



**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMA**

**GABRIEL DE SOUZA ADÃO
GABRIEL RIBEIRO GUEDES**

**DESENVOLVIMENTO DE UM TMS (TRANSPORTATION MANAGEMENT
SYSTEM) COM OBJETIVO EDUCATIVO**

Guarulhos

2024

GABRIEL DE SOUZA ADÃO
GABRIEL RIBEIRO GUEDES

DESENVOLVIMENTO DE UM TMS COM OBJETIVO EDUCATIVO

Trabalho de Conclusão de Curso apresentado ao Curso de Análise e Desenvolvimento de Sistemas como requisito parcial para obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador (a): Profa. Esp. Jane Maria dos Santos Eberson.

Guarulhos

2024
GABRIEL DE SOUZA ADÃO
GABRIEL RIBEIRO GUEDES

DESENVOLVIMENTO DE UM TMS COM OBJETIVO EDUCATIVO

Trabalho de Graduação apresentado ao Curso de Análise e Desenvolvimento de Sistemas como requisito parcial para obtenção do **Título de Tecnólogo em Análise e Desenvolvimento de Sistemas**.

Banca Examinadora

Orientador: _____

Profa. Esp. Jane Maria dos Santos Eberson
Fatec Guarulhos

Banca: _____

Titulação + nome completo
IES de origem

Banca: _____

Titulação + nome completo
IES de origem

Guarulhos, 18/06/2024

AGRADECIMENTOS

Gostaríamos de expressar nossa profunda gratidão a todos aqueles que contribuíram para a realização deste Trabalho de Conclusão de Curso.

Aos professores, nosso mais sincero agradecimento. Suas orientações, paciência e dedicação foram fundamentais para a construção deste trabalho. Em especial, aos nossos orientadores, agradecemos pela orientação precisa, pelos ensinamentos compartilhados e pelo incentivo constante.

Às nossas famílias, nosso porto seguro, agradecemos por todo o suporte emocional durante esta jornada. Em especial, aos nossos pais, que sempre acreditaram em nós e nos incentivaram a nunca desistir dos nossos sonhos.

Aos nossos amigos da faculdade, que compartilharam conosco os desafios e as conquistas ao longo dessa trajetória acadêmica. Agradecemos a companhia, pelas discussões enriquecedoras e pelos momentos de descontração que tornaram esta experiência ainda mais especial.

A todos vocês, nosso sincero agradecimento. Este trabalho é um reflexo do apoio e incentivo de cada um de vocês.

RESUMO

ADÃO, Gabriel de Souza; RIBEIRO, Gabriel Guedes. Uma Proposta de TMS com fins didáticos para a Fatec Guarulhos. 2023. 85p. Trabalho de Conclusão de Curso – Faculdade de Tecnologia de Guarulhos, Guarulhos.

Este Trabalho de Conclusão de Curso desenvolveu um Sistema de Gerenciamento de Transporte (TMS) para fins educacionais, otimizando a gestão de transporte e logística. Utilizando GraphQL, o sistema oferece funcionalidades eficientes para gestão de pedidos, controle de veículos e monitoramento de entregas, com uma interface intuitiva. A arquitetura modular e escalável facilita a manutenção e futuras expansões, com base em diversas referências bibliográficas.

Palavras-chave: Transporte. TMS. Educação. Logística. Software. Gestão de Transporte.

ABSTRACT

This Final Course Project developed a Transportation Management System (TMS) for educational purposes, aiming to optimize transportation and logistics management. Using GraphQL, the system offers efficient functionalities for order management, vehicle control, and delivery monitoring, with an intuitive interface. The modular and scalable architecture facilitates maintenance and future expansions, based on numerous bibliographical references.

Keywords: Transportation. TMS. Education. Logistics. Transportation Management.

LISTA DE ABREVIATURAS

TMS.	Transportation Management System
DB.	Data Base
PIB.	Produto Interno Bruto
DDD.	Domain Driven Design
SRP.	Single Responsibility Principle
OCP.	Open-Closed Principle
LSP.	Liskov Substitution Principle
ISP.	Interface Segregation Principle
DIP.	Dependency Inversion Principle
JS.	JavaScript
TS.	TypeScript
API.	Application Programming Interface
REST.	Representational State Transfer
GRAPHQL.	Graph Query Language
DER.	Diagrama de Entidade e Relacionamento
PWA.	Progressive Web App

LISTA DE QUADROS

Quadro 1 - Diferenças entre <i>Mysql</i> e <i>PostgreSQL</i>	28
Quadro 2 – Requisitos Funcionais	30
Quadro 3 - Requisitos Não Funcionais	31

LISTA DE FIGURAS

Figura 1 - Representatividade dos custos Logísticos em relação ao PIB	15
Figura 2 - Tipos de sistemas de informação	17
Figura 3 – Exemplos de Endpoints de uma API REST	24
Figura 4 – GraphQL – query para procurar todos os emails e nomes de usuários	25
Figura 5 – GraphQL: query para retorna apenas os nomes dos usuários	25
Figura 6 – GraphQL: query para buscar os nomes dos usuários que contém a letra T	26
Figura 7 - Caso de uso do TMS	32
Figura 8 - Diagrama de sequência de pedido de frete	33
Figura 9 - Diagrama de sequência do gerenciamento de manutenção	34
Figura 10- Diagrama de sequência do gerenciamento de motorista próprio ..	34
Figura 11- Diagrama de sequência do processamento de pedido	35
Figura 12- Diagrama de sequência do gerenciamento de clientes	35
Figura 13 - Estrutura Geral	36
Figura 14 – Estrutura da Tabela de Usuários (parte 1)	37
Figura 15 - Estrutura Tabela de Usuário (parte 2)	38
Figura 16 – Estrutura da tabela de carroceria de veículos	39
Figura 17 – Estrutura da tabela de tipo de veículos	40
Figura 18 – Estrutura da tabela de marcas de veículos	40
Figura 19 - Estrutura da tabela de modelo de veículo	41
Figura 20 - Estrutura da tabela de veículos	42
Figura 21 – Estrutura da tabela de motorista próprios	43
Figura 22 – Estrutura da tabela de pessoa física	44
Figura 23 - Estrutura da tabela de pessoa jurídica	45
Figura 24 - Estrutura da tabela de veículos próprio	46
Figura 25 - Estrutura da tabela de transportadoras	47

Figura 26 - Estrutura da tabela de cliente físico	48
Figura 27 - Estrutura da tabela de Cliente Jurídico	48
Figura 28 - Estrutura de dados da tabela de remetente	49
Figura 29 – Estrutura de dados da tabela de destinatário	50
Figura 30 - Estrutura da Tabela de ICMS	50
Figura 31 - Estrutura da tabela de cotação de clientes jurídicos (parte 1)	51
Figura 32 - Estrutura da tabela de cotação de clientes jurídicos (parte 2)	52
Figura 33 - Estrutura de dados da Tabela de Cotação de clientes Físicos (Parte 1)	53
Figura 34 - Estrutura de dados da Tabela de Cotação de clientes Físicos (Parte 2)	53
Figura 35 - Estrutura de dados da tabela de contratos de frete para clientes jurídicos	54
Figura 36 - Estrutura da Tabela Gasto de Frete	55
Figura 37 - Estrutura da Tabela de Pedidos de clientes jurídicos	55
Figura 38 - Estrutura da Tabela de Pedidos para Clientes Físicos	57
Figura 39 - Estrutura da Tabela de Ocorrências	58
Figura 40 - Estrutura da Tabela de Processamento de Pedido	59
Figura 41 - Data Table	60
Figura 42 - Exemplo de Combo Box	60
Figura 43 - Componente de criação de pessoa jurídica(empresa)	61
Figura 44 - Componente de Criação de Pessoa Física	61
Figura 45 - Componente de Interface - Menu Lateral	61
Figura 46 - Interface <i>Light mode</i>	62
Figura 47 - Interface <i>Dark Mode</i>	63
Figura 48 - Instalação da versão <i>desktop</i> do TMS	64
Figura 49 - Instalação da versão <i>mobile</i>	64
Figura 50 - Tela de Permissão de Instalação da versão mobile	65
Figura 51 - Mostrando aplicativo instalado no celular	66
Figura 52 - Tela de login da versão mobile	66
Figura 53 - Exemplo de tela da versão mobile	67
Figura 54 - Tela de Login	68
Figura 55 - Tela de <i>Dash Board</i>	68

