. São eles: Transportation Management System - TMS e Active Corp, que apresentam recursos como consultas, atualizações e cadastro de veículos, motoristas, destinos, materiais e outros, tendo algumas diferenças, como o checklist. Já o sistema Aethos, possui um serviço específico para "transporte" que gera um relatório de informações da viagem, porém não disponibiliza um PDF para consulta e também não apresenta a gestão do motorista em seu sistema, o que limita a especificação de algumas informações relevantes, como por exemplos, quem é o motorista, que está realizando a viagem.

Figura 1 - Tabela Comparativa

TABELA COMPARATIVA				
FUNCIONALIDADES	TMS	ACTIVE	AETHOS	STM
GESTÃO DE MOTORISTA	✓	✓	✗	✓
GESTÃO DE FROTA	✓	✓	✓	✓
DEMANDA DE CARGA	✓	✓	✓	✓
PAINEL DE CARGAS	✓	✓	✓	✓
RELATÓRIO DO ITINERÁRIO	✗	✗	✓	✓
COMPROVANTE DE VIAGEM	✓	✓	✓	✓

Fonte: elaborado pelos autores (2023).

O nosso sistema define dois tipos de operadores: o administrador e o usuário, delimitando as suas autorizações. O administrador tem a responsabilidade de confirmar o agendamento da viagem, incluindo os destinos e materiais envolvidos. O motorista, que não tem acesso ao sistema, recebe um arquivo em PDF contendo todas as informações necessárias para a realização da transferência de materiais. Essa gestão dos motoristas é realizada pelo administrador, responsável pelas transferências dos materiais.

2 FUNDAMENTAÇÃO TEÓRICA

O objetivo deste capítulo é introduzir os conceitos de logística e transporte, fornecendo uma base sólida para compreender o problema abordado e as tecnologias empregadas no desenvolvimento do sistema.

2.1 LOGÍSTICA

A logística é uma área de grande abrangência, que se faz presente em diversos aspectos do nosso cotidiano, desde a organização pessoal até a entrega de produtos aos clientes nas empresas. De acordo com Paura (2016), a logística empresarial tem suas raízes nos conceitos desenvolvidos em guerras e conflitos históricos, que inspiraram a aplicação desse gerenciamento nas empresas, com o propósito de sincronizar todas as atividades internas e externas.

Ao implementar a logística, o valor é integrado ao produto e aos serviços demandados pelo cliente, com o intuito de atender às suas necessidades e impulsionar as vendas da empresa (BALLOU, 2006). A ausência de uma estratégia logística é prejudicial para as empresas, pois impede a movimentação eficiente de materiais, o desempenho de operações eficazes, a entrega pontual de produtos e um atendimento adequado ao cliente (WATERS, 2021).

Portanto, a logística aplicada nas empresas e sua ausência representam duas realidades opostas. Uma logística empresarial bem executada envolve a comunicação entre todas as partes envolvidas, incluindo a própria empresa, seus setores e os consumidores, além de uma gestão adequada nos processos de pedidos, armazenamento, seleção de produtos, transporte e entrega, visando ao benefício geral, à satisfação do cliente e à redução de custos mediante a conclusão eficiente dos processos. Por outro lado, as empresas que não adotam o conceito de logística acabam estagnadas, demorando mais para concluir a entrega de seus produtos ou serviços, causando insatisfação ao cliente pela falta de comunicação e gestão interna, e deixando de se beneficiar com a aplicação de técnicas logísticas (BALLOU, 2006; WATERS, 2021).

2.2 TRANSPORTE

Um dos conceitos fundamentais da logística é o transporte, que tem como objetivo principal a movimentação de materiais, mercadorias, produtos ou matéria-prima do ponto de origem ao destino final, abrangendo transferências entre diferentes pontos (CABRAL FILHO, 2023).

Resumidamente, o transporte implica na transferência de mercadorias, tanto de uma região para outra, ou para o cliente, quanto internamente, quando há necessidade de movimentar materiais entre estoques ou unidades da mesma empresa. Essa movimentação interna também é considerada transporte (PAURA, 2016). O transporte pode ser otimizado por meio de uma boa comunicação interna entre os setores que solicitam materiais e os responsáveis pelas entregas, permitindo o agrupamento de diversos materiais em um mesmo transporte, evitando custos e perdas de tempo desnecessários para a empresa.

No contexto do transporte, existem diferentes tipos, conhecidos como modais de transporte. Waters (2021) descreve cinco modais principais: rodoviário, hidroviário, ferroviário, dutoviário e aéreo. Cada modal possui características distintas que o diferenciam dos demais, e a escolha do modal adequado depende de fatores como o tipo de material a ser transportado, o local de entrega, a distância entre o ponto de origem e o destino final, entre outros procedimentos a serem considerados para determinar a melhor opção de transporte para uma determinada atividade.

No caso específico das entregas dentro de uma empresa, o modal rodoviário é frequentemente utilizado, especialmente quando se trata de transferências entre unidades da mesma cidade ou estado. Esse modal envolve o uso de carros ou caminhões para transportar documentos, patrimônios de diferentes tamanhos, conforme a quantidade solicitada e a disponibilidade de transporte (PAURA, 2016).

No Brasil, o principal modal utilizado é o rodoviário que se utiliza de carros, carretas e caminhões para o transporte. De todos os modais é o que apresenta maior flexibilidade, uma vez que com ele chegamos a vários pontos da cidade (PAURA, 2016, p.37).

Considerando o modal que melhor se adequa ao tema em questão, que é o transporte de materiais entre sedes, é possível obter flexibilidade suficiente para aproveitar uma única viagem e efetuar a entrega de várias solicitações, tanto para o mesmo destino quanto para outros destinos ao longo do percurso até o ponto final da viagem (PAURA, 2016).

Ao considerar o conceito de transporte e a escolha do modal rodoviário para viagens entre localidades, é relevante avaliar se o transporte será de propriedade da empresa ou terceirizado (PAURA, 2016).

Conforme mencionado por Cabral Filho (2023), existem vantagens e desvantagens em relação ao transporte próprio e terceirizado para as empresas. A decisão geralmente se

baseia na necessidade de a empresa garantir o controle sobre o transporte e os materiais, permitindo gerenciá-los de acordo com suas preferências, especialmente quando não encontram uma empresa terceirizada confiável. Um exemplo disso é a Procuradoria Geral do Estado de Santa Catarina - PGE SC, que possui seu próprio transporte para realizar transferências entre suas sedes em todo o estado.

2.3 LEI DO TRANSPORTE DE CARGA PARA O MOTORISTA

A lei que regula a profissão de motorista, conforme estabelecido na Lei nº 13.103, determina que a jornada de trabalho para motoristas de carga não deve exceder 8 horas diárias, podendo ser estendida por mais 2 horas extras, desde que não ultrapasse 44 horas semanais. Além disso, é garantido ao motorista um intervalo mínimo de 1 hora. Segundo essa lei, que foi mencionada anteriormente em 2 de março, é permitida a realização de inspeções para veículos com excesso de peso, desde que haja autorização de um órgão específico. Essa inspeção técnica deve ser realizada a cada 12 meses. No caso de transporte de cargas perigosas, o motorista deve possuir uma habilitação conforme as normas de segurança legislativa (BRASIL, 2015).

2.4 TECNOLOGIAS

Nesta seção serão apresentadas as tecnologias utilizadas no desenvolvimento da aplicação do STM.

2.4.1 CSS

Cascading Style Sheet (CSS) é uma linguagem de estilo utilizada para controlar a aparência e o layout de documentos HyperText Markup Language (HTM). Permite definir cores, fontes, tamanhos, espaçamentos e outros estilos para elementos específicos. O CSS seleciona elementos HTML e aplica regras de estilo, escritas em um arquivo separado e referenciado no HTML por meio da tag <link>. Utilizando um método de cascata, determina a prioridade das regras e define o estilo final dos elementos, possibilitando designs consistentes e personalizados para páginas da web (Mozilla Foundation, 2023).

2.4.2 HTML

HTML é a linguagem de marcação utilizada para criar páginas da web. Define a estrutura e o conteúdo de uma página, organizando texto, imagens, links e outros elementos visuais por meio de elementos e tags. Cada tag tem um propósito específico, como <h1> para títulos de nível 1, <p> para parágrafos e para imagens. O navegador interpreta o HTML e exibe o conteúdo de forma visualmente agradável e navegável, seguindo as instruções fornecidas na marcação (Mozilla Foundation, 2023).

2.4.3 Javascript

JavaScript ou apenas JS é uma linguagem de programação orientada a objetos que possui ampla utilização no desenvolvimento de páginas da web e também em outros ambientes, como o node.js e o Adobe Acrobat. Além disso, o JS oferece recursos como protótipos, suporte a múltiplos paradigmas e tipagem dinâmica, permitindo a adoção de estilos de programação orientados a objetos, imperativos e declarativos, seguindo o padrão ECMAScript. A partir de 2012, a maioria dos navegadores modernos possui suporte completo ao ECMAScript 5.1, enquanto versões mais antigas suportam pelo menos o ECMAScript 3. Em junho de 2015, foi publicada a sexta versão do ECMAScript, oficialmente chamada de ECMAScript 2015 ou ES6. Desde então, novas versões do ECMAScript são lançadas anualmente, sendo a mais recente a ECMAScript 2018 (Mozilla Foundation, 2023).

2.4.4 Typescript

O Typescript ou TS é uma extensão do JavaScript que visa facilitar o desenvolvimento de aplicativos em Javascript em grande escala. Embora todo programa em Javascript seja também um programa em TypeScript, este último oferece um sistema de módulos, classes, interfaces e um sistema de tipos gradual avançado. A intenção é que o TypeScript forneça uma transição suave para os programadores que já estão familiarizados com o Javascript, pois os idiomas bem estabelecidos na programação em Javascript são suportados sem a necessidade de grandes reescritas ou anotações. Uma consequência interessante é que o sistema de tipos do TypeScript não é estaticamente seguro por design (Microsoft, 2023).

2.4.5 Angular

O Angular é um framework de código aberto para front-end, desenvolvido pelos engenheiros do Google em 2012. O angular é usado para construção de sistemas web baseado em uma única página dinâmica ou SPA – Single Page Application, que também possibilita a criação de menus animados usando HTML (ANGULAR, 2022).

Em operação, o Angular cria a aplicação de maneira parecida ao MVC - Model View Controller. Os dados são apresentados na sua grande parte são através de estruturas JSON (JavaScript Object Notation), denominadas como modelos (JUNQUEIRA, 2018).

O angular inclui uma vasta coleção bibliotecas com diversos recursos, uma estrutura baseada em componentes para a criação de aplicativos web escaláveis e uma vasta gama de ferramentas para auxiliar ao desenvolvedor na parte de criação, testes e atualizações do código (ANGULAR, 2022).

2.4.6 Java

O ecossistema Java é um conjunto de tecnologias, ferramentas e frameworks relacionados ao desenvolvimento de software em Java. Ele oferece um ambiente robusto e diversificado para a criação de aplicativos, abrangendo desde a escrita do código-fonte até a implantação e execução em diferentes plataformas (ORACLE, 2023).

De acordo com a Oracle, empresa responsável pelo Java, o ecossistema Java é composto por uma ampla gama de recursos, incluindo a linguagem de programação Java em si, a Máquina Virtual Java (JVM), a plataforma Java SE (Standard Edition), a plataforma Java EE (Enterprise Edition) e uma infinidade de bibliotecas, frameworks e ferramentas de desenvolvimento (ORACLE, 2023).

Algumas das principais vantagens do ecossistema Java incluem a portabilidade pois o Java é projetado para ser executado em diferentes plataformas incluindo Windows, macOS, Linux e dispositivos móveis, o que possibilita a criação de aplicativos que podem ser executados em diversos ambientes. O Java e suas ferramentas permitem o desenvolvimento de aplicativos escaláveis, capazes de lidar com grandes volumes de dados e tráfego, apresentando também uma vasta comunidade ativa e engajada, que contribui para o desenvolvimento de bibliotecas, frameworks e soluções, além de oferecer suporte e compartilhar conhecimento por meio de fóruns, grupos de discussão e eventos. A vantagem mais importante é a sua abundância

de bibliotecas e frameworks disponíveis, como Spring, Hibernate, JavaFX, JUnit, entre outros, que facilitam o desenvolvimento de aplicações robustas, seguras e eficientes (ORACLE, 2023).

É crucial destacar que o ecossistema Java está em constante evolução, com atualizações regulares da linguagem e da plataforma, além do surgimento de novas tecnologias e práticas de desenvolvimento. Essas características contribuem para a longevidade e relevância do ecossistema Java no cenário do desenvolvimento de software.

2.4.7 Maven

Maven é um conjunto de ferramentas e tecnologias desenvolvido para facilitar o gerenciamento de projetos de software em Java. Ele é amplamente utilizado na comunidade de desenvolvimento Java devido à sua eficiência, simplicidade e reutilização de componentes (APACHE, 2023).

Conforme definido pela Apache Software Foundation, o Maven é um sistema de automação de compilação e gerenciamento de dependências. Ele permite que os desenvolvedores definam a estrutura do projeto, gerenciem as dependências de bibliotecas externas e automatizem tarefas como compilação, teste e empacotamento (APACHE, 2023).

O Maven possui diversas vantagens em seu ecossistema, como o gerenciamento de dependências, de um projeto, permitindo que as bibliotecas externas sejam especificadas em um arquivo de configuração centralizado (arquivo pom.xml), assim ele baixa automaticamente as dependências necessárias e as torna disponíveis para o projeto, facilitando a integração de bibliotecas e evitando conflitos de versão, deixando padronizado, pois segue uma estrutura e conversão bem definida, o que ajuda na manutenção e organização dos projetos. Ele define uma estrutura de diretórios padrão e um ciclo de vida de construção, o que permite que os desenvolvedores se concentrem no desenvolvimento do código, sem se preocupar com a configuração do ambiente (APACHE, 2023).

Além disso o Maven possui um repositório central que contém um vasto número de bibliotecas e artefatos disponíveis para uso. Isso permite que os desenvolvedores acessem facilmente bibliotecas populares e confiáveis, economizando tempo e esforço na busca e gerenciamento de dependências externas. Inclui auxílio na construção e automação de tarefas do projeto, como compilação, teste, empacotamento, geração de relatórios e implantação, sendo possível definir configurações personalizadas e executar essas tarefas de forma rápida e consistente (APACHE, 2023).

Várias IDEs (Integrated Development Environments) apresentam suporte ao Maven, como Eclipse e IntelliJ, facilitando a integração e o desenvolvimento colaborativo em equipe. Além disso pode ser facilmente integrado a outras ferramentas de desenvolvimento, como servidores de integração contínua e ferramentas de análise de código (APACHE, 2023).