Figura 56 - Tela de Controle de Usuários	69
Figura 57 - Tela de Criação/Atualização de usuários	69
Figura 58 - Tela de Criação/Atualização de Transportadora	70
Figura 59 - Exemplo de tela da versão mobile	70
Figura 60 - Tela de Criação/Atualização de marca de veículo	71
Figura 61 - Tela de Criação/Atualização de Carroceria	71
Figura 62 - Tela de Criação/Atualização de Tipo de veículo	71
Figura 63 - Tela de Criação/Atualização de modelo de veículo	72
Figura 64 - Tela de Criação/Atualização de Veículo	72
Figura 65 - Cadastro de destinatário com CPF já criado no sistema	73
Figura 66 - Criação de destinatário com CNPJ já cadastrado sistema	73
Figura 67 - Criação de destinatário com CNPJ já cadastrado no sistema	74
Figura 68 - Criação de empresa na tela de destinatário	74
Figura 69 - Tela de Criação/Atualização de Jurídico	75
Figura 70 - Tela de Criação/Atualização de cliente Jurídico	75
Figura 71 - Tela de Cadastro/Atualização de ICMS	76
Figura 72 - Tela de Criação/Atualização de Despesa de Frete	76
Figura 73 - Tela de Cadastro/Atualização de incidentes de frete (Sinistro)	76
Figura 74 - Tela de Cadastro/Atualização de cotação – Dados da cotação ...	77
Figura 75 - Tela de Cadastro/Atualização de Cotação – Dados da origem	77
Figura 76 - Tela de Cadastro/Atualização de Cotação – Dados de Destino ...	77
Figura 77 - Tela de Cadastro/Atualização de Pedido de Cliente Fisco	78
Figura 78 - Tela de Criação/Atualização de Pedido de Cliente Jurídico	78
Figura 79 - Tela de Criação/Atualização de Contrato de Cliente jurídico	79
Figura 80 - Tela de Criação/Atualização de Manutenção	79
Figura 81 - Tela de Criação/Atualização de empresa de manutenção	80
Figura 82 - Tela de cadastro/atualização processamento de pedido	80
Figura 83 - Tela de menu de opções do pedido	81
Figura 84 - Modal geração de CTE	81
Figura 85 - CTE gerado	82

SUMÁRIO

1 INTRODUÇÃO	13
2 FUNDAMENTAÇÃO TEÓRICA.....	14
2.1 Transportation Management System (TMS) e sua importância.....	14
2.2 Por que aprender a utilizar um TMS.....	15
2.3 TMS como um Sistema de informação.....	16
3. METODOLOGIA	18
3.1 Coleta de Material Bibliográfico	18
3.2 Seleção e Análise de Fontes	18
3.3 Organização e Estruturação da Revisão Bibliográfica.....	19
3.4 Síntese e Discussão dos Resultados	19
3.5 Conclusão da Pesquisa Bibliográfica	19
3.6 Limitações da Metodologia	20
4 DESENVOLVIMENTO	20
4.1 Designer do Software	20
4.2 Arquitetura do Software	23
4.3 Stack de desenvolvimento	26
4.4 Banco de dados: PostgreSQL	28
4.5 Descrição do Software	29
4.6 Escopo do Software	29
4.7 REQUISITOS DO SOFTWARE	30
4.7.1 Requisitos Funcionais	30
4.7.2 Requisitos Não Funcionais	31
4.8 Especificação do Software	32
4.8.1 Caso de uso	32
4.8.2 Diagrama de Sequência	32

4.9 diagrama de entidade relacionamento (der)	36
4.9.1 DANCO DE DADOS	36
4.9.2 Estrutura de Dados	37
4.9.3 Tabela relacionadas com Veículos	39
4.9.4 Tabelas relacionadas com Pessoas Físicas	43
4.9.5 Tabela de Pessoa Jurídica	44
4.9.6 Tabela de Motorista Próprio	45
4.9.7 Tabela de Transportadora	46
4.9.8 Tabelas de clientes	47
4.9.9 Tabela de Destinatário e Remetente	48
4.9.10 Tabela de ICMS	50
4.9.11 Tabelas de Cotação	51
4.9.12 Contrato de Frete para Clientes Jurídicos	54
4.9.11 Tabela de Gastos de frete	54
4.9.11 Tabelas de Pedidos	55
4.9.12 Tabela de Incidentes	57
4.9.13 Tabela de Processamento de Pedido	58
5 RESULTADOS E DISCUSSÕES	59
5.1 componentes de interface	59
5.2 Interfaces graficas	62
6 CONSIDERAÇÕES FINAIS	82
REFERÊNCIAS	84

1 INTRODUÇÃO

O processo de aprendizagem pode ser complicado e difuso para muitos, especialmente quando precisa-se conectar o conhecimento teórico à sua aplicação prática. Em um ambiente de sala de aula, muitas vezes é apresentado conceitos que parecem distantes ou abstratos, o que pode dificultar nossa compreensão do que o professor está tentando transmitir.

Essa noção de entendimento só piora em casos onde só pode-se entender na prática, como no caso de explicações de softwares, visto que muitas vezes os alunos não têm familiaridade ou se quer conhecem sobre o software e por isso não conseguem de fato aprender o que é e como é o seu funcionamento, por este motivo o objetivo do projeto é desenvolver um modelo de TMS voltado para formação de alunos, a fim de auxiliar os professores a passar o conhecimento sobre o software de forma mais clara e prática.

Um Sistema de Gestão de Transporte (TMS), conhecido em inglês como *Transportation Management System*, é um software desenvolvido para auxiliar empresas em uma de suas atividades fundamentais: o gerenciamento do transporte de cargas. Embora o transporte de mercadorias ou matérias-primas possa parecer simples à primeira vista, em empresas com redes de distribuição consideráveis, a realidade é bem diferente. Isso ocorre porque o manuseio e transporte de cargas representam aproximadamente um terço dos custos logísticos das organizações (CAXITO, 2011), essa atividade envolve variáveis complexas e requer um planejamento robusto, assim, o objetivo do TMS é fornecer ferramentas que tornem as empresas competitivas no gerenciamento de frota.

Contudo, o software TMS a ser desenvolvido não possui um propósito comercial. Trata-se, principalmente, de uma ferramenta de estudo destinada aos estudantes de Logística da FATEC Guarulhos. Portanto, embora deva atender aos principais requisitos de um sistema desse tipo, ele será útil no processo de aprendizagem dos alunos permitindo a experimentação em um sistema comum no mercado de trabalho da área agregando mais conhecimento prático e enriquecendo a experiência do aluno.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 *TRANSPORTATION MANAGEMENT SYSTEM (TMS) E SUA IMPORTÂNCIA.*

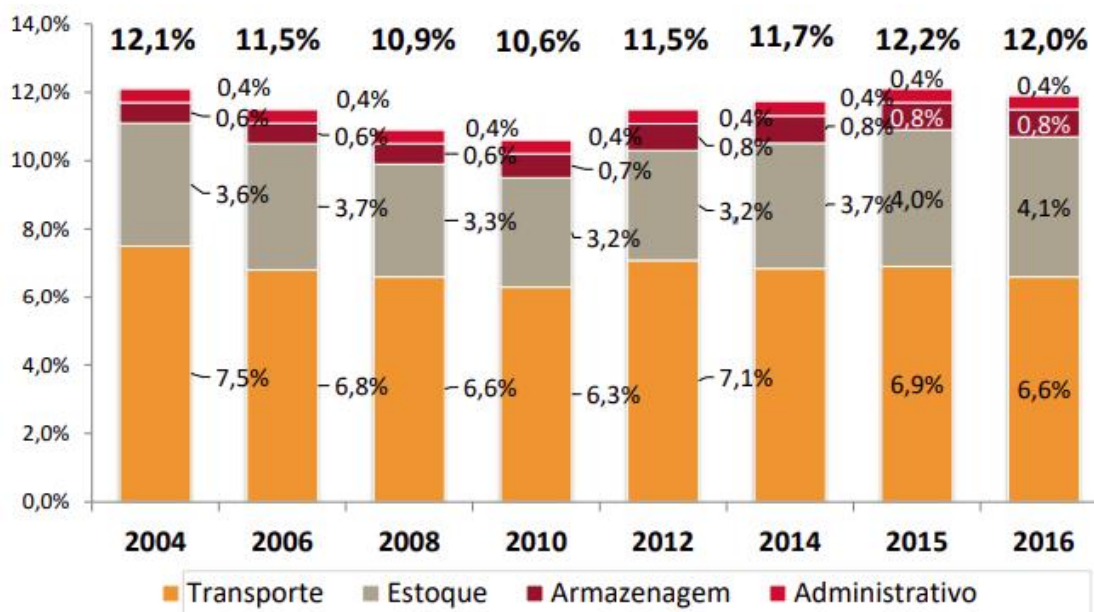
De forma simplificada e mais abrangente o TMS é um software, que assim como o nome sugere, é responsável por gerenciar todo o sistema de transporte de uma empresa, mais do que isso ele é responsável por permitir que grandes empresas que lidam com o transporte de carga, consigam lidar com grandes volumes de produtos, entregas, coletas, etc. visto que além de melhorar a qualidade e produtividade de todo o processo de distribuição ele também pode ser um roteirizador que planeja viagem utilizando localização, distância e trajeto entre destinos para realizar todos os cálculos envolvendo uma entrega (DIAS, 2012).

Alguns outros benefícios e facilidades citados por DIAS (2012) São:

- quanto à expedição: rastreabilidade das informações e a emissão do romaneio de carga com as notas fiscais expedidas e consulta para a montagem dos embarques;
- programação dos transportes: controle dos transportes contratados pelo embarcador. Conferência dos valores de frete entre o negociado e o valor cobrado pela transportadora;
- registro de ocorrências: controle e monitoramento das situações de roubo, avarias, reentregar, devolução, durante o trajeto de transporte;
- controle das tabelas de frete entre o embarcador, os transportadores ou agregados, contratados para o transporte;
- controle dos boletos emitidos para pagamento.
- redução nos custos de transportes e melhoria de serviço;
- melhor utilização dos serviços de transportes;
- consolidação de cargas e definição de rotas; • redução do tempo para programar a distribuição e preparação de embarques;
- informação dos custos de transporte por cliente e por produto;
- análise da evolução dos custos com transportes;
- disponibilidade de informações on-line;
- indicadores de desempenho para aferir a gestão de transportes.

Com essas ideias pode-se observar como um TMS se faz essencial em qualquer empresa que lide com cargas, ainda mais no nosso país onde mais de 10% do PIB (Produto interno Bruto) é gasto nos custos Logísticos e principalmente no transporte como demonstrado na Figura 1 a seguir:

Figura 1 - Representatividade dos custos Logísticos em relação ao PIB



Fonte:

Ilos (2017)

2.2 POR QUE APRENDER A UTILIZAR UM TMS.

No cenário empresarial atual, marcado pela crescente competitividade e dinamismo, as organizações se veem constantemente pressionadas a desenvolver soluções inovadoras que lhes permitam tomar decisões mais assertivas e eficazes. Como afirmado por Carvalho (2006), a sobrevivência e o sucesso das empresas dependem, em grande parte, de sua capacidade de adaptação e inovação em um ambiente de negócios em constante evolução. Nesse contexto, a adoção e o domínio de tecnologias avançadas se tornaram imperativos para a competitividade empresarial.

Um exemplo notável dessa necessidade de adaptação e inovação ocorre no setor de comércio eletrônico, onde as empresas buscam incansavelmente aprimorar suas operações logísticas. Um caso ilustrativo é o do Mercado Livre, uma das principais plataformas de comércio eletrônico da América Latina. A empresa

reconheceu a importância estratégica da logística em seu modelo de negócios e, como destacado por Lóssio (2016), realizou a aquisição da Axado, uma empresa especializada em tecnologia logística. Essa aquisição tinha o objetivo de maximizar a participação de lojas oficiais externas na plataforma do Mercado Livre, ampliar o acesso da empresa a uma vasta rede de operadores logísticos no país e aprimorar o seu sistema de gerenciamento de transporte (TMS).

O exemplo do Mercado Livre demonstra claramente como as empresas líderes reconhecem a importância estratégica do TMS na gestão logística. O TMS não é mais apenas uma ferramenta auxiliar, mas sim uma peça fundamental para garantir a eficiência e a eficácia das operações de transporte em um ambiente de negócios altamente competitivo.