2.4.8 JWT

Tokens JWT (JSON Web Token) são uma forma de autenticação e autorização amplamente utilizada em aplicações web. Eles consistem em três partes: um cabeçalho, uma carga útil (payload) e uma assinatura digital. (PUTRA et al., 2018).

O cabeçalho contém informações sobre o algoritmo de criptografia utilizado. A carga útil contém os dados do usuário ou outras informações relevantes. A assinatura digital verifica a integridade do token. Quando um usuário faz login, um token JWT é gerado e enviado ao cliente. O cliente envia o token em cada requisição posterior. O servidor verifica a assinatura para garantir que o token não tenha sido modificado e, em seguida, usa as informações da carga útil para autenticar e autorizar o usuário permitindo assim a escalabilidade em sistemas distribuídos (PUTRA et al., 2018).

2.4.9 Spring Framework

O Spring Framework é um dos frameworks mais populares para desenvolvimento de aplicações Java. Criado em 2003 por Rod Johnson, ele foi projetado para tornar o desenvolvimento de aplicativos Java mais fácil e rápido. O Spring é uma estrutura modular e abrangente que oferece suporte a vários recursos, como injeção de dependência, gerenciamento de transações, segurança, controle de versão e muito mais. Além disso, o Spring é altamente configurável e pode ser integrado com outros frameworks e tecnologias, como Hibernate, Struts, JSF, entre outros. Com sua ampla adoção e constante evolução, o Spring Framework continua a ser uma escolha popular para desenvolvedores Java em todo o mundo. (VMware, 2023).

2.4.10 Open Api

A OpenAPI Specification (OAS), anteriormente conhecida como Swagger, é a especificação principal do ecossistema OpenAPI. Ela define um formato padrão para descrever APIs RESTful, permitindo que desenvolvedores e equipes de projeto documentem e compartilhem informações sobre suas APIs. A OAS é escrita em formato YAML ou JSON e fornece uma descrição detalhada dos recursos, endpoints, parâmetros, respostas e outras informações relevantes sobre a API (OpenApi, 2022).

A geração automática de documentação interativa para APIs, é uma das funcionalidades presentes na OAS, para facilitar a compreensão e o uso da mesma por parte dos desenvolvedores. Essa documentação inclui exemplos de solicitações e respostas, descrições de parâmetros e códigos de status, o que torna mais fácil para os consumidores entenderem como interagir com a API. A utilização da especificação padronizada facilita a comunicação transparente entre os sistemas, permitindo que diferentes implementações e ferramentas sejam usadas para criar, testar e consumir APIs compatíveis com a OAS (OpenApi, 2022).

Com base na especificação OpenAPI, é possível realizar validação e testes automatizados para garantir que a API esteja de acordo com as especificações definidas. Isso ajuda a evitar erros e inconsistências durante o desenvolvimento e garante a conformidade com as melhores práticas. O seu ecossistema é suportado por diversas linguagens de programação e frameworks, o que facilita a integração e o desenvolvimento de APIs em diferentes ambientes (OpenApi, 2022).

2.4.11 Banco de dados

Um banco de dados relacional é uma forma de organizar e armazenar dados em tabelas estruturadas, onde os dados são relacionados por chaves e relacionamentos. Cada tabela representa uma entidade e cada linha representa um registro específico. As tabelas podem ser conectadas por meio de chaves estrangeiras, permitindo a criação de relacionamentos entre os dados (ORACLE, 2023).

O acesso aos dados em um banco de dados relacional é feito por meio da linguagem SQL (Structured Query Language), permitindo a consulta, inserção, atualização e exclusão dos registros. Os bancos de dados relacionais são amplamente utilizados devido à sua capacidade de garantir a consistência e integridade dos dados. (ORACLE, 2023)

2.4.11.1 PostgreSQL

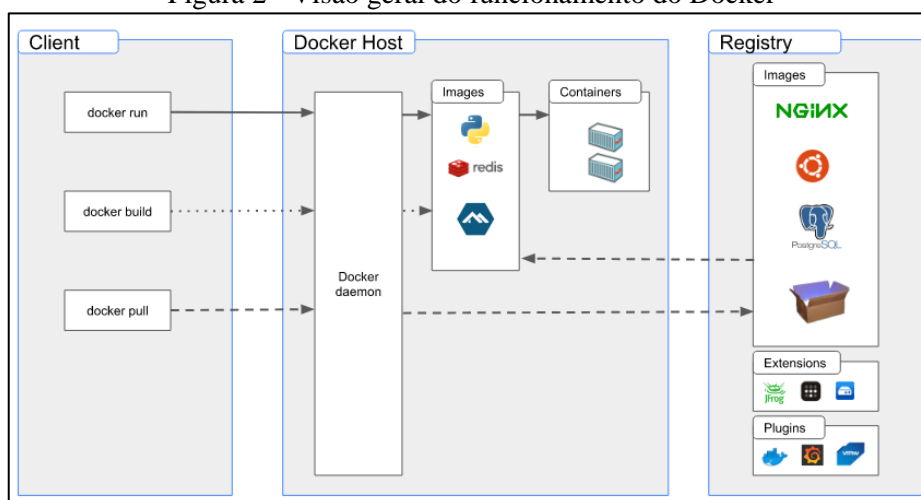
O PostgreSQL é um banco de dados relacional que utiliza o SQL, é um banco de dados relacional de código aberto e gratuito. Ele armazena dados de forma organizada em tabelas com linhas e colunas. Os dados são persistidos em disco para garantir sua durabilidade e oferece uma consulta para manipular os dados (MILANI, 2008).

Tem suporte a transações ACID (Atômicas, Consistentes, Isoladas e Duráveis), integridade referencial, chaves estrangeiras e suporte a funções e procedimentos armazenados e altamente escalável pois possui recursos avançados de segurança, como autenticação, controle de acesso e criptografia (MILANI, 2008).

2.4.11.2 Docker

A Figura 2 demonstra a visão geral do funcionamento do Docker. O Docker é uma plataforma de código aberto que permite empacotar, distribuir e executar aplicativos de forma isolada em contêineres. Um contêiner Docker inclui todo o ambiente necessário para a execução de um aplicativo, incluindo o código, as dependências e as configurações. Essa abordagem permite que os aplicativos sejam executados de maneira consistente em qualquer ambiente, independentemente das diferenças entre os sistemas operacionais e configurações de hardware. O Docker facilita o desenvolvimento, o teste e a implantação de aplicativos, fornecendo uma maneira padronizada e eficiente de gerenciar os recursos e garantir a portabilidade dos aplicativos em diferentes ambientes de implantação. (DOCKER, 2023).

Figura 2 - Visão geral do funcionamento do Docker



Fonte: DOCKER (2023).

3 METODOLOGIA

Nesta seção serão abordados os seguintes tópicos: pesquisa, requisitos e os diagramas referentes ao sistema STM.

3.1 PESQUISA

O projeto abordado neste documento tem uma natureza aplicada, pois visa criar um sistema que solucione um problema específico, ao mesmo tempo em que apresenta uma perspectiva exploratória, buscando compreender melhor as necessidades do público-alvo e as possíveis soluções para o problema.

A pesquisa realizada adotou uma abordagem quali-quantitativa, combinando elementos qualitativos para compreender e explicar as causas prováveis do problema enfrentado pelas empresas com o mesmo cenário, e elementos quantitativos para realizar comparações necessárias para validar o sistema.

Foram aplicados dois questionários por meio do Google Forms, direcionados às empresas que realizam transferências de materiais entre suas sedes, conforme descrito anteriormente. No primeiro questionário, foi apresentado o conceito e o propósito do nosso sistema para melhor compreensão dos participantes. Participaram 16 pessoas, que foram identificadas como público-alvo do sistema. Os dados quantitativos coletados revelaram que 100% das empresas realizam transferências de materiais, 75% enfrentam viagens desnecessárias e como na Figura 3 (68,8%) preferem comunicação por e-mail para informar os usuários sobre as viagens planejadas.

Figura 3 - Gráfico dos meios de comunicação

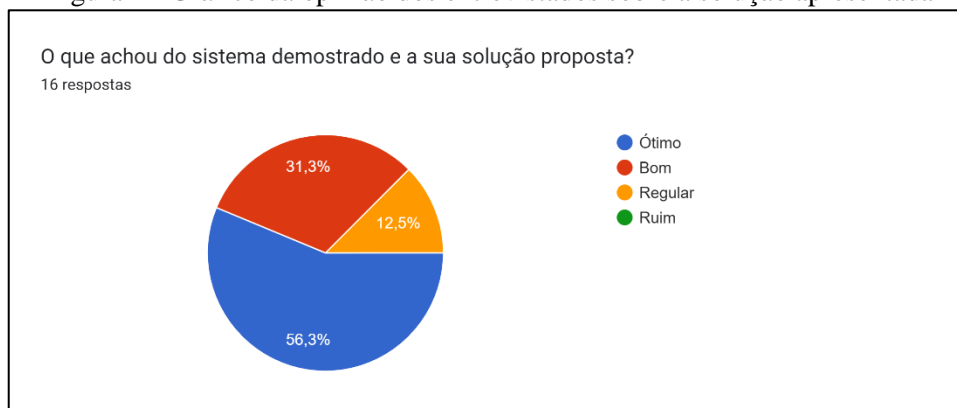


Fonte: elaboração dos autores (2023).

Referente ao primeiro questionário, as sugestões de funcionalidades levantadas pelo público-alvo foram discutidas pela equipe de desenvolvimento em reuniões presenciais e online. Durante essas discussões, avaliamos a importância da implementação dessas funcionalidades no sistema, considerando a relevância de cada sugestão e as necessidades dos possíveis usuários. Algumas dessas funcionalidades foram implementadas, juntamente com outras sugeridas ao longo do processo de desenvolvimento do sistema.

O segundo questionário teve como objetivo validar a solução, apresentando o sistema em desenvolvimento aos colaboradores. Responderam a esse questionário 16 pessoas, e suas respostas forneceram insights sobre a percepção do sistema. A Figura 4 apresenta uma das perguntas da validação da solução. As demais informações e perguntas relacionadas à pesquisa podem ser encontradas no "APÊNDICE A – Resultados das pesquisas".

Figura 4 - Gráfico da opinião dos entrevistados sobre a solução apresentada



Fonte: elaboração dos autores (2023)

As demais informações e perguntas referentes a pesquisa estão no final deste documento no “APÊNDICE A – Resultados das pesquisas”.

3.2 REQUISITOS

A realização do levantamento de requisitos se dá ao verificar as descrições, serviços e restrições que devem ser incluídas no sistema ou produto. A engenharia de requisitos é o processo de levantar, analisar, documentar e verificar todo o sistema (SUMMERVILLE, 2011).

Os requisitos funcionais são os serviços prestados pelo sistema, os comportamentos que ele deve apresentar ao usuário, dependendo de suas ações dentro do sistema, e também

explicam o que pode ou não ser realizado, dependendo do propósito da aplicação (SUMMERVILLE, 2011).

A representação das funcionalidades do STM está descrita na Tabela 1, que descreve as permissões e restrições do sistema e seus usuários.

Tabela 1 - Requisitos funcionais do sistema STM

Identificação	Requisitos funcionais
RF-01	O sistema deve permitir que o usuário efetue login, através dos campos de usuário (nome do usuário cadastrado no sistema) e senha (padrão alfanumérico).
RF-02	O sistema deve verificar e autorizar o usuário logado no sistema com os papéis “ADMIN” ou “USER”.
RF-03	Usuários identificados como “ADMIN”, podem manter (cadastrar, editar, consultar e excluir) viagem, destino, material, sede, motorista e veículo.
RF-03.01	O usuário autorizado “ADMIN” pode editar, excluir e consultar um motorista pelo seu ID, somente após inserir o motorista, preenchendo seus campos obrigatórios como (Nome, categoria da carteira e e-mail).
RF-03.02	O usuário “ADMIN” pode editar, excluir e consultar uma sede pelo seu ID, somente após inserir a sede, preenchendo todos os campos obrigatórios como (Nome, rua, número, CEP, cidade, UF) e ainda é possível deixar uma observação da sede caso necessário.
RF-03.03	O usuário “ADMIN” pode editar, excluir e consultar um veículo pelo seu ID, somente após inserir o veículo, preenchendo seus campos obrigatórios como (Marca, modelo, placa, ano, RENAVAM e tamanho).
RF-03.04	O “ADMIN” pode editar, excluir e consultar uma viagem pelo seu ID, somente após realizar a criação da viagem, preenchendo todos os campos obrigatórios como (Sede de origem, data de saída, destino(s), veículo, motorista).
RF-04	Usuário “ADMIN” é autorizado realizar algumas atividades extras em relação a viagem, destino e sede.
RF-04.01	Ao usuário “ADMIN” é permitido CONFIRMAR ou CANCELAR uma Viagem desde que ela ainda não tenha sido realizada ou encerrada (confirmar uma viagem significa que ela ocorrerá, cancelar significa que ela não mais acontecerá).

RF-04.02	Ao usuário “ADMIN” é permitido CONFIRMAR ou CANCELAR um Destino (confirmar um destino significa que ele ocorrerá, cancelar significa que ela não será um ponto de parada na viagem vigente).
RF-05	Após a confirmação da viagem o sistema deve notificar o motorista, via e-mail com um PDF anexado, apresentando os dados da viagem.
RF-06	O sistema deve gerar um PDF com os seguintes dados da viagem, data de saída, nome do motorista, placa do veículo, destino, setor e material para o usuário “ADMIN”.
RF-07	Os usuários identificados como “USER”, podem criar, editar e consultar o destino e os seus materiais, realizar a consulta no sistema e ainda inserirem-se ou retirarem-se da lista de interessados das sedes.
RF-07.01	O “USER” pode editar e consultar um destino com seus materiais na viagem, somente após inserir o destino e material, preenchendo todos os campos obrigatórios como (setor e destinatário) sobre o destino e (material e quantidade) referentes aos materiais que serão enviados, porém sendo necessário a confirmação por parte de usuário autorizado.
RF-07.02	O usuário com o papel de “USER” pode realizar as consultas de viagem, sede, motorista e veículo, pelos seus respectivos IDs.
RF-08	Quando um Destino (que contém uma sede como parada) for adicionado à uma viagem os interessados que estiverem inscritos na lista de notificações dessa sede específica receberão um e-mail de notificação.
RF-09	O sistema deve apresentar uma lista das viagens, motoristas, veículos, e sedes ativas no sistema.
RF-10	O sistema deve possibilitar a identificação das viagens como abertas ou fechadas.