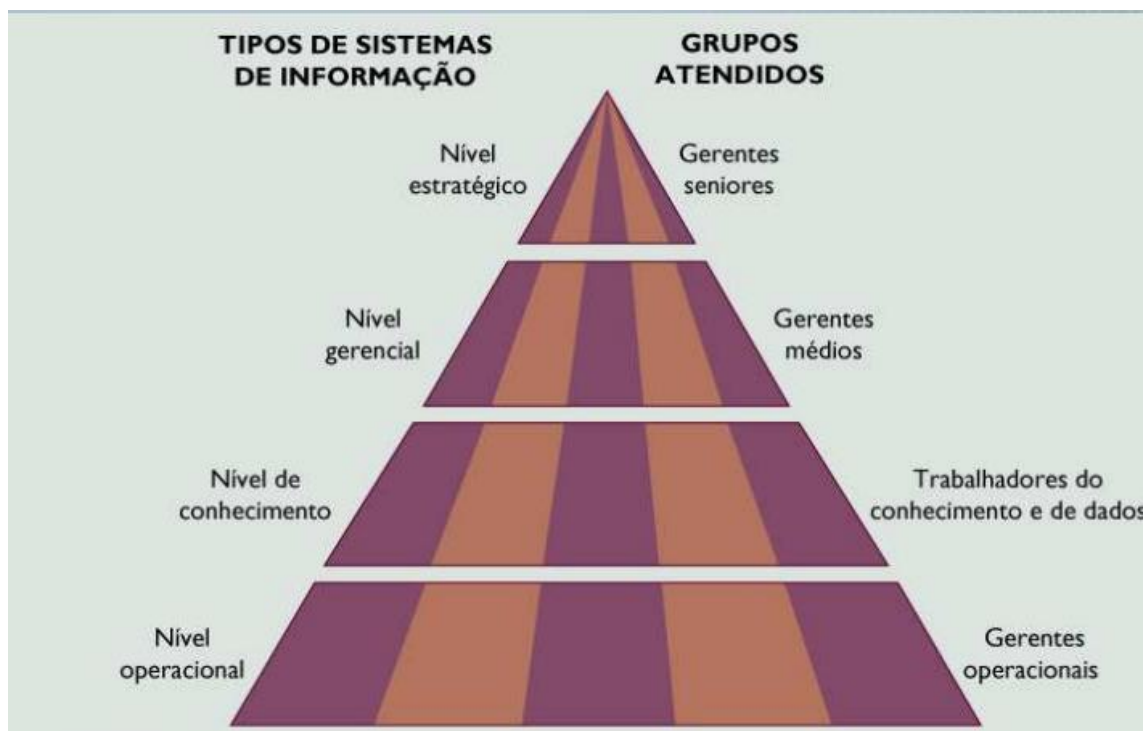
Além disso, o TMS desempenha um papel crucial na capacidade das empresas de se adaptarem às demandas dos consumidores modernos, que esperam entregas rápidas e precisas. O gerenciamento eficaz do transporte não só reduz os custos operacionais, mas também melhora a experiência do cliente, aumentando a satisfação e a fidelidade.

Portanto, aprender a utilizar um TMS não é apenas uma vantagem competitiva para os alunos, mas uma necessidade para se manter relevantes e bem-sucedidos em um mercado de trabalho mais acirrado e competitivo.

2.3 TMS COMO UM SISTEMA DE INFORMAÇÃO.

Um Sistema de informação pode ser definido como vários softwares que se comunicam, coletam/recuperam e armazenam dados para então processá-los e transformá-los em informações que apoiam as decisões tomadas, além de servir como forma de coordenação e controle de uma organização. (LAUDON; LAUDON, 2007), sendo que os sistemas dentro de uma empresa podem ser caracterizados em alguns níveis como demonstrado na Figura 2 a seguir:

Figura 2 - Tipos de sistemas de informação



Fonte: Lima (2021)

Observando a imagem vemos que um TMS pode se encaixar em diversos níveis já que se trata de um software de gerência de frota e frete e gera relatórios sobre o funcionamento logístico de uma organização. Para melhor entender o porquê dele se encaixar em diversos níveis, podemos observá-lo de duas formas diferentes, sendo elas:

- Visão Funcional: nesse tipo de visão o TMS é visto como parte de um sistema de manufatura e produção, ou seja, ele é responsável por lidar com os bens e serviços de uma empresa quando se trata da logística dos mesmos.
- Visão de grupos de usuários: já neste tipo de visão a perspectiva passa a enxergar todos os sistemas em relação aos vários níveis de gerência existentes (a Figura 2 acima demonstra esses níveis) e tipos de decisão que eles apoiam (LAUDON; LAUDON, 2007), fazendo assim o TMS um software fundamental na camada Gerencial de uma empresa por permitir todo o controle da frota logística, destacando-se principalmente na camada operacional por permitir o controle e monitoramentos dos veículos e entregas/coletas de uma organização, além de fornecer *insights* sobre a frota e fretes dentro da organização.

Em um mundo empresarial impulsionado pela inovação e eficiência, o aprendizado e domínio do TMS não são apenas benéficos, são essenciais. O TMS não apenas melhora a eficiência operacional, mas também aprimora a experiência do cliente, resultando em satisfação e fidelidade. Ao posicionar o TMS em diferentes perspectivas dentro da estrutura de sistemas de informação, percebemos sua grande importância e seu impacto abrangente na operação e estratégia organizacional. Portanto, investir no entendimento e aplicação eficaz do TMS é uma estratégia sábia para qualquer empresa que busque prosperar em um cenário competitivo.

3. METODOLOGIA

A metodologia empregada neste trabalho se baseia principalmente na pesquisa bibliográfica, que consiste na análise e revisão crítica de literatura relevante relacionada ao Transportation Management System (TMS) e sua importância no contexto da logística empresarial. A seguir, apresentam-se os passos e abordagens adotados para conduzir esta pesquisa:

3.1 COLETA DE MATERIAL BIBLIOGRÁFICO

Para iniciar a pesquisa, foi realizada uma extensa busca por literatura relacionada ao TMS e sua aplicação na gestão logística. Utilizamos uma variedade de fontes de pesquisa, incluindo livros acadêmicos, artigos científicos, relatórios técnicos, documentos governamentais e recursos online confiáveis. A busca abrangeu bases de dados acadêmicas, como Google Scholar, Scopus, etc., bem como bibliotecas digitais de universidades e sites de instituições relacionadas à logística e tecnologia de transporte (como a própria FATEC).

3.2 SELEÇÃO E ANÁLISE DE FONTES

Uma vez coletado o material bibliográfico, procedemos com a seleção das fontes mais relevantes e adequadas aos objetivos deste trabalho. Priorizamos publicações que abordavam os seguintes tópicos:

- Conceitos e definições relacionados ao TMS.
- Importância e benefícios do TMS na gestão logística.
- Aplicações práticas do TMS em empresas de logística e transporte.

Para cada fonte selecionada, apresenta uma análise crítica e uma síntese das informações pertinentes, destacando os principais conceitos, argumentos e descobertas relevantes para este estudo.

3.3 ORGANIZAÇÃO E ESTRUTURAÇÃO DA REVISÃO BIBLIOGRÁFICA

Com as informações obtidas por meio da pesquisa bibliográfica, ordenando a revisão de forma a seguir uma estrutura lógica. Destacam-se os principais tópicos da revisão de acordo com os aspectos relevantes do TMS, incluindo: O que é um Transportation Management System (TMS) e suas funcionalidades; A importância do TMS na logística empresarial e qual a necessidade de se aprender a utilizar um TMS.

Cada tópico foi abordado com base nas fontes bibliográficas selecionadas, e suas informações foram integradas de forma a fornecer uma visão completa e coerente sobre o assunto.

3.4 SÍNTESE E DISCUSSÃO DOS RESULTADOS

Após a organização da revisão bibliográfica, procedemos à síntese e discussão dos resultados obtidos. Nesta etapa, destacam-se os principais pontos de acordo com a relevância e a contribuição para o entendimento do TMS e sua aplicação na logística empresarial. Também comparamos diferentes perspectivas e abordagens encontradas na literatura.

3.5 CONCLUSÃO DA PESQUISA BIBLIOGRÁFICA

Assim a pesquisa bibliográfica com uma conclusão que resume os principais *insights* e descobertas obtidas a partir da revisão da literatura. Além disso, essa conclusão servirá como base para a próxima fase deste trabalho, que envolverá o desenvolvimento do modelo de TMS voltado para a educação de alunos.

3.6 LIMITAÇÕES DA METODOLOGIA

É importante reconhecer que a pesquisa bibliográfica pode apresentar algumas limitações, como a possibilidade de não abranger todas as fontes relevantes disponíveis. No entanto, fizemos esforços para abranger uma ampla gama de materiais e fontes confiáveis a fim de mitigar essas limitações.

A metodologia de pesquisa bibliográfica adotada neste trabalho permitiu a obtenção de uma base sólida de conhecimento teórico sobre o TMS e sua importância na logística empresarial. Essa base será fundamental para o desenvolvimento do modelo de TMS educacional planejado para atender aos objetivos deste projeto.

4 DESENVOLVIMENTO

Neste capítulo apresentam-se o processo do projeto de software, ferramentas e tecnologias utilizadas no desenvolvimento do TMS, descrevendo nossas escolhas de linguagem de programação, arquitetura de software, requisitos funcionais e não funcionais entre outros.

4.1 DESIGNER DO SOFTWARE

Para o desenvolvimento do designer do software optamos por seguir um modelo que se concentre na modelagem do software e então optamos por um método conhecido como DDD (*Domain Driven Designer*), cujo seu ideal é colocar o domínio do software, isto é, o problema que o software quer resolver, através do entendimento das regras de negócio, entidades e processos envolvidos com o problema em si, para que o software represente de forma real e concreta a realidade do problema.

1° Camada de Domínio (*Domain*): Essa camada é responsável por conter as regras de negócio e as lógicas essenciais para paridade entre o software e a realidade, como escrito por Evans (2003) indica o domínio como o conjunto de regras, conceitos, entidades e processos que definem a atividade principal de uma organização ou sistema.

2° Camada de Aplicação: É uma camada fina que coordena a atividade da aplicação. Ela não contém lógica de negócios, mas gerencia o fluxo de controle e a interação entre a interface do usuário, a camada de domínio e a infraestrutura. É nesta camada que as regras de negócio são colocadas em execução e repassada as partes do software que necessita delas, e é nela também que é criada uma interface entre a camada de domínio e a camada de infraestrutura. Sendo assim ela serve como uma forma de separar as responsabilidades entre as camadas e manter a coesão do sistema, facilitando o processo manutenção e evolução do software.

3° Camada de Infraestrutura: Por fim a camada de infraestrutura é responsável por fornece recursos e serviços que são essenciais para o funcionamento da aplicação, mas que não estão ligados diretamente com as regras de negócio como serviços externos (como se conectar a API¹ de um terceiro), conter os componentes de suporte (*frameworks*², ferramentas etc.) e segurança (autentificação, criptografia e etc.)

Entendido o funcionamento do DDD optamos por implementar outros conceitos e padrões que são baseados no SOLID para que os ideais do DDD sejam alcançados.

SOLID são princípios de desenvolvimento de software criado por Robert C. Martin (também conhecido como Uncle Bob) que é um grande desenvolvedor de software que influenciou e escreveu livros como *Clean Code*, que visam mostrar conceitos para os programadores desenvolverem códigos de fácil manutenção e leitura, o significado de cada letra do SOLID:

-S representa o SRP - Single Responsibility Principle (Princípio de responsabilidade Única), que como indicado por C. Martin et al. (2011, pág. 138):

“O Princípio da Responsabilidade Única (SRP, sigla em inglês para *Single Responsibility Principle*) afirma que uma classe ou módulo deve ter um, e apenas um, motivo para mudar. Este princípio nos dá uma definição de responsabilidade e uma orientação para o tamanho da classe. Estas devem ter apenas uma responsabilidade e um motivo para mudar”.

¹ API é um conjunto de regras, protocolos e ferramentas que permite a comunicação e interação entre diferentes softwares ou componentes de software.

² Um framework é uma estrutura ou conjunto de bibliotecas, ferramentas e padrões de desenvolvimento de software que fornecem uma base para a criação e o desenvolvimento de aplicativos.

Ou seja, uma classe não pode fazer várias coisas e deve ser responsável por realizar apenas uma tarefa.

-O representa OCP - *Open Closed Principle* (Princípio Aberto Fechado):

“As classes devem ser abertas para expansão, mas fechadas para alteração” (C. Martin et al., 2011, pág. 149). Isso quer dizer que após a criação de uma classe que já funciona não há motivos para modificá-la diretamente pois isso pode causar erros e alterações indesejáveis no sistema. O ideal é estender essa classe para então adicionar novos comportamentos a classe estendida.

-L: LSP - *Liskov Substitution Principle* (Princípio da substituição de Liskov):

“... este princípio diz que, para criar sistemas de software a partir de partes intercambiáveis, essas partes devem aderir a um contrato que permita que elas sejam substituídas umas pelas outras”. (MARTIN 2000, pág. 70). Isso quer dizer que uma classe x que recebe uma classe base também pode receber as classes que derivam a classe base, garantindo que o sistema possa mudar, mas sem precisar refatorar o código diretamente, permitindo uma flexibilidade e a reutilização de código.

-I: ISP - *Interface Segregation Principle* (Princípio da segregação de interfaces):

Este princípio defende que uma classe não pode ser forçada a implementar interfaces e métodos que não vai utilizar, o que significa que devemos criar interfaces específicas para o contexto ao invés de criar interfaces genéricas, garantindo então que as classes que implementa essas interfaces e funções não tenham comportamentos inúteis que não serão usados, promovendo a coesão e a modularidade do código, garantindo que as interfaces sejam enxutas e focadas em um conjunto específico de responsabilidades, ao invés de serem genéricas e abrangentes. Isso facilita a manutenção do código, tornando-o mais claro e menos propenso a erros.

-D: DIP - *Dependency Inversion Principle* (Princípio da inversão de dependência):

Por fim o DIP basicamente diz que as classes devem depender de abstrações, como interfaces ou classes abstratas, ao invés de ficarem atreladas a implementações concretas de uma classe. Essa ideia é fundamental pois torna o código mais flexível e fácil de manter, já que permite que diferentes partes do sistema sejam modificadas sem afetar umas às outras. Quando se usa interfaces, a lógica do sistema se descola da tecnologia e da linguagem específica que estamos

utilizando, porque as interfaces abstraem os conceitos fundamentais das classes subjacentes. Isso torna o código mais flexível, fácil de manter e independente de tecnologia específica. Em resumo, ajuda a construir um software modular e adaptável.

Durante o desenvolvimento aplicamos principalmente os conceitos do DIP, ISP e SRP, visto que são conceitos que acreditamos ser mais simples de implementar e que nos ajudou a entender o funcionamento do software e separar os conceitos. Vale salientar sobre o uso durante o processo de desenvolvimento do TMS visamos utilizar esses conceitos, porém não utilizamos necessariamente todos os conceitos.