Fonte: elaboração dos autores (2023)

Os requisitos não funcionais, são representadas pelas restrições que o sistema deve tomar e funcionalidades que o sistema contém, muitas vezes não estão ligadas pelos serviços que o sistema oferece para os seus usuários, mas sim as características internas que o sistema deve apresentar (SUMMERVILLE, 2011).

A representação deste requisitos não funcionais dentro do sistema STM está presente na Tabela 2.

Tabela 2 - Requisitos não funcionais do sistema STM

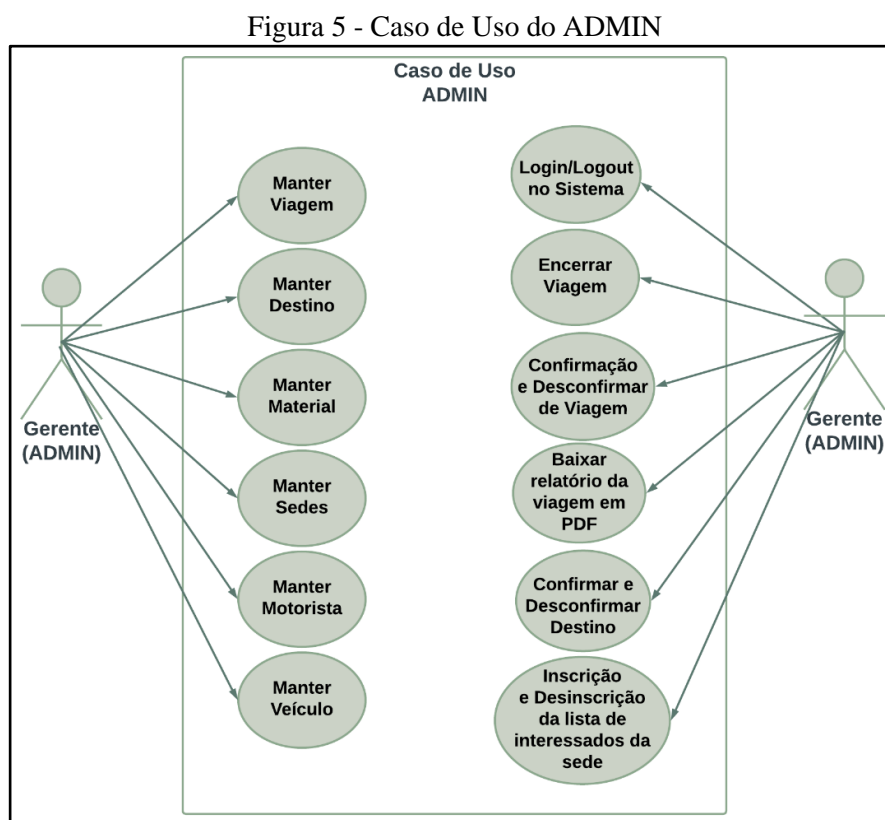
Identificação	Requisitos não funcionais
RNF-01	O sistema disponibilizará uma interface gráfica utilizando os padrões comumente praticados no Material Design definido pelo Framework Angular. (https://material.angular.io/)
RNF-02	O sistema permitirá o logout do usuário através do botão “Logout”.
RNF-03	Os dados persistidos pelo usuário devem apresentar integridade, busca quase instantânea e a consistência ao longo do tempo.
RNF-04	O sistema deverá funcionar pelo menos nos navegadores Google Chrome e FireFox.
RNF-05	As telas do sistema são projetadas para monitores de computador com resolução mínima de 1366x768p.
RNF-06	O sistema deve reconhecer todos os usuários utilizando o sistema de papéis (Roles), são elas “ADMIN” e “USER”.
RNF-07	Somente pessoas autorizadas podem fazer login no sistema, através dos dados de usuário e senha corretos.
RNF-08	O sistema de backend deve ser implementado: em Java utilizando SpringFramework e SpringBoot.
RNF-09	O sistema de frontend deve ser implementado: HTML, CSS, JavaScript, AngularJS e TypeScript.
RNF-10	O sistema deverá persistir os dados em uma instância do banco de dados PostgreSQL.
RNF-11	O sistema deve notificar o sucesso ou erro para o usuário, ao realizar cadastros, edições, consultas e exclusões através de Toasts.
RNF-12	O sistema deve alertar aos seus usuários sobre os campos obrigatórios para a realização dos cadastros.
RNF-13	Todas as requests devem ser autenticadas e autorizadas pelo Keycloak.

Fonte: elaboração dos autores (2023)

3.3 DIAGRAMAS

Com o uso da Linguagem de Modelagem Unificada - UML, é possível classificar, visualizar e documentar as características do sistema, assemelhando-se a uma "planta baixa" do sistema a ser desenvolvido. Essa representação é feita por meio de vários diagramas que têm como propósito compreender o sistema, dispensando a necessidade de entender o código-fonte (GONÇALVES, 2015; CORTÉS, 2015). Para criar os diagramas, seguindo os padrões UML, foram utilizadas as ferramentas gratuitas Lucidchart e Draw.io. Neste tópico, são apresentados os principais diagramas que compõem o sistema STM.

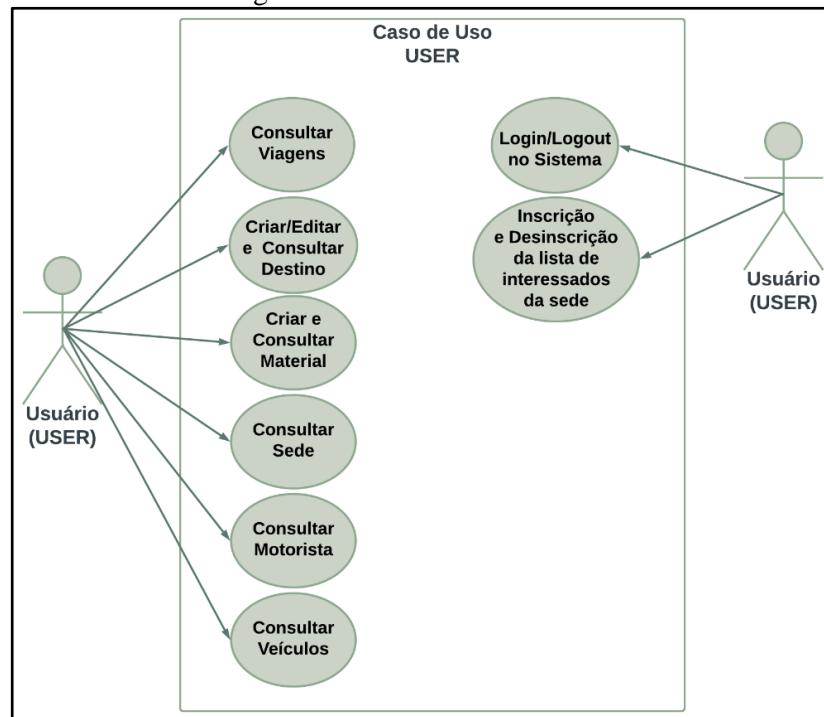
A Figura 5 representa as competências do usuário identificado como "ADMIN" no sistema, apresentando suas permissões e atividades de manutenção (cadastro, edição, consulta e remoção), além de outras funcionalidades relacionadas a viagens, sedes e destinos.



Fonte: elaboração dos autores (2023)

A Figura 6 apresenta às competências do usuário identificado como "USER" no sistema, demonstrando suas atividades, sendo possível somente realizar a criação, edição e consulta de determinados objetos no sistema e ainda possibilitando colocar ou retirar seu nome de uma lista referente as sedes da empresa.

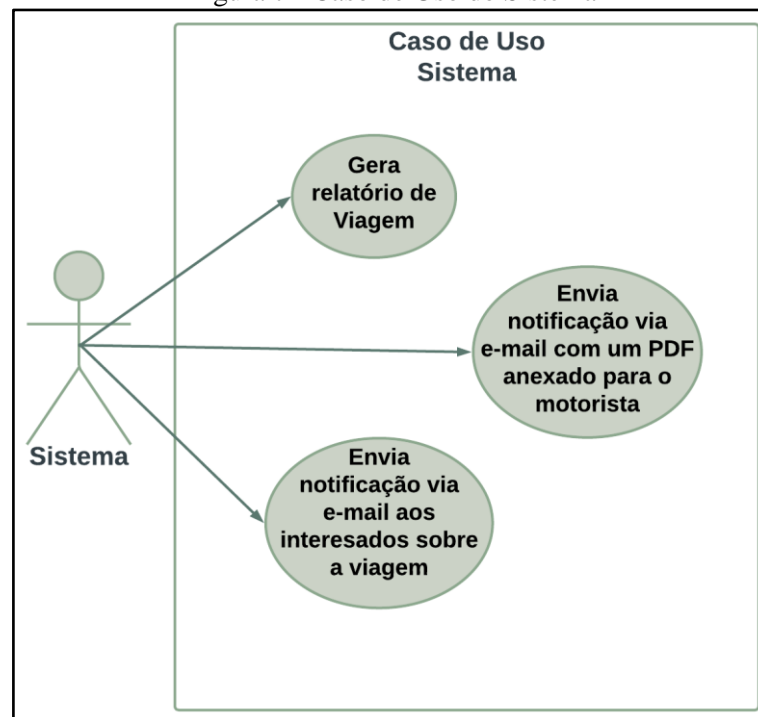
Figura 6 - Caso de Uso do USER



Fonte: elaboração dos autores (2023)

A Figura 7 demonstra as funcionalidades em que o sistema “STM” deve apresentar aos seus usuários (ADMIN ou USER), através de determinadas ações realizadas no sistema.

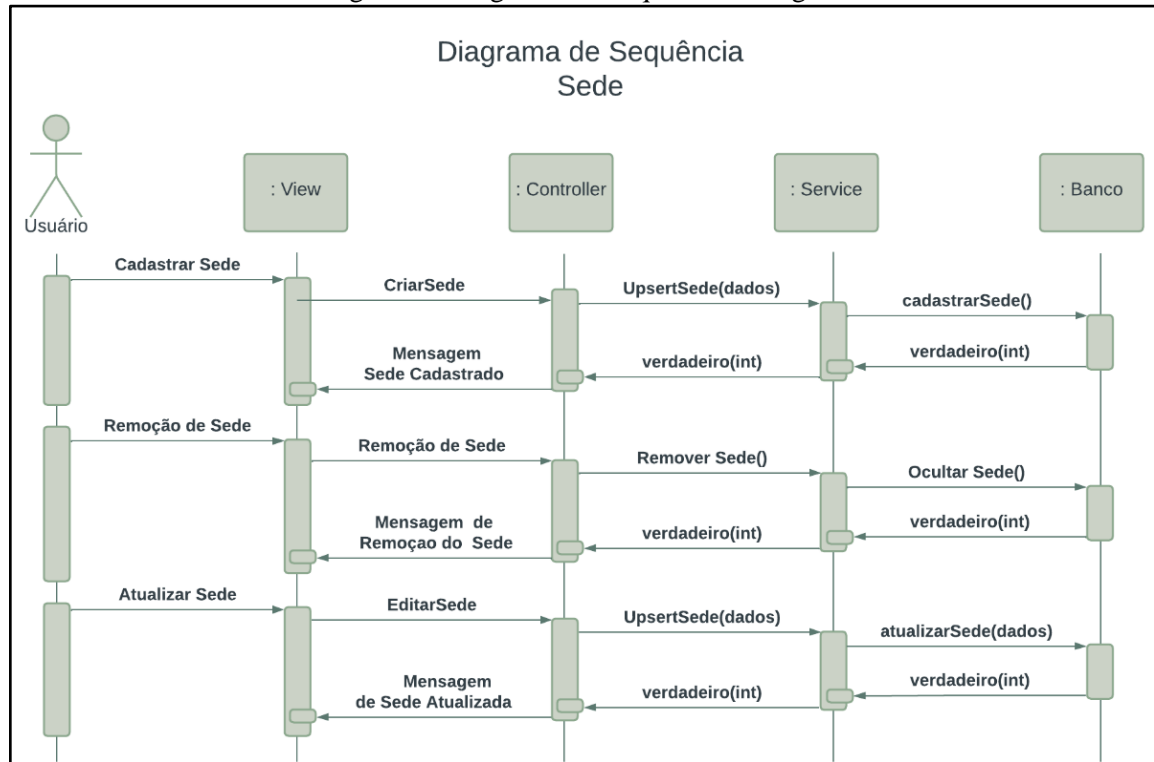
Figura 7 - Caso de Uso do Sistema



Fonte: elaboração dos autores (2023)

A Figura 8 representa o funcionamento de cadastrar, editar e remover a sede no sistema.

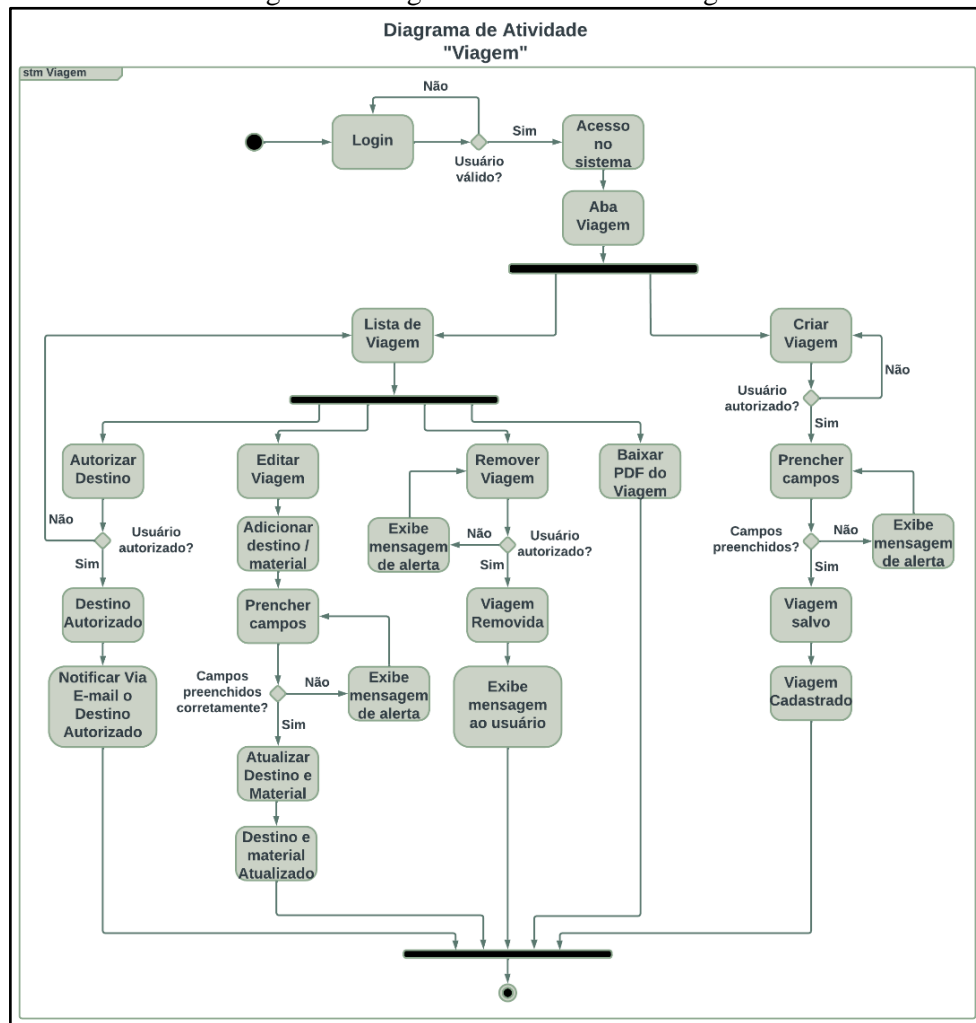
Figura 8 - Diagrama de Sequência / Viagem



Fonte: elaboração dos autores (2023)

A Figura 9 O diagrama de atividade mostra como os usuários navegaram no sistema “STM” começando pela tela login e após o redirecionamento será possível visualizar outras funcionalidades como criar, consultar, remover, editar viagem e a baixar o PDF daquela viagem.