4.2 ARQUITETURA DO SOFTWARE

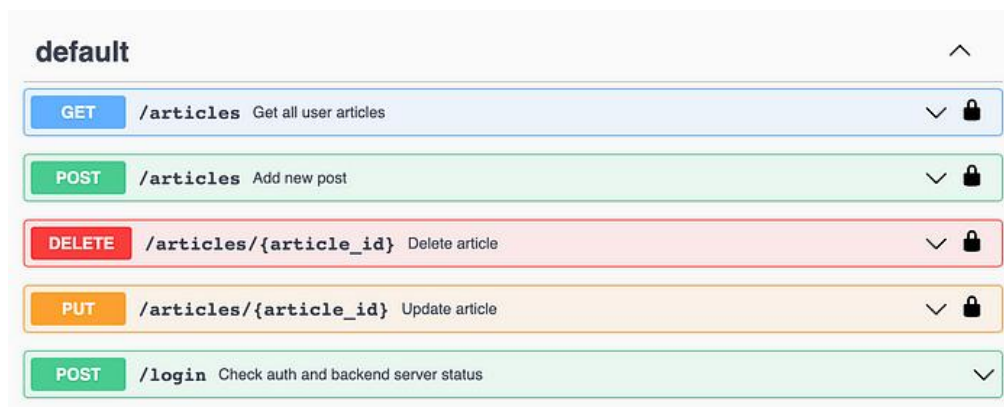
Referente a arquitetura, foi escolhido adotar o modelo cliente-servidor como estrutura principal para o desenvolvimento, uma vez que possibilita a segmentação das tarefas em duas vertentes distintas. A primeira delas é o *back-end* (também conhecido como *server-side*), responsável de receber, processar, manipular e retornar os dados aos clientes. Por sua vez, o *front-end* (*client-side*) é responsável pela interação com o usuário, englobando desde a recepção e validação dos dados inseridos pelo usuário, até o envio desses dados ao servidor, bem como a manipulação da apresentação dos dados aos usuários de acordo com a lógica estabelecida.

Para criação da API (*Application Programming Interface*) que disponibilizará os dados dos *clients*, optamos por seguir o estilo arquitetônico chamado GraphQL (*Graph Query Language*) que é um modelo relativamente novo desenvolvido pela empresa Meta (na época ainda atendia como Facebook) em 2012 como contraponto a outro estilo muito usado até hoje chamado REST (*Representational State Transfer*) que tem sido utilizado desde o início dos anos 2000, como forma de desacoplar uma API de seu *client*. O estilo REST tem como fundamento separar cada serviço disponibilizado pelo servidor em um *endpoint*³ (também chamado de rota), como

³ Um endpoint é uma URL específica ou um URI (*Uniform Resource Identifier*) que identifica um recurso específico em uma API. Esses *endpoints* são usados para realizar operações ou obter informações relacionadas a esse recurso. Por exemplo, uma API de mídia social pode ter *endpoints*

mostrado na Figura 3:

Figura 3 – Exemplos de Endpoints de uma API REST



Fonte: Medium, 2022

Como evidenciado na Figura 3, cada endpoint representa um serviço específico, como a rota/articles, que retorna todos os artigos, ou a rota/post, que adiciona um novo artigo. Contudo, quando consideramos a implementação desse sistema em um ambiente grande e complexo, onde podem existir inúmeros serviços, isso pode acarretar uma complexidade considerável tanto para os desenvolvedores *back-end*, responsáveis pela criação e manutenção dos serviços, quanto para os desenvolvedores *front-end*, que consomem esses serviços. Este cenário destaca a importância de estratégias eficazes de gerenciamento e organização para lidar com a complexidade crescente dos sistemas de software. (VIDANYA, 2021)

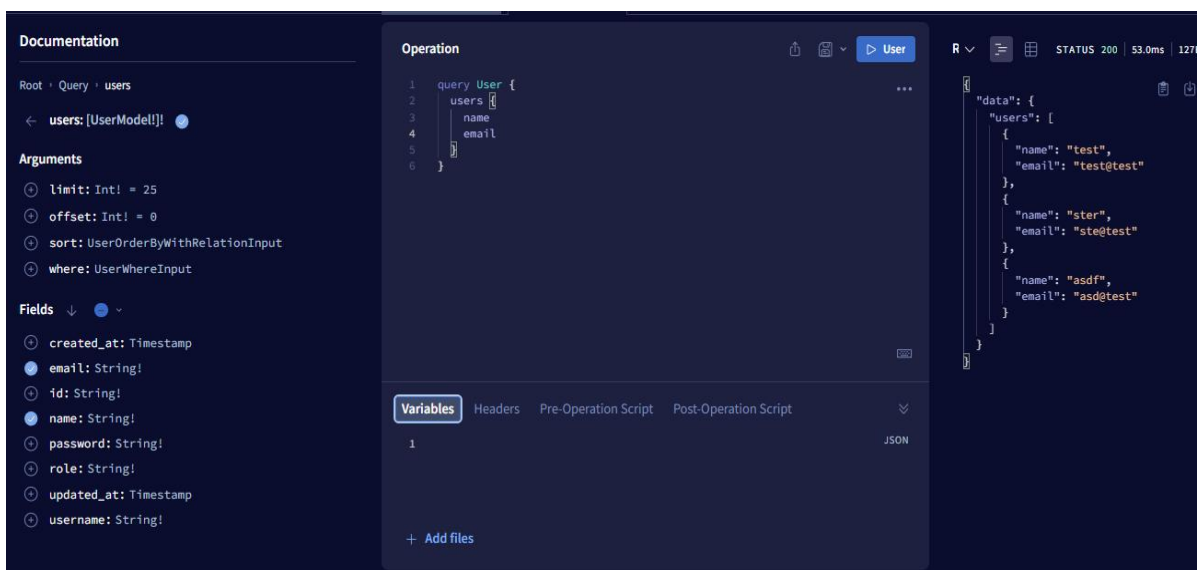
Considere também o cenário em que o consumidor da API precisa filtrar os artigos a partir de uma data específica. Nesse caso, existem apenas duas opções: criar um novo serviço no servidor (introduzindo mais uma rota para manter em caso de mudanças) ou baixar todos os artigos no cliente (aumentando o tráfego desnecessário na rede e o consumo de dados) para então filtrar os artigos. Esse problema é conhecido como *Over-fetching*. (VIDANYA, 2021)

Além disso, outra situação que pode surgir é quando o *front-end* necessita recuperar os artigos de todos os usuários e os dados dos autores. Isso coloca os desenvolvedores diante do dilema de criar um serviço ou ter que consumir dois serviços distintos (o que aumentaria a latência de resposta do sistema). Esse problema é denominado *under-fetching*. (VIDANYA, 2021)

para recuperar informações do perfil de um usuário, postar uma nova mensagem ou fazer uma pesquisa. (Alves,2023, Online)

Por outro lado, o GraphQL é criado para tudo estar concentrado em uma URL que disponibilizará todos os serviços, mas ele vai além disso, os seus clientes consomem através de consultas que possibilitam filtrar e buscar somente os dados que o cliente necessita, ficando o servidor responsável por entender e lidar com essas consultas, como demonstrado nas Figuras 4, 5 e 6 a seguir:

Figura 4 – GraphQL – query para procurar todos os emails e nomes de usuários



The screenshot shows the GraphQL Playground interface. On the left, the 'Documentation' panel displays the schema for the 'users' query, including arguments like 'limit', 'offset', 'sort', and 'where', and fields like 'created_at', 'email', 'id', 'name', 'password', 'role', 'updated_at', and 'username'. The 'Operation' panel contains the query:

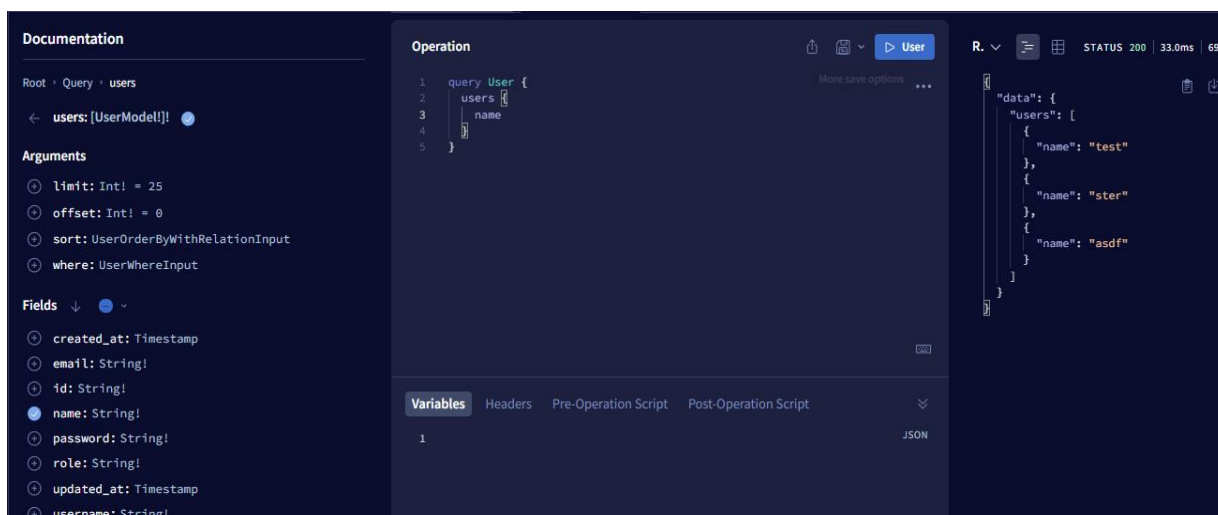
```
1 query User {
2   users {
3     name
4     email
5   }
6 }
```

 Below the query, the 'Variables' tab is active, showing an empty JSON object. On the right, the response is displayed in JSON format:

```
{
  "data": {
    "users": [
      {
        "name": "test",
        "email": "test@test"
      },
      {
        "name": "ster",
        "email": "ste@test"
      },
      {
        "name": "asdf",
        "email": "asd@test"
      }
    ]
  }
}
```

Fonte: Autoria própria, 2023.

Figura 5 – GraphQL: query para retorna apenas os nomes dos usuários



The screenshot shows the GraphQL Playground interface. On the left, the 'Documentation' panel displays the schema for the 'users' query. The 'Operation' panel contains the query:

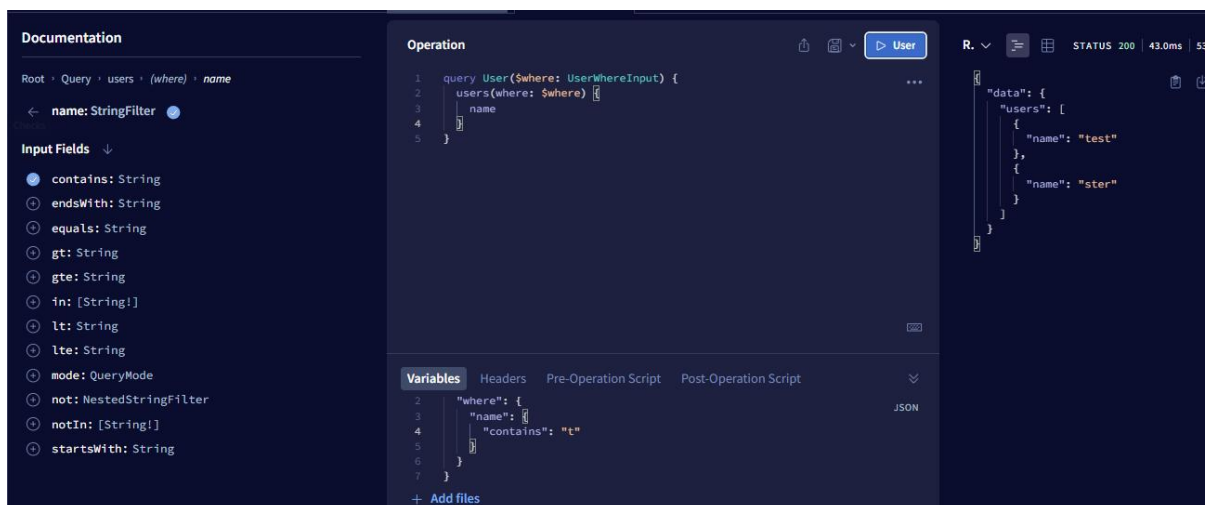
```
1 query User {
2   users {
3     name
4   }
5 }
```

 Below the query, the 'Variables' tab is active, showing an empty JSON object. On the right, the response is displayed in JSON format:

```
{
  "data": {
    "users": [
      {
        "name": "test"
      },
      {
        "name": "ster"
      },
      {
        "name": "asdf"
      }
    ]
  }
}
```

Fonte: Autoria própria, 2023.

Figura 6 – GraphQL: query para buscar os nomes dos usuários que contém a letra T



Fonte: Autoria própria, 2023.

É bom salientar que os dois modelos de desenvolvimento de API têm seus benefícios e maléficos, sendo que cada se sai melhor dependendo do contexto.

Só ressaltando que escolhemos o GraphQL pois o TMS tem uma quantidade enorme de entidades que se relacionam umas com a outra o que aumenta a complexidade e quantidade de serviços que o *back-end* deve fornecer para o *front-end* que por sua vez requer uma grande quantidade de telas e manipulação de dados que vão ser enviados para o *back-end* e assim se usássemos o modelo *rest* teríamos mais trabalho e dificuldade de criar e dar manutenção nos serviços.

4.3 STACK DE DESENVOLVIMENTO

Linguagem de programação: Como linguagem para desenvolver o projeto (indo do back end ao front) escolhemos o TypeScript (também conhecido como TS) que é uma linguagem de programação que traduz o código de uma linguagem para outra) código JavaScript (JS). Ele adiciona tipagem estática ao JS, o que ajuda a evitar muitos erros comuns durante o desenvolvimento e torna o código mais legível e escalável, reduzindo a ocorrência de erros e tornando o código mais fácil de entender, manter e escalar. (ROVEDA, 2023)

Apollo GraphQL: É uma implementação GraphQL completa em JS. GraphQL é uma linguagem de consulta para APIs e um tempo de execução para executar essas consultas com seus dados existentes. Com o Apollo, você pode facilmente

gerenciar o estado da aplicação com consultas GraphQL. **Eficiência e velocidade de desenvolvimento:** Simplifica o gerenciamento do estado da aplicação com consultas GraphQL, reduzindo a necessidade de múltiplas chamadas de API e manipulação manual do estado. **Escalabilidade e manutenibilidade:** Facilita a organização e a manutenção do código ao centralizar as consultas de dados em um único local.

Next.js: Next.js é um *framework* de desenvolvimento Web React de código aberto que permite funcionalidades como renderização do lado do servidor (SSR) e geração de páginas estáticas. Ele oferece uma ótima experiência de desenvolvimento, performance e escalabilidade. **Eficiência e velocidade de desenvolvimento:** Oferece funcionalidades como SSR (*Server-Side Rendering*) e geração de páginas estáticas, reduzindo o tempo de carregamento e melhorando a experiência do usuário. **Performance:** Melhora a performance do aplicativo, especialmente em termos de SEO e tempo de carregamento da página.

Radix UI: Radix UI é uma biblioteca de componentes React acessíveis e com foco em desempenho. Ela fornece uma variedade de componentes prontos para uso que podem ser facilmente estilizados e personalizados. **Eficiência e velocidade de desenvolvimento:** Fornece uma variedade de componentes prontos para uso, reduzindo a necessidade de desenvolvimento e estilização manual de componentes UI.