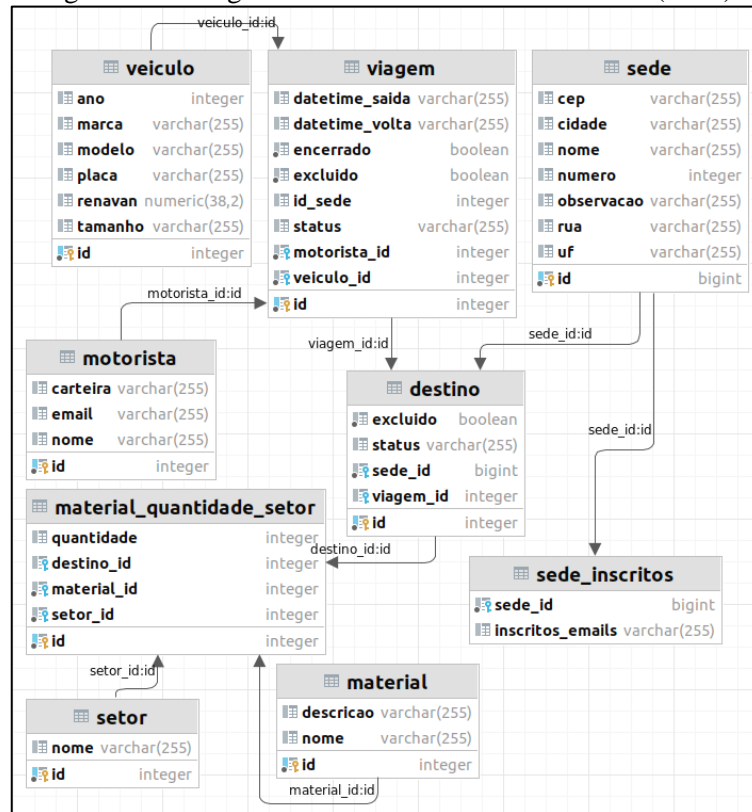
Figura 9 - Diagrama de Atividade / Viagem



Fonte: elaboração dos autores (2023)

A Figura 10 apresenta o DER (Diagrama Entidade Relacionamento) a estrutura do banco de dados do sistema STM, a criação deste modelo de dados foi realizada automaticamente pela interface do IntelliJ Idea Ultimate com base nos modelos criados anteriormente usando ORM (Object-Relational Mapping), nela, é possível notar as principais entidades, atributos e relacionamentos das entidades do sistema.

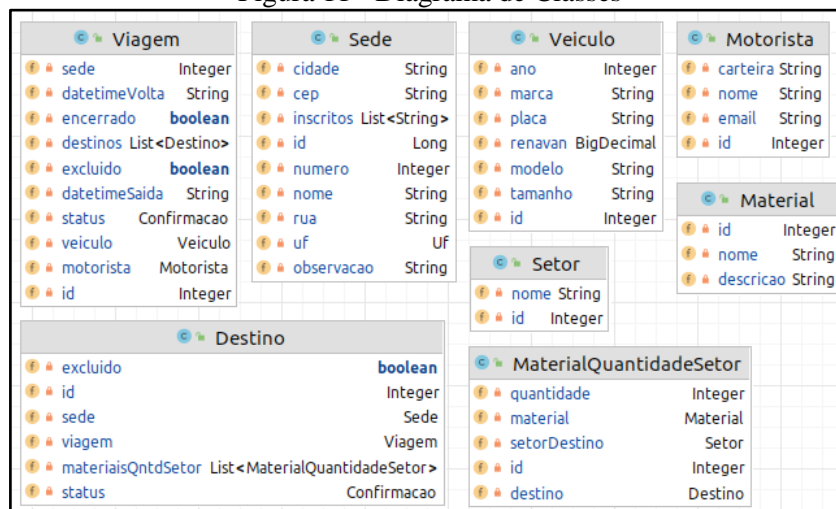
Figura 10 - Diagrama de Entidade-Relacionamento (DER)



Fonte: elaboração dos autores (2023)

O STM foi modelado usando DDD (Domain Driven Design ou Design Orientado ao Domínio) e na Figura 11 são apresentadas grande parte das classes desse domínio. No total elas se constituem de 10 classes/entidades, apenas duas delas não são apresentadas neste diagrama são elas UF e Confirmação.

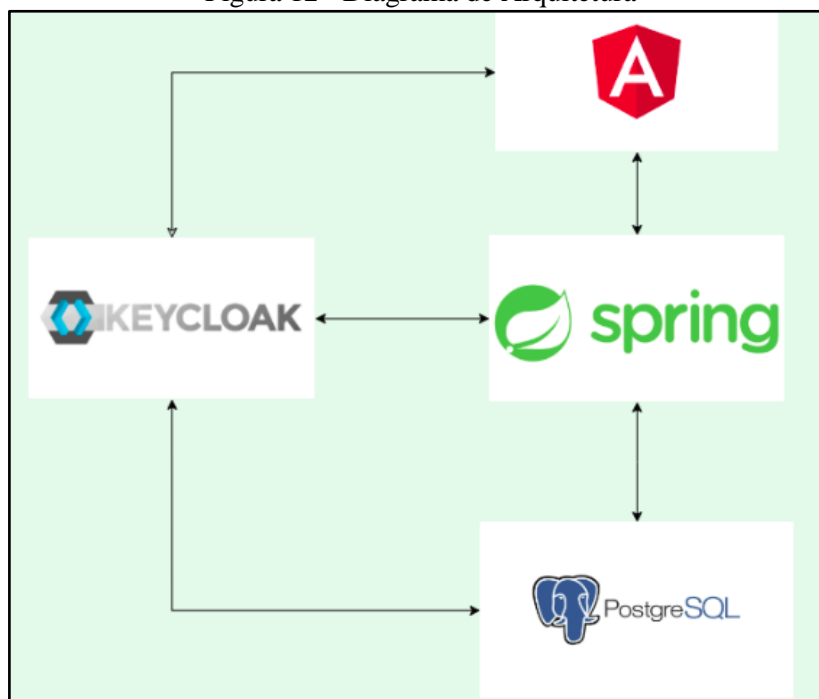
Figura 11 - Diagrama de Classes



Fonte: elaboração dos autores (2023)

A Figura 12 representa em alto grau de abstração a arquitetura proposta para o STM, e é composta por quatro elementos, a saber: Aplicação Web desenvolvida com o framework Angular, o Keycloak um provedor de autenticação que é de código aberto, a aplicação em si que suporta as principais funcionalidades do sistema desenvolvida em Spring Framework e um robusto banco de dados o PostgreSQL.

Figura 12 - Diagrama de Arquitetura



Fonte: elaboração dos autores (2023)

4 DESENVOLVIMENTO

Nesta seção serão descritas as atividades diretamente relacionadas com o desenvolvimento do sistema propriamente dito, serão expostas as decisões do projeto, a construção e desenvolvimento das telas.

4.1 DESCRIÇÃO DO PROJETO

O projeto alvo deste trabalho é um sistema de apoio logístico com a intenção de reduzir os custos financeiros e de tempo gastos com viagens desnecessárias e redundantes, bem como auxiliar na eficácia da comunicação quando diversos atores precisam enviar remessas para outras filiais ou sedes de uma mesma empresa.

A aplicação foi elaborada para contemplar à priori dois tipos de usuários, os administradores chamados de “ADMIN” e os usuários comuns denominados “USER” ambas as denominações são utilizadas apenas internamente ao sistema. Estas denominações têm por objetivo permitir que dois atores distintos, com responsabilidades distintas dentro da empresa realizem tarefas diferentes dentro do sistema.

Os usuários ADMIN são responsáveis pela criação, edição, encerramento e confirmação/cancelamento das principais rotinas do software. Algumas das suas atribuições contemplam criar viagens, inserir destinos nas viagens registradas, confirmar, cancelar ou até mesmo encerrar uma viagem, dentre outras funcionalidades descritas na “Tabela 1 - Requisitos funcionais do sistema STM” presente na subseção 3.2 deste documento.

Os usuários USER tem menos responsabilidades se comparados aos ADMINs, dentre suas atribuições estão consultar viagens registradas, solicitar a inclusão de uma nova viagem ou destino, incluir novos materiais a serem transportados em uma determinada viagem, cadastrar motoristas e veículos, dentre outros requisitos que estão mais bem descritos na “Tabela 1 - Requisitos funcionais do sistema STM” presente na subseção 3.2 deste documento.

Um procedimento muito relevante que contempla ambos os usuários são a inclusão e exclusão de seus e-mails em listas de observação, se um dado usuário deseja ser notificado toda vez que uma sede específica for alvo de uma parada em uma viagem ele pode se inscrever em uma lista de e-mails e ser notificado toda vez que essa sede em questão for adicionada à uma viagem. O usuário pode remover seu e-mail a qualquer tempo e então cessarão os e-mails recebidos referentes àquela sede.

4.2 DECISÕES DO PROJETO

O desenvolvimento do STM foi pensado para ser comercializado de duas maneiras como SaaS (Software as a Service).

No primeiro caso, o cliente, sob pagamento de uma mensalidade é capaz de acessar a aplicação remotamente de onde estiver, utilizando suas credenciais, para performar as ações inerentes ao seu papel de usuário dentro da aplicação. Esta licença permite até um certo número de usuários registrados dependendo do plano contratado (limite de usuários será implementado futuramente e planos serão definidos).

No segundo caso, a empresa contratante recebe todo o apoio necessário para acoplar seu sistema, já em uso, à nossa plataforma utilizando assim nosso sistema como um módulo do sistema do contratante. Para este intento será disponibilizada uma especificação OpenAPI definida em arquivo com a extensão YAML que contém todos os dados necessários para este acoplamento, bem como uma chave de serviço para transferência como um header x-apikey em cada requisição ao nosso serviço (chave de serviço será implementada futuramente, bem como os planos de cobrança e as regras para número de requisições ou por processamento).

Levando os modelos, planos de faturamento, os fluxos e modelagens feitas anteriormente foi definido pela equipe que seriam criados dois projetos seguindo a atual tendência do mercado de desenvolvimento de software com relação aos microsserviços.

Para a parte visual, e primeiro projeto, foi escolhido o Angular um framework de desenvolvimento de aplicações web de código aberto, mantido pelo Google e que é amplamente utilizado para criar interfaces interativas e dinâmicas. Esta ferramenta utiliza o padrão de arquitetura MVC um padrão já bastante corroborado no mercado de desenvolvimento web que permite a clara separação de responsabilidades entre as camadas durante o progresso da aplicação. Para que a equipe pudesse contribuir com o desenvolvimento do frontend foi criado um repositório no Github.

O segundo projeto refere-se ao backend consumido pela aplicação frontend. Para esta etapa foram tomadas decisões que facilitassem o desenvolvimento, bem como o acoplamento com o frontend e com outras plataformas e sistemas. Foi desenhada uma especificação OpenAPI, um contrato, entre backend e frontend que serviu como farol para todas as decisões de design de código. A fim de permitir que todos os integrantes da equipe pudessem participar do desenvolvimento da aplicação foi criado um repositório no Github.

O banco de dados seria a próxima decisão crucial para o início do desenvolvimento da plataforma, foi escolhido o banco de dados relacional PostgreSQL por sua robustez e facilidade de uso e também porque um terceiro elemento da nossa arquitetura é perfeitamente acoplável a este banco de dados, o Keycloak, uma ferramenta de autenticação também bastante relevante no mercado de segurança da informação. Esta ferramenta abstrai muito da complexidade da manutenção de cadastros de usuários, papéis, permissões e tudo mais relativo à autenticação e autorização.

A última ferramenta abordada provê o seguinte recurso que encerra a arquitetura do STM que é o JWT um padrão aberto para tokens de acesso baseados em JSON, uma forma compacta de transmitir informações entre dois elementos de uma arquitetura de sistemas à saber: cliente-servidor e servidor-servidor. O JWT permite que as informações sejam assinadas digitalmente utilizando algoritmos de criptografia, garantindo assim a integridade dos dados e evita que estes sejam alterados durante a transmissão. Outra vantagem do token é que após a autenticação do usuário suas informações relevantes ficam “persistidas” no token e basta o usuário reenviar o token ao servidor para que este verifique sua autenticidade permitindo que o usuário realize a ação desejada sem precisar se autenticar novamente.

Na Figura 13, são apresentadas as logomarcas de algumas das tecnologias apresentadas anteriormente neste documento e utilizadas para o desenvolvimento deste projeto.

Figura 13 - Ferramentas Utilizadas



Fonte: elaboração dos autores (2023).

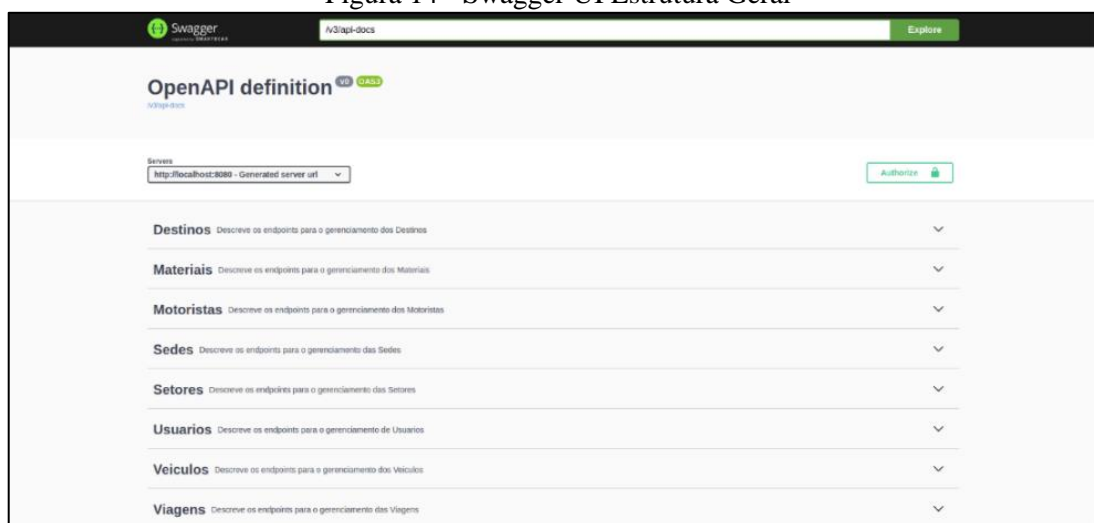
4.3 CONSTRUÇÃO DO PROJETO

O primeiro passo para o desenvolvimento do BackEnd foi utilizando um recurso da IDE IntelliJ Ultimate a criação de um novo Projeto Spring com Java e Maven. Na sequência dois submódulos foram criados, um para recepcionar a especificação OpenAPI e outro para servir como implementação do serviço Figura 16.

A etapa seguinte consistiu em redigir o contrato de OpenAPI para a comunicação com o FrontEnd permitindo assim o início dos trabalhos desse projeto e o fácil acoplamento com outros sistemas. Neste módulo também são gerados os artefatos necessários para a implementação dos controllers no módulo de serviço, bem como a geração dos artefatos clientes utilizados em sistemas Java que desejem se conectar ao nosso sistema de BackEnd, a disponibilidade desses artefatos pode ser feita em um Registry público ou privado como o JFrog ou o Nexus, se houver necessidade pode-se inclusive disponibilizar nos repositórios oficiais de cada linguagem de programação suportada como o próprio Maven Central para Java ou no NPM para NodeJs.

A interface do usuário (UI) gerada pelo Swagger para nossa aplicação Java é excelente para facilitar a integração de outros sistemas ao nosso. A sua interface amigável e completa, oferece uma documentação interativa e visualmente atraente, que permite que os desenvolvedores visualizem e entendam facilmente a estrutura da API do nosso sistema como demonstrado na **Erro! Fonte de referência não encontrada..**

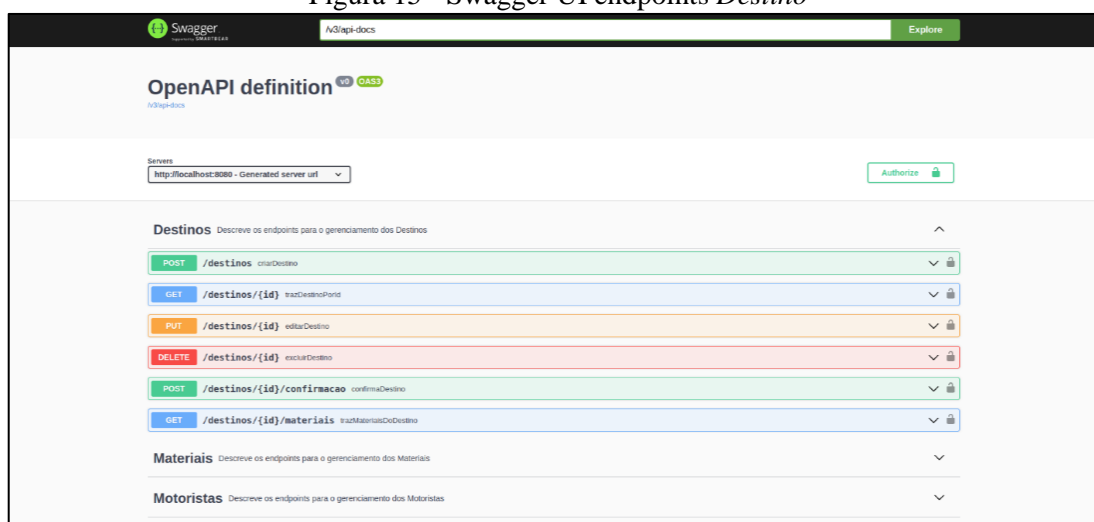
Figura 14 - Swagger UI Estrutura Geral



Fonte: elaborado pelos autores (2023).

Isso aumenta a atratividade do nosso sistema para integrações externas, promovendo a colaboração e a expansão do STM como uma solução confiável e de fácil integração para outras aplicações. Com a UI (User Interface) do Swagger, os desenvolvedores podem explorar e testar as diferentes rotas e endpoints disponíveis em nossa API, como na Figura 15, que mostra os endpoints referentes ao *Destino*. Isso proporciona uma experiência mais intuitiva e eficiente, pois os desenvolvedores podem visualizar as opções de entrada, parâmetros e respostas esperadas para cada chamada de API.

Figura 15 - Swagger UI endpoints *Destino*



Fonte: elaborado pelos autores (2023).

Além disso, a UI do Swagger permite que os desenvolvedores gerem automaticamente códigos de cliente em várias linguagens de programação, como Java, Python ou JavaScript. Essa funcionalidade acelera o processo de integração, fornecendo aos programadores um ponto de partida sólido para consumir nossos serviços e integrar seus sistemas ao STM.

Ao longo do projeto pontuais alterações foram necessárias nesse contrato apresentado, porém nada que tivesse médio ou grande impacto no trabalho já implementado do FrontEnd. Dentre as principais mudanças podem-se citar alterações nas descrições de métodos, adição de novos campos não obrigatórios nas requisições ao servidor, bem como a criação de novos EndPoints por demandas externas.

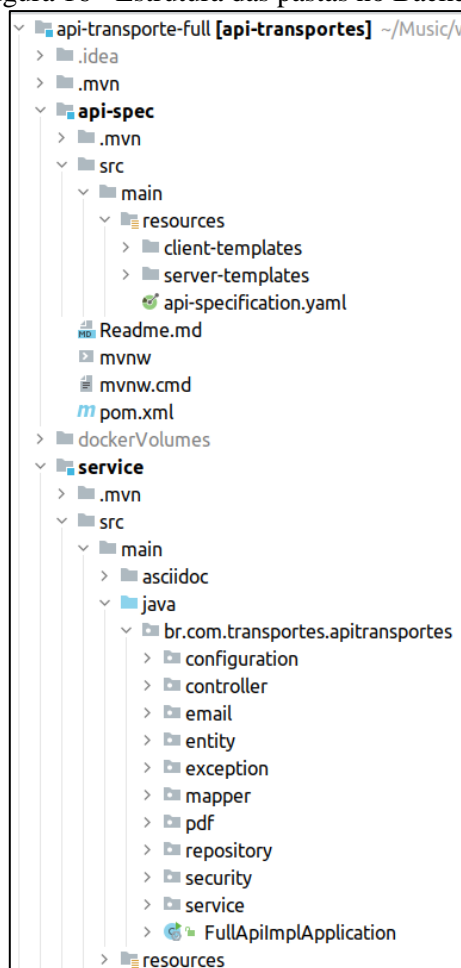
No módulo de serviço as entidades do sistema foram modeladas utilizando o JPA (Jakarta Persistence Api) uma especificação para bibliotecas de ORM que utilizam orientação

à objetos que também está presente na linguagem Java para mapear objetos do sistema com tabelas e colunas no banco de dados.

O desenvolvimento se seguiu com sucessivas iterações implementando no módulo de serviço as especificações definidas anteriormente para cada EndPoint. Os trabalhos se deram das entidades menos complexas para as mais complexas. Primeiramente foram criadas as implementações para *Setor*, *Material*, *Motorista*, *Veículo*, *Sede*, *MaterialQuantidadeSetor*, *Destino* e por fim *Viagem*.

Posteriormente foram agregadas novas funcionalidades além das comuns criar, listar, atualizar e deletar, dentre estes novos recursos podemos citar a deleção lógica das entidades onde o registro não é excluído do banco de dados, mas para a aplicação é como se fosse, a geração de relatórios em PDF e o envio de e-mails.

Figura 16 - Estrutura das pastas no Backend

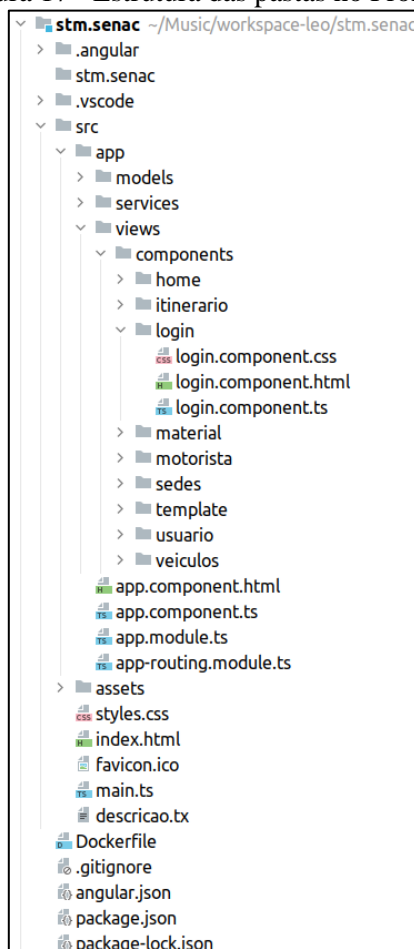


Fonte: elaborado pelos autores (2023).

Para o desenvolvimento do FrontEnd, foi utilizada a IDE WebStorm. A divisão das funcionalidades presentes no projeto foi feita em módulos, como Models, Services e Views. A pasta Models contém todas as interfaces utilizadas, a pasta Services contém todas as funções responsáveis por se comunicar com o BackEnd por meio de requisições HTTP, e a pasta Views contém os componentes do FrontEnd, como home, viagem, login, material, motorista, sedes e veículos.

Cada componente View contém três partes: um arquivo HTML, um CSS e um TypeScript. Cada um desempenha um papel fundamental na renderização daquele componente. O HTML é a base para a construção da interface do usuário. No Angular, um componente é uma combinação de um template HTML, que define a estrutura e o conteúdo, com um arquivo CSS que é utilizado para estilizar os componentes HTML e aplicar estilos específicos a eles. Já o componente TypeScript é usado para controlar a lógica e o comportamento dos componentes. Eles são responsáveis por lidar com os dados, responder a eventos e interagir com outros componentes e serviços. Na Figura 17 é demonstrada a estrutura das pastas para um melhor entendimento visual, pode-se notar os três componentes presentes na pasta login como um exemplo.

Figura 17 - Estrutura das pastas no Frontend



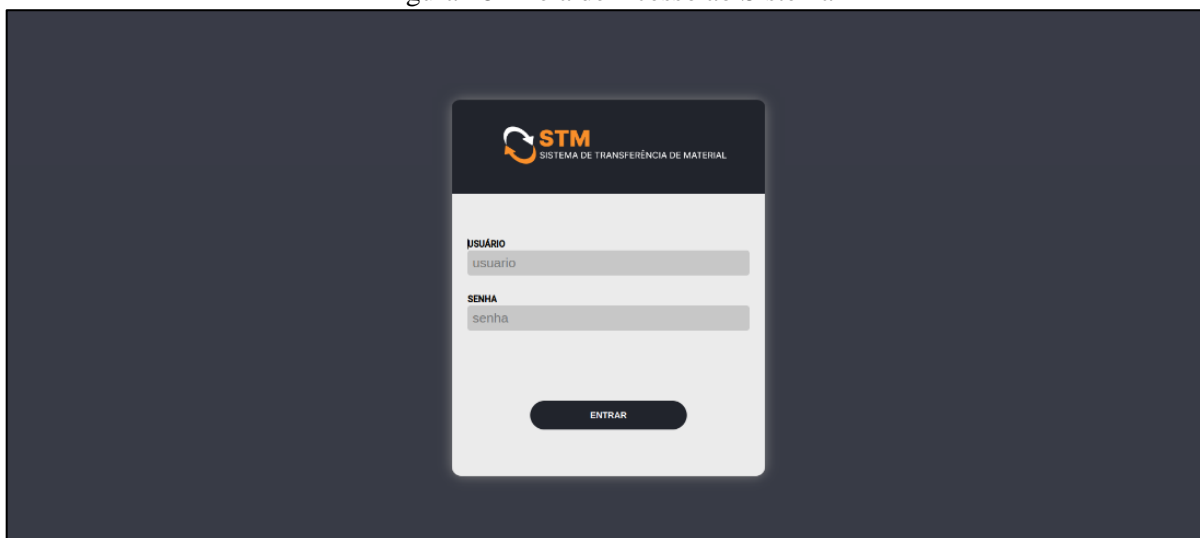
Fonte: elaborado pelos autores (2023).

Na finalização da implementação, foram criadas 13 telas presentes no sistema. É possível acessar todas essas telas tanto por parte do usuário "ADMIN" quanto pelo usuário "USER". O controle de acesso é gerenciado pela camada Controller, no *backend*, onde existe uma anotação chamada @PreAuthorize em cada método protegido. Essa configuração pertence ao Spring Security e facilita a segregação de permissões com especificidade a nível de métodos.

Para iniciar o acesso do usuário no sistema, é realizada a autenticação na aplicação, conforme ilustrado na Figura 18. Todos os usuários obtêm acesso ao sistema após suas credenciais de login e senha serem validadas. Essas informações são capturadas pelo frontend no LoginComponent e enviadas via requisição REST (Transferência de Estado Representacional) para o endpoint '/realms/api-transportes/protocol/openid-connect/token' do Keycloak. O Keycloak, por sua vez, realiza o processamento necessário para retornar uma resposta indicando a validade do usuário. Se o usuário fornece corretamente o login e senha, a ferramenta de autenticação retornará um token válido, e o usuário será redirecionado para a

próxima tela. Caso contrário, o usuário permanecerá na tela atual e receberá avisos em vermelho nos campos de "Senha e Usuário incorretos".

Figura 18 - Tela de Acesso ao Sistema

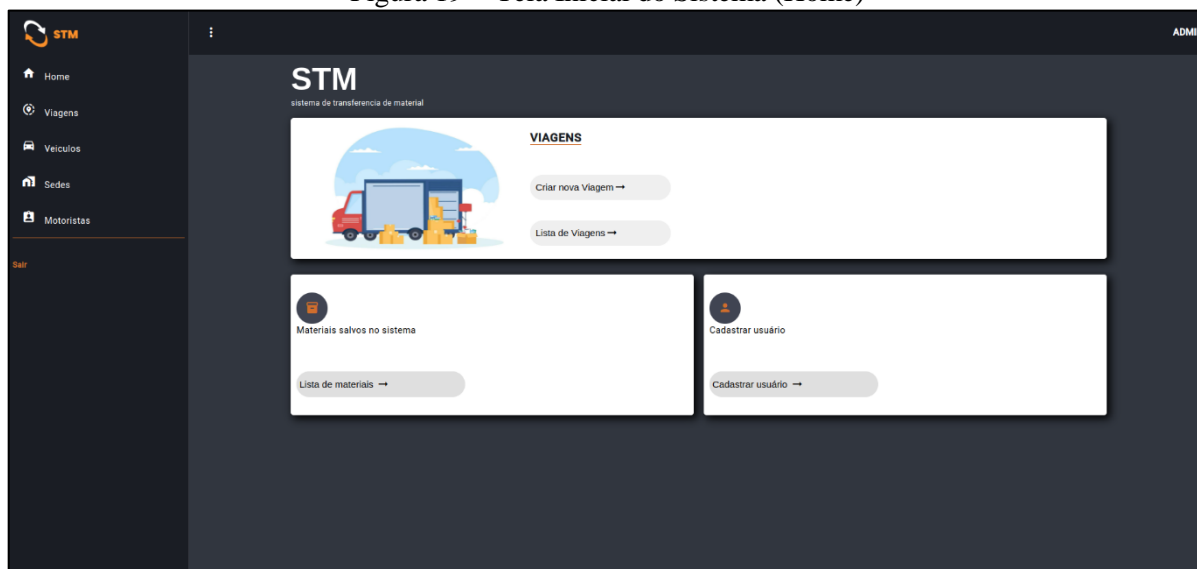


Fonte: elaborado pelos autores (2023).

Após ser redirecionado o usuário se encontrará na Tela Inicial do sistema (Home) Figura 19. No menu sidenav expansível e responsivo nativo do Angular Material ele pode navegar pelas opções viagens, veículos, sedes, motoristas, manual e o botão de sair.

No centro desta tela ainda é possível acessar outras partes do sistema como criar uma nova viagem ou listá-las, acessar a lista de materiais já cadastrados, bem como cadastrar um novo usuário. A tela home encontra-se no projeto do FrontEnd em `/src/app/views/home` sendo distribuída em três componentes sendo CSS, HTML e TypeScript.

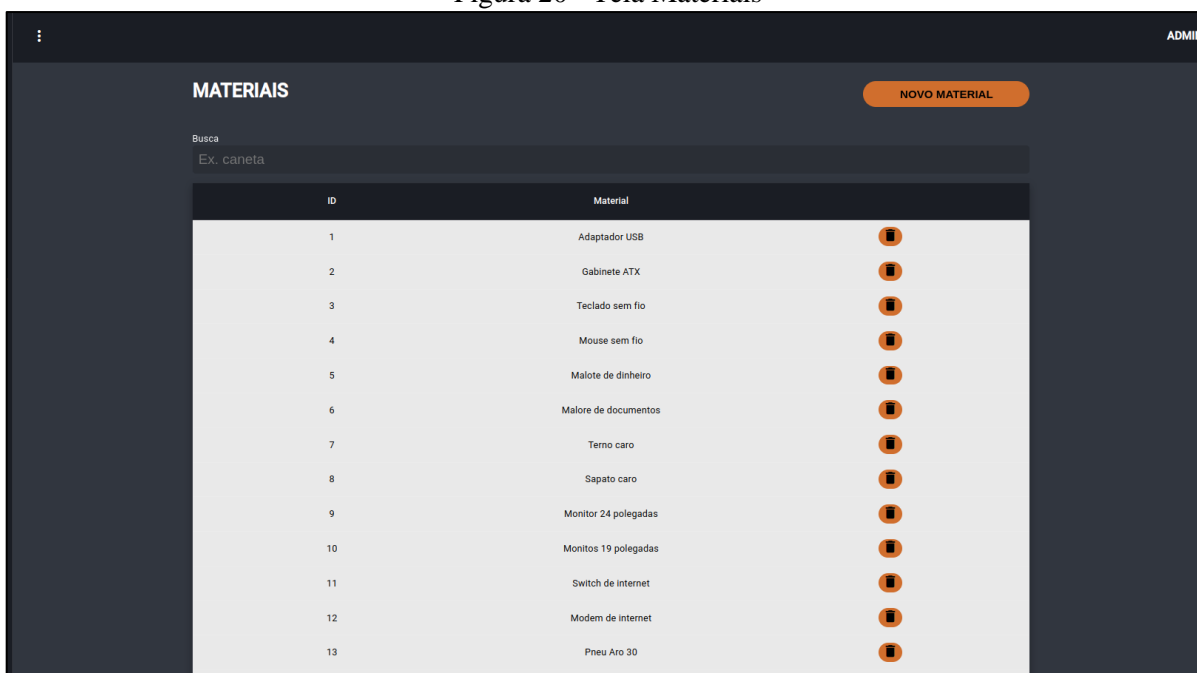
Figura 19 - Tela Inicial do Sistema (Home)



Fonte: elaborado pelos autores (2023).

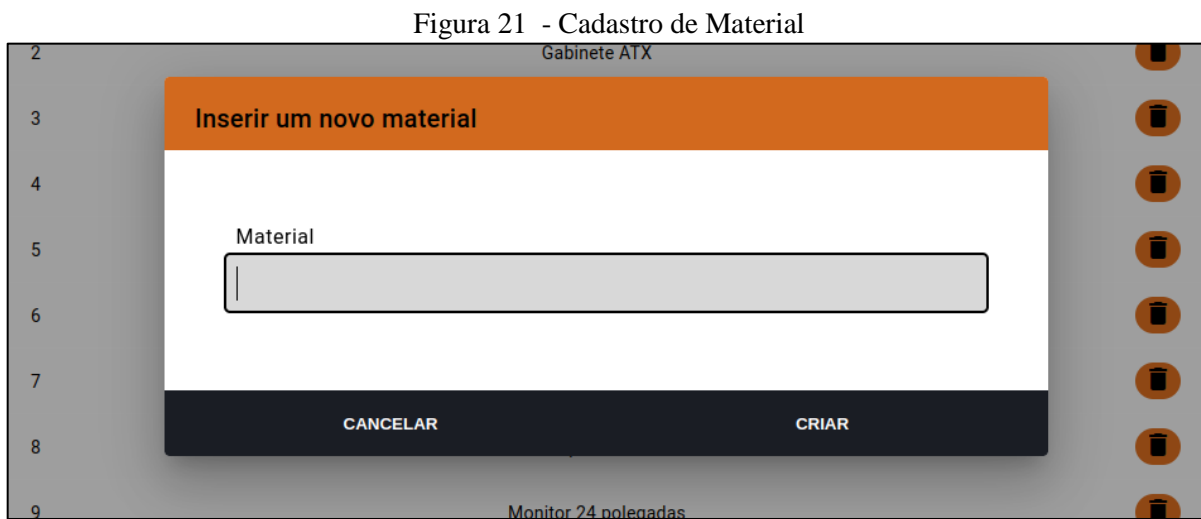
Ao clicar no botão “Lista de todos os materiais” o usuário é redirecionado para a tela de gerenciamento de materiais Figura 20. Nesta tela são possíveis as seguintes operações: listar todos os materiais (comportamento padrão ao abrir a página), cadastrar um novo material ao clicar no botão “Novo Material” e excluir um já cadastrado clicando no ícone de lixeira. Um campo de texto está presente e ao digitar aparecerão na tabela apenas os materiais que contenham os argumentos digitados, a pesquisa é instantânea.

Figura 20 - Tela Materiais



Fonte: elaborado pelos autores (2023).

Na Figura 21 abaixo mostra como é cadastrado um novo material, sendo necessário apenas seu nome. O componente responsável pela captura e posterior processamento desta tela é o `MaterialController` presente em `stm.senac-main/src/app/views/material`.



Fonte: elaborado pelos autores (2023).

Detalhando um pouco mais os recursos da Home têm-se o botão de cadastrar usuário que leva o usuário para a Figura 22 onde um novo usuário pode ser cadastrado. Como este tipo de registro na ferramenta de autenticação escolhida, o Keycloak, é bastante complexo foi criada uma camada de abstração onde apenas um formulário é enviado do frontend para o backend e neste toda a complexidade do cadastro é gerenciada.

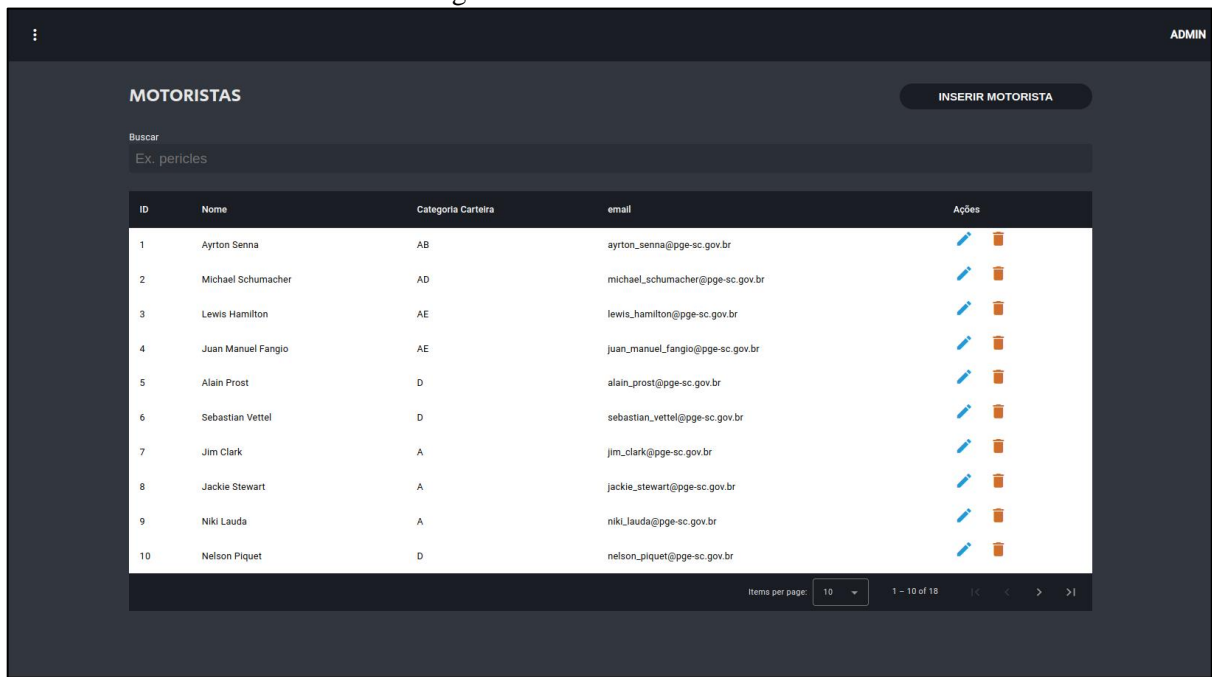
Figura 22 - Cadastro de Usuário

A interface de Cadastro de Usuário apresenta um formulário centralizado em uma página de fundo escura. No topo da página, há um ícone de menu hambúrguer à esquerda e o texto 'ADMIN' à direita. Abaixo do menu, um link '< Home' em laranja indica o caminho de volta. O formulário, intitulado 'Cadastro de usuário', contém os seguintes campos: 'Nome do usuário', 'Senha', 'Nome', 'Sobrenome', 'Email' e 'Tipo do usuário' (um menu suspenso com uma seta para baixo). Um botão laranja com o texto 'CADASTRAR' em branco está posicionado na base do formulário.

Fonte: elaborado pelos autores (2023).

A figura seguinte (Figura 23) diz respeito às operações relativas aos motoristas e pode ser acessada pelo menu sidenav. O estado inicial desta página mostra, usando paginação, uma lista com todos os motoristas cadastrados no sistema, também é possível notar uma barra superior de pesquisa usada para encontrar um ou mais motoristas ao digitar fragmentos de nome ou e-mail. Um botão lápis faz referência à funcionalidade de edição de motoristas que pode ser verificada na Figura 24. Já o botão de lixeira remete à funcionalidade de exclusão daquele cadastro específico (Figura 25).





















Figura 23 – Tela de Motoristas



MOTORISTAS ADMIN

INSERIR MOTORISTA

Buscar
Ex. pericles

ID	Nome	Categoria Carteira	email	Ações
1	Ayrton Senna	AB	ayrton_senna@pge-sc.gov.br	 
2	Michael Schumacher	AD	michael_schumacher@pge-sc.gov.br	 
3	Lewis Hamilton	AE	lewis_hamilton@pge-sc.gov.br	 
4	Juan Manuel Fangio	AE	juan_manuel_fangio@pge-sc.gov.br	 
5	Alain Prost	D	alain_prost@pge-sc.gov.br	 
6	Sebastian Vettel	D	sebastian_vettel@pge-sc.gov.br	 
7	Jim Clark	A	jim_clark@pge-sc.gov.br	 
8	Jackie Stewart	A	jackie_stewart@pge-sc.gov.br	 
9	Niki Lauda	A	niki_lauda@pge-sc.gov.br	 
10	Nelson Piquet	D	nelson_piquet@pge-sc.gov.br	 

Items per page: 10 1 - 10 of 10 < > >|

Fonte: elaborado pelos autores (2023).

Figura 24 - Edição de Motorista



ATUALIZAÇÃO DE MOTORISTA ADMIN

Nome
Ayrton Senna

Categoria da Carteira
AB

Email
ayrton_senna@pge-sc.gov.br

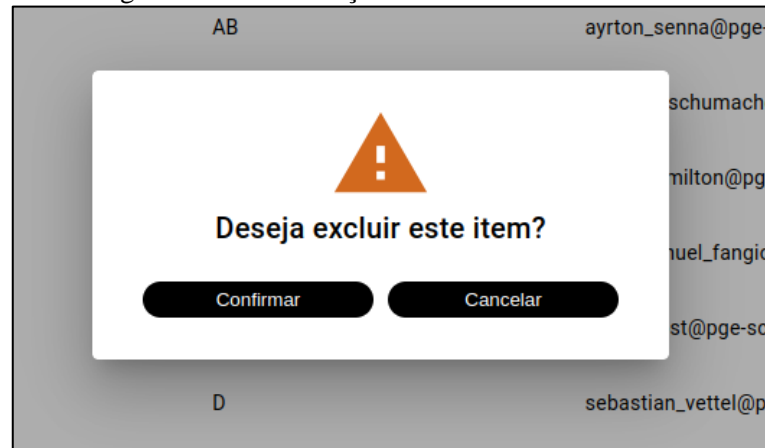
ATUALIZAR

STM

- Home
- Viagens
- Veiculos
- Sedes
- Motoristas

Fonte: elaborado pelos autores (2023).































Figura 25 - Confirmação de exclusão de Motorista



Fonte: elaborado pelos autores (2023).

A (Figura 26) seguinte diz respeito às operações relativas as sedes e pode ser acessada pelo menu sidenav. O estado inicial desta página mostra, usando paginação, uma lista com todas as sedes cadastradas no sistema, também é possível notar uma barra superior de pesquisa usada para encontrar um ou mais sedes ao digitar fragmentos de dados presentes em quaisquer um dos campos. Um botão lápis faz referência à funcionalidade de edição de sedes que pode ser verificada na Figura 27. Já o botão de lixeira remete à funcionalidade de exclusão daquele cadastro específico (Figura 28).

Figura 26 - Tela de Sedes

SEDES								
ADMIN								
INserir SEDE								
Buscar								
Ex. regional 1								
ID	Nome	Numero	Cep	Cidade	UF	Rua	Observação	Ações
1	Procuradoria Regional de São Miguel do Oeste	134	89900-000	São Miguel do Oeste	SC	R. Santos Dumont	R. Santos Dumont, 134 - Centro, São Miguel do Oeste - SC	  
2	Procuradoria Regional de Chapecó	985	89802-210	Chapecó	SC	Avenida General Osório - D	Av. General Osório, 985 D - Ed. Alleanza - Centro, Chapecó - SC CEP: 89802-210	  
3	Procuradoria Regional de Joaçaba	205	89600-000	Joaçaba	SC	R. Getúlio Vargas	R. Getúlio Vargas, 205 - 5º andar - Joaçaba, SC, 89600-000 +55 49 3527-9619	  
4	Procuradoria Regional de Caçador	185	89500-000	Caçador	SC	R. Fernando Machado	Telefones: (49) 3561-6465 / 3561-6469	  
7	Procuradoria Regional de Mafra	1405	89304-060	Mafra	SC	Rua Tenente Ary Rauhen	Telefones: (47) 3647-0310 / 3647-0311 Sala 02	  
8	Procuradoria Regional de Rio do Sul	83	89160-182	Rio do Sul	SC	Rua Marechal Rondon	Telefone: (47) 3526-3137	  
9	Procuradoria Regional de Criciúma	1011	88801-100	Criciúma	SC	Rua Coronel Marcos Roarvis	Telefones: (48) 3403-1181 / 3403-1183 Sala 230	  
10	Procuradoria Regional de Tubarão	150	88701-150	Tubarão	SC	Rua Tubalcaim Faraco	Telefones: (48) 3631-9439 (Geral) / 3631-9432 (Geral) / 3631-9443 (Fiscal) / 3631-9437 (Contencioso)	  
11	Procuradoria Regional de Blumenau	118	89010-170	Blumenau	SC	Rua Paulo Zimmermann	+55 47 3378-8425 / 3378-8423	  
12	Procuradoria Regional de Jaraguá do Sul	933	89251-100	Jaraguá do Sul	SC	Rua Presidente Epitácio Pessoa	Telefones: (47) 3276-9350 / 3276-9353	  

Fonte: elaborado pelos autores (2023).

Figura 27 - Edição de Sede

ATUALIZAR DE SEDE

Nome: Procuradoria Regional de Chapecó

CEP: 89802-210 Rua: Avenida General Osório - D Número: 985

Cidade: Chapecó UF: SC

Observações: Av. General Osório, 985 D - Ed. Alleanza - Centro, Chapecó - SC
CEP: 89802-210

ATUALIZAR

Fonte: elaborado pelos autores (2023).

Figura 28 - Confirmação de exclusão de Sede

Deseja excluir este item?

Confirmar **Cancelar**

134	89900-000	São Miguel do Oeste	SC	R. Santos Dumont
985				
205				
185				
1405				
83	89160-182	Rio do Sul	SC	Rua Marechal Rondon

Fonte: elaborado pelos autores (2023).

Ainda na página de gerenciamento de sedes é possível que o usuário se inscreva ou se desinscreva na lista de notificações daquela sede, para tal ele deve clicar no ícone de usuário logo à direita do ícone de lixeira colocar seu email e então clicar em Inscrever-se para ter seu email cadastrado na lista de notificação ou em Desinscrever para ter seu cadastro removido dessa lista como mostrado na Figura 29.

Figura 29 – Inscrever-se / Desinscrever-se na lista de Interessados da Sede

Insira o seu email para ser notificado quando ocorrer uma viagem que tenha como destino Procuradoria Regional de São Miguel do Oeste ou insira o email que deseja remover da lista de notificação desta sede

Email

DESINSCREVER INSCREVER-SE

Fonte: elaborado pelos autores (2023).

A figura seguinte (Figura 30) diz respeito às operações relativas aos veículos e pode ser acessada pelo menu sidenav. O estado inicial desta página mostra, usando paginação, uma lista com todos os veículos cadastrados no sistema, também é possível notar uma barra superior de pesquisa usada para encontrar um ou mais veículos ao digitar fragmentos de quaisquer dados dessa entidade. Um botão lápis faz referência à funcionalidade de edição de veículos que pode ser verificada na Figura 31. Já o botão de lixeira remete à funcionalidade de exclusão daquele cadastro específico (Figura 32). O cadastramento de veículos está sendo ilustrado pela Figura 33 e são aceitas placas de veículos registradas no Brasil no padrão antigo ou registradas no padrão Mercosul.

Figura 30 - Tela de Veículos

VEICULOS ADMIN INSERIR VEICULO

Buscar
Ex. marea

Id	Modelo	Marca	Placa	Ano	Renavan	Tamanho	Ações
2	Fiorino	Fiat	UIC2V33	2021	1914569752	Médio	
3	Spin	Chevrolet	HAIZL36	2019	7865197902	Médio	
1	Hyundai HR	Hyundai	ABC-2223	2020	3214569871	3,58m	
4	Duster	Renault	OPQ8P11	20123	16450237922	Médio	
5	Siena	Fiat	ABA1C23	20118	33164205913	Médio	
6	Master	Renault	BAB1E01	2008	90643018012	Grande	

Items per page: 10 1 - 6 of 6

Fonte: elaborado pelos autores (2023).

Figura 31 - Edição de Veículo

The screenshot shows a web application interface for updating a vehicle. On the left is a dark sidebar with the STM logo and navigation links: Home, Viagens, Veiculos, Sedes, and Motoristas. The main area has a dark header with a back arrow and the title 'ATUALIZAÇÃO DE VEICULO', and a user role 'ADMIN' in the top right. A light gray form box contains the following fields:

Marca	Fiat	Modelo	Fiorino
Placa	UIC2V33	Ano	2021
Renavam	1914569752	Tamanho	Médio

Below the fields is an orange button labeled 'ATUALIZAR'.

Fonte: elaborado pelos autores (2023).

Figura 32 - Confirmação de exclusão de Veículo

The screenshot shows a confirmation dialog box overlaid on a table of vehicles. The dialog has a white background, a large orange triangle with a white exclamation mark, and the text 'Deseja excluir este item?'. At the bottom are two black buttons: 'Confirmar' and 'Cancelar'.

Fiat	UIC2V33	2021	1914
Chevrolet			786
Hyundai			321
Renault			164
Fiat			331
Renault	BAB1F01	2008	906

Fonte: elaborado pelos autores (2023).

Figura 33 - Cadastramento de Veículo

Fonte: elaborado pelos autores (2023).

A próxima figura (Figura 34) apresenta a lista de todas as viagens cadastradas no sistema e um resumo das suas características e recursos. Pode-se notar o ponto de origem, o status, a data de saída e data de volta agendada se houver, o nome do motorista e o veículo que será utilizado, a lista com os destinos também é apresentada. Nota-se ainda a abaixo que a viagem não foi autorizada a acontecer, ao contrário da viagem apresentada na Figura 35 onde ela já foi autorizada e será de fato realizada. Um botão editar presente no final da viagem, faz referência à funcionalidade de edição de viagem que pode ser verificada na Figura 36.

Figura 34 - Tela de Viagens

Andamento	Status	Data Saída	Data Volta	Motorista	Veículo
NÃO ENFERMADO	NÃO CONFIRMADO	2023-07-04 00:13		Leila Hamilton	Fiorino

DESTINOS

DESTINO: Procuradoria Regional de Itajaí STATUS DO DESTINO: NÃO CONFIRMADO

Material	Quantidade	Sector de destino
Material: Extintor de incêndio	1	Sector de destino: Pesquisa e desenvolvimento
Material: Adaptador USB	3	Sector de destino: Pesquisa e desenvolvimento
Material: Máquina de desenho	2	Sector de destino: Gerência
Material: Gólete ATX	5	Sector de destino: Financeiro
Material: Switch de internet	2	Sector de destino: Estoque

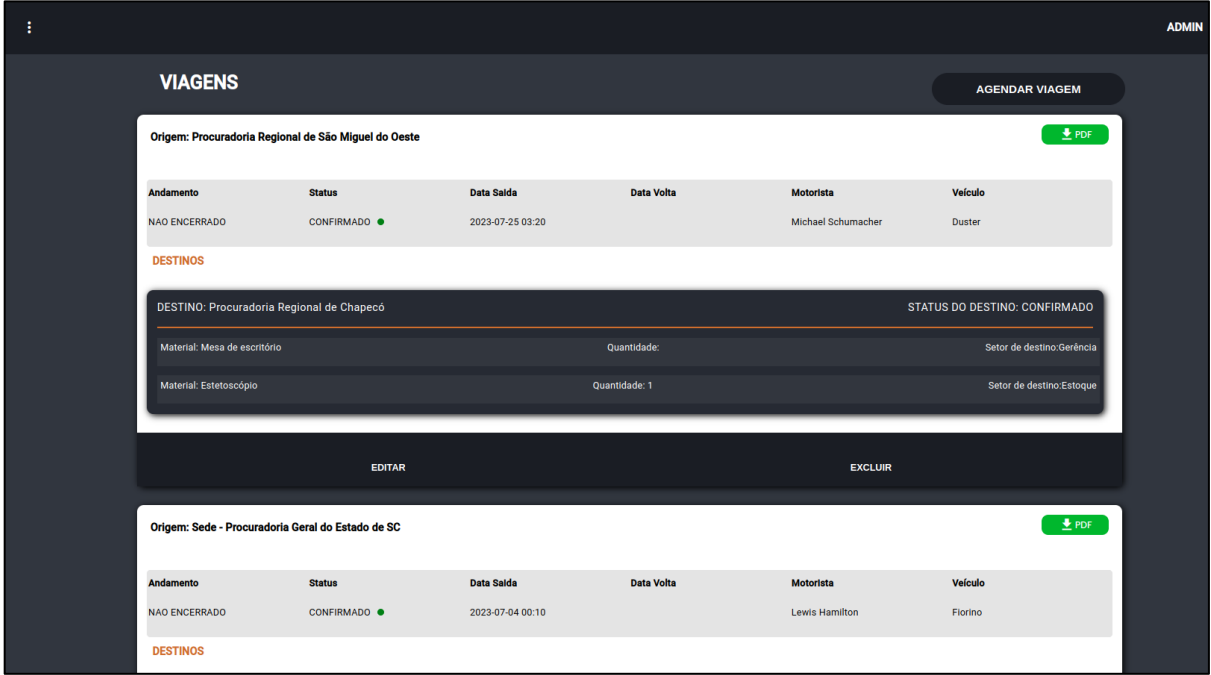
DESTINO: Procuradoria Regional de Joinville STATUS DO DESTINO: NÃO CONFIRMADO

Material	Quantidade	Sector de destino
Material: Máquina de documentos	5	Sector de destino: Financeiro
Material: Monitores 19 polegadas	3	Sector de destino: Financeiro
Material: Sapato caro	10	Sector de destino: RH

Fonte: elaborado pelos autores (2023).

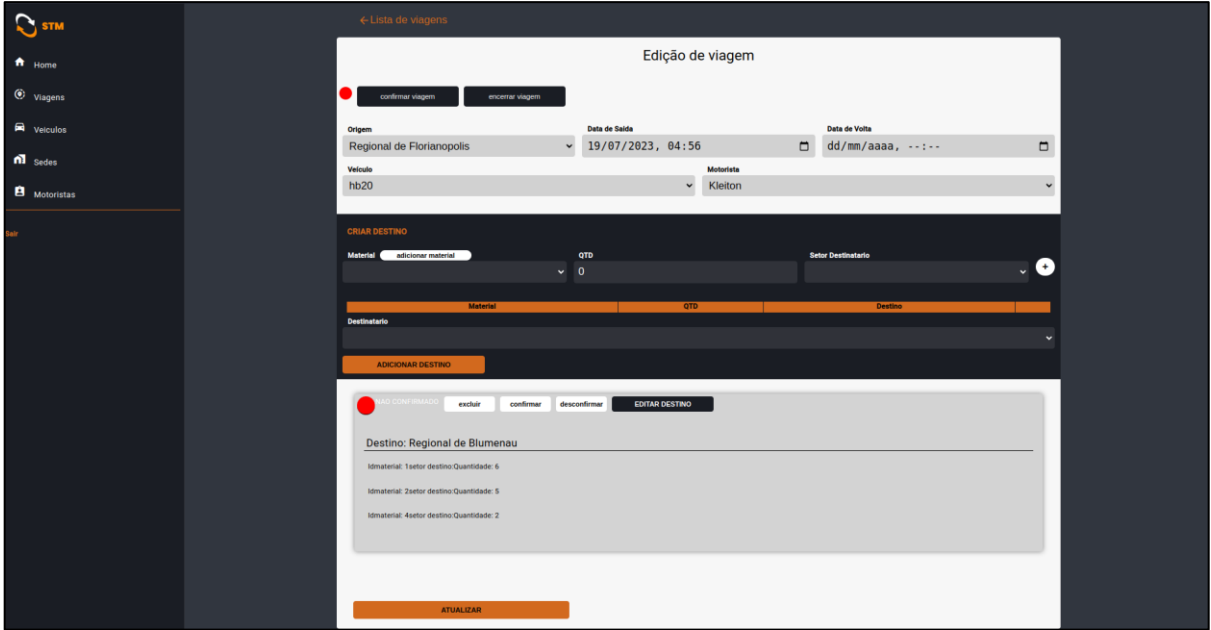
Em ambos os casos é possível notar um botão que diz PDF, este botão realiza o download do relatório da viagem em formato PDF com todas as informações mais relevantes da viagem como mostrado na Figura 37.

Figura 35 - Viagem Confirmada



Fonte: elaborado pelos autores (2023).

Figura 36 - Edição da Viagem



Fonte: elaborado pelos autores (2023).

Figura 37 – Relatório da Viagem (PDF)

Viagem ID: 7
 Saída:
 Motorista: Jim Clark
 CNH: A
 Email: jim_clark@pge-sc.gov.br
 Veículo: Hyundai HR
 Marca: Hyundai
 Placa: ABC-2223

Destino: Procuradoria Regional de Lages
 Status: NAO_CONFIRMADO

Material	Quant	Setor Destino	Ass. Recebedor
Teclado sem fio	3	Almoxarifado	
Teclado sem fio	2	Estoque	

Destino: Procuradoria Regional de Criciúma
 Status: CONFIRMADO

Material	Quant	Setor Destino	Ass. Recebedor
Sapato caro	10	RH	
Estetoscópio	2	Financeiro	
Extintor de incêndio	2	RH	
Cartela de ponto	2	RH	

Destino: Procuradoria Regional de Joinville
 Status: CONFIRMADO

Material	Quant	Setor Destino	Ass. Recebedor
Estetoscópio	1	Estoque	
Caneca MonsterHigh	4	Secretaria	

Fonte: elaborado pelos autores (2023).

A Figura 38 a seguir mostra o agendamento de uma nova viagem, para esse intento são obrigatórios os campos origem, data de saída, veículo e motorista. O campo destinado à introdução dos destinos e por consequência os campos de destinatário (sede destino), materiais, quantidade do material e setor não são obrigatórios e podem ser cadastrados posteriormente na tela de edição de viagem.

Figura 38 - Cadastro de Viagem

Fonte: elaborado pelos autores (2023).

A Figura 39 demonstra o recebimento de um e-mail de notificação. Para o envio de e-mail foi utilizado um servidor SMTP (Simple Mail Transfer Protocol) oferecido gratuitamente pela Google. A implementação foi feita com Jakarta Mail e para a geração do PDF a biblioteca ITextPDF foi usada.

Figura 39 - Notificação por E-mail

Fonte: elaborado pelos autores (2023).

4.4 RESULTADOS E TESTES

Os testes unitários desempenham um papel fundamental no desenvolvimento de software, pois asseguram a qualidade e confiabilidade do código. Eles têm como objetivo verificar minuciosamente o funcionamento de cada unidade individual do programa, isolando-a e testando-a de forma independente do restante do sistema, como ao identificar e corrigir erros de maneira ágil, garantindo que as alterações no código não resultem em regressões e

facilitando a refatoração. Além disso, eles servem como uma documentação viva do comportamento esperado das unidades de código, fomentando a colaboração entre os desenvolvedores e melhorando a manutenibilidade do software a longo prazo.

No momento, a cobertura de linhas de código é de 66% e todas as controllers foram abrangidas nos testes, como demonstrado na Figura 40.

Figura 40 - Cobertura de Testes Unitários

service												
service												
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
br.com.transportes.apitransportes.service		62%		51%	62	136	142	419	42	106	1	9
br.com.transportes.apitransportes.mapper		38%		22%	112	143	236	350	17	45	5	16
br.com.transportes.apitransportes.security		45%		0%	11	19	29	52	5	13	1	3
br.com.transportes.apitransportes.pdf		96%		n/a	0	17	4	76	0	17	0	1
br.com.transportes.apitransportes.entity		96%		100%	2	27	4	74	2	26	0	10
br.com.transportes.apitransportes.email		94%		n/a	0	6	5	47	0	6	0	2
br.com.transportes.apitransportes.controller		97%		n/a	1	41	2	87	1	41	0	8
br.com.transportes.apitransportes		37%		n/a	1	2	2	3	1	2	0	1
br.com.transportes.apitransportes.exception		100%		n/a	0	1	0	2	0	1	0	1
Total	1,439 of 4,286	66%	151 of 216	30%	189	392	424	1,110	68	257	7	51

Fonte: elaborado pelos autores (2023).

5 CONCLUSÃO

A partir do objetivo inicial traçado neste documento, o qual consiste no desenvolvimento de uma aplicação de gerenciamento de transferências de material, direcionada a empresas de médio ou grande porte com múltiplas sedes, visando aprimorar a comunicação entre os setores, é gratificante constatar que o propósito do sistema foi alcançado de forma satisfatória. Embora reconheçamos que, neste estágio inicial, o sistema ainda não seja uma versão completa para atender plenamente o mercado em larga escala, ele está trilhando o caminho adequado, considerando seu objetivo e o que foi desenvolvido até o momento.

O software foi concebido com base na ideia principal e ao longo do tempo, passou por alguns ajustes, porém, não se afastou do propósito original. Até o momento presente, o sistema está completamente alinhado com os objetivos estabelecidos no escopo inicial deste documento e nos requisitos do projeto. Assim, abre-se a possibilidade de futuras implementações no sistema, com o intuito de aprimorá-lo para os futuros usuários, oferecendo novas funcionalidades, atendendo a solicitações dos usuários e realizando correções necessárias.

É importante ressaltar que o sistema pode ser utilizado de duas maneiras: como uma solução completa, ou como um módulo de outro sistema, utilizando apenas nosso back-end como uma API.

Durante todo o desenvolvimento do projeto e da elaboração deste documento, houve um aprimoramento individual e coletivo do conhecimento de cada membro da equipe. Embora tenhamos enfrentado dificuldades ao longo do caminho, como o desafio de gerenciar o tempo dos membros da equipe para conciliar as atividades relacionadas a este projeto com suas responsabilidades diárias e trabalhos acadêmicos do semestre final, superamos esses obstáculos em equipe. Todos compartilhamos o mesmo objetivo comum de desenvolver o sistema STM.

6 TRABALHOS FUTUROS

O desenvolvimento deste projeto concentrou-se na criação de um sistema web para gerenciar a logística e comunicação das transferências de materiais entre as sedes de uma mesma empresa. Seu objetivo era proporcionar uma ferramenta eficiente tanto para os gerentes de transporte quanto para os demais setores das diferentes sedes.

O sistema em questão demonstra coerência em relação a seu propósito, no entanto, existem oportunidades de aprimoramento para atender às necessidades dos usuários de forma mais eficiente. Nesse sentido, foram selecionadas as sugestões externas e as ideias da equipe que mais se adequavam ao sistema, visando realizar novas implementações, adicionar funcionalidades e realizar correções no projeto.

Uma das principais sugestões foi a expansão do sistema para incluir uma versão mobile, direcionada aos motoristas. Essa extensão permitiria que os motoristas tivessem acesso prático a todos os dados relacionados às viagens que estão realizando, incluindo um mapa com os locais de entrega e rotas otimizadas para essas entregas.

Além disso, sugerimos alguns pontos interessantes para futuros trabalhos, com o intuito de aprimorar o sistema dentro da ideia principal:

- Desenvolvimento de um sistema mobile dedicado aos motoristas.
- Implementação de roteirização das entregas por GPS.
- Adição de uma opção de coleta de materiais.
- Inclusão de um chat para comunicação entre motoristas e gerentes de transporte.
- Geração automática de custos das viagens.
- Ampliação da cobertura de testes para garantir a qualidade e estabilidade do sistema.
- Essas propostas visam enriquecer o sistema, tornando-o ainda mais completo e capaz de atender às demandas dos usuários de maneira abrangente.

REFERÊNCIAS

ANGULAR. Angular - **What is Angular?**. Disponível em: <<https://angular.io/guide/what-is-angular>>. Acesso em: 5 mar. 2023.

BALLOU, Ronald H. **Gerenciamento da Cadeia de Suprimentos: Logística Empresarial**. Bookman editora, 2006.

BRASIL. Lei Nº 13.103, de 14 de mar de 2015. Dispõe sobre o exercício da profissão de motorista. Brasília, DF: Diário Oficial da União, 2015.

CABRAL FILHO, Djalma Alves. **Logística de transporte: fundamentos e importância**. Brazilian Journal of Development, v. 9, n. 4, p. 13567-13583, 2023.

CARVALHO, Marcos Roberto. **Gestão dos canais de distribuição**. Curitiba: IESDE Brasil SA, 2009.

DOCKER. **Docker overview | Docker Documentation**. Disponível em: <<https://docs.docker.com/get-started/overview/>>. Acesso em: 5 mar. 2023.

FOUNDATION, L. **OpenAPI Specification v3.1.0 | Introduction, Definitions, & More**. Disponível em: <<https://spec.openapis.org/oas/v3.1.0#openapi-specification>>. Acesso em: 26 abr. 2023.

FOUNDATION, M. **CSS: Cascading Style Sheets | MDN**. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/CSS#key_resources>. Acesso em: 6 mar. 2023.

FOUNDATION, M. **HTML: HyperText Markup Language | MDN**. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/HTML>>. Acesso em: 6 mar. 2023.

FOUNDATION, M. **JavaScript | MDN**. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/javascript>>. Acesso em: 6 mar. 2023.

GONÇALVES, Enyo José Tavares; CORTÉS, Mariela Inés. **Análise e Projeto de Sistemas**. História, v. 9, p. 3, 2015.

MICROSOFT. **TypeScript - JavaScript that scales**. Disponível em: <<https://www.typescriptlang.org/>>. Acesso em: 2 mar. 2023.

MILANI, André. **PostgreSQL-Guia do Programador**. Novatec Editora, 2008.

ORACLE. **O que é um banco de dados? | Oracle Brasil**. Disponível em: <<https://www.oracle.com/br/database/what-is-database/>>. Acesso em: 4 abr. 2023.

ORACLE. **Software Java | Oracle Brasil**. Disponível em: <<https://www.oracle.com/br/java/>>. Acesso em: 5 abr. 2023.

PAURA, Glávio Leal. **Fundamentos da logística**. 2016.

PUTRA, Andri Warda Pratama; BHAWIYUGA, Adhitya; DATA, Mahendra. **Implementasi Autentikasi JSON Web Token (JWT) Sebagai Mekanisme Autentikasi Protokol MQTT Pada Perangkat NodeMCU**. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, v. 2, n. 2, p. 584-593, 2018.

SOFTWARE FOUNDATION, A. **Maven – Introduction**. Disponível em: <<https://maven.apache.org/what-is-maven.html>>. Acesso em: 7 abr. 2023.

SOMMERVILLE, Ian. **Engenharia de Software/Ian Sommerville**. Tradução Ivan Bosnic e karlinka G. de O. Gonçalves. revisão técnica Kechi Hirama. — 9. ed. — São Paulo : Pearson Prentice Hall, 2011.

SOMMERVILLE, Ian. **Sommerville: Software Engineering**. Monthly Notices of, 2011.

VMWARE. **Spring Framework**. Disponível em: <<https://spring.io/projects/spring-framework>>. Acesso em: 5 abr. 2023.

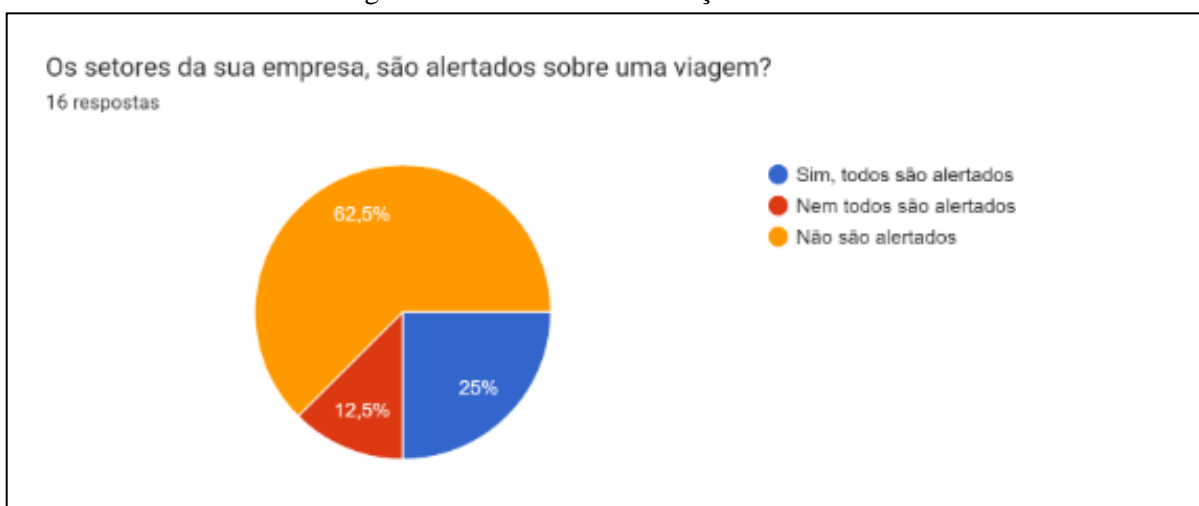
WATERS, Donald. **Logistics An Introduction to supply chain management**. Palgrave macmillan, 2021.

APÊNDICE A – Resultados das pesquisas

As valiosas contribuições dos colaboradores nesta pesquisa revelaram um denominador comum ao serem indagados sobre as possíveis razões subjacentes às viagens desnecessárias em seus ambientes de trabalho. A maioria deles expressou dificuldades no planejamento das viagens, apontando a escassez de comunicação interdepartamental como um fator limitante. Alguns participantes mencionaram a ausência de um sistema centralizado que lhes permitisse saber se havia viagens programadas para os destinos que necessitavam, enquanto outro entrevistado admitiu a falta de conhecimento quanto à realização ou não de determinada viagem.

Em outro aspecto investigado pelos colaboradores, conforme evidenciado na Figura 1Figura 41 questionou-se se todos os setores eram devidamente notificados sobre as viagens. O resultado revelou que 62,5% dos respondentes afirmaram que não recebiam qualquer tipo de alerta, enquanto 25% declararam ser informados e 12,5% admitiram que nem todos os setores eram devidamente alertados. É importante ressaltar que quando os setores não são devidamente informados sobre as viagens, a maximização do potencial dessas se torna impraticável, culminando na necessidade de realizar novas viagens para entregar materiais aos setores que foram negligenciados, resultando em despesas adicionais e retrabalho.

Figura 41 - Gráfico da notificação aos setores



Fonte: elaboração dos autores (2023)