Three.js: Three.js é uma biblioteca JavaScript/API para criar e exibir gráficos 3D interativos em navegadores web. É frequentemente usado para criar visualizações 3D impressionantes e experiências interativas na web. **Experiência do usuário:** Permite a criação de gráficos 3D interativos, melhorando a experiência do usuário e permitindo a criação de visualizações complexas e envolventes.

Storybook: Storybook é uma ferramenta de desenvolvimento de UI de código aberto para React, Vue e Angular. Ele permite desenvolver componentes de forma isolada e interativa, facilitando a visualização e o teste de diferentes estados e props⁴ de um componente.

Testing Library: Testing Library é uma família de bibliotecas de teste para escrever testes de unidade e integração em JavaScript. Ela promove uma

⁴ Props são mecanismos que permitem transmitir dados de um componente pai para um componente filho

abordagem centrada no usuário para testes, incentivando os desenvolvedores a escreverem testes que espelhem o comportamento do usuário.

PWA (*Progressive Web App*): PWA é uma aplicação web que utiliza tecnologias web modernas para fornecer uma experiência de usuário semelhante a aplicativos nativos, incluindo funcionalidades como trabalhar offline, notificações *push* e acesso ao hardware do dispositivo.

Tailwind CSS: Tailwind CSS é uma estrutura de design CSS de baixo nível que fornece classes utilitárias para ajudar a construir interfaces de usuário rapidamente. Ele permite estilizar aplicativos sem a necessidade de escrever CSS personalizado, o que pode acelerar o desenvolvimento.

4.4 BANCO DE DADOS: POSTGRESQL

Existem diversos bancos de dados e cada um com suas características e funções específicas como o MongoDB que é orientado a documento ou o NoSQL que é não relacional, porém optamos um DB (Data Base) relacional (que armazena os dados em tabelas linhas e colunas) e que seja robusto já que um TMS envolve muitos relacionamentos entre tabelas e uma grande quantidade de entidades. Por este motivo escolhemos PostgreSQL por ser um DB relacional e robusto que garante o armazenamento de diversos tipos de dados como XML, JSON, etc., que comparado ao MySQL (seu maior concorrente quando se trata de DB relacional) se mostra um pouco melhor em alguns casos como apresentado nessa tabela 1 conforme o site da *Amazon Aws* (2024)

Quadro 1 - Diferenças entre *Mysql* e *PostgreSQL*

BD	Mysql	PostgreSQL
Tipo de BD	Relacional	Relacional
Recursos	Suporte limitado a recursos como <i>triggers</i> (gatinhos) e <i>procedures</i> (procedimentos)	Suporta recursos avançados como <i>triggers</i> e <i>procedures</i> além de visões materializadas
Tipo de dados	Suporta dados numéricos, caracteres, data e hora, espaciais e json	Suporta todos os tipos do Mysql, mais arrays, xml, matrizes, enums e etc.

Performance	O mysql tem melhor performance para operações de leitura de alta frequência	Melhor para operações de gravação de dados
Uso	Possui ferramentas que facilita o seu uso como o Xampp que cria uma interface web que permite manipular o BD de forma simples e sem a necessidade de conhecimento profundo	Não é muito amigável com pessoas não técnicas e não possui muitas ferramentas que possa facilitar sua manipulação.

Fonte: Amazon Aws (2024)

4.5 DESCRIÇÃO DO SOFTWARE

O TMS educativo em desenvolvimento tem como principal objetivo capacitar os professores a instruírem os alunos sobre o funcionamento prático de um TMS. Este software tem a responsabilidade de simular um ambiente semelhante ao mundo real, embora não seja necessariamente real, uma vez que exigiria integrações com sistemas reais, como os de emissão de conhecimento de transporte, validação de CNH, ANTT, entre outros.

Entre as funcionalidades do software, está a capacidade de criar usuários, isso permite que cada aluno tenha sua senha usuário e realize suas próprias operações, garantindo que o professor saiba o que cada aluno está fazendo.

4.6 ESCOPO DO SOFTWARE

Como o objetivo do software é simular um TMS real, ele deverá contar com os seguintes cadastros e funções:

- Gerenciamento de usuários.
- Gerenciamento de motoristas e veículos
- Geração de contratos.

- Gerenciamento de pedido (frete)
- Gerenciamento de rotas de transporte
- Gerenciamento de manutenção de veículo
- Gerenciamento de empresa de manutenção

A fim de obter tal resultados, o software desenvolvido deverá:

- 1) Ser uma ferramenta útil que ajude a alunos a entender o funcionamento e importância de se gerenciar frotas e o frete em empresas de logísticas de forma prática e eficiente;
- 2) Auxiliar professores que tem como objetivo mostrar o funcionamento de um TMS e explicar na prática como funciona, suas vantagens, características etc. e;
- 3) Auxiliar alunos a testar na prática conhecimentos obtidos em suas aulas como por exemplo o cálculo de frete, em um modelo de software que é usado nas empresas.

4.7 REQUISITOS DO SOFTWARE

Os requisitos de software especificam os serviços e as características necessárias ao sistema, bem como as funções necessárias para garantir que as funcionalidades sejam entregues corretamente e de acordo com o esperado (SOMMERVILLE, 2011). As quadro a seguir, apresentam os requisitos funcionais e não funcionais respectivamente.

4.7.1 Requisitos Funcionais

Quadro 2 – Requisitos Funcionais

RF01	O software deve permitir cadastrar, atualizar usuários
RF02	O software deve permitir cadastrar, atualizar e excluir motoristas
RF03	O software deve permitir cadastrar, atualizar e excluir veículos
RF04	O software deve permitir cadastrar, atualizar e excluir marca de veículos
RF05	O software deve permitir cadastrar, atualizar e excluir tipos de veículos

RF06	O software deve permitir cadastrar, atualizar e excluir carroceria de veículo
RF07	O software deve permitir cadastrar, atualizar e excluir contratos com pessoas jurídicas
RF08	O software deve permitir cadastrar, atualizar e excluir clientes jurídicos e físicos
RF09	O software deve permitir cadastrar, atualizar processamento de pedido
RF10	O software deve permitir cadastrar, atualizar e excluir sinistros
RF11	O software deve permitir cadastrar, atualizar veículos que estão em manutenção
RF12	O software deve permitir cadastrar, atualizar e excluir empresas de manutenção
RF13	O software deve permitir cadastrar, atualizar e excluir tipos de manutenção

4.7.2 Requisitos Não Funcionais

Quadro 3 - Requisitos Não Funcionais

Requisito	Propriedade	Medida
RNF01	Desempenho	O software deve ser capaz de operar de maneira ágil para melhorar a produtividade e usabilidade dos usuários
RNF02	Segurança	No software, é necessário que cada usuário tenha um tipo de acesso que corresponda à sua função e que seja autenticado no momento de ingresso no sistema e que a senha criada no login seja criptografada.
RNF03	Usabilidade	O software deve ter uma interface gráfica de usuário intuitiva e fácil de usar, com menus de navegação claramente identificados e botões de fácil acesso.

RNF04	Confiabilidade	É necessário que o sistema tenha acesso a internet para o bom funcionamento
RNF05	Portabilidade	O sistema pode ser executado em Windows, MAC, Linux e mobile visto que se trata de um software web, com versões para <i>desktop</i> e <i>mobile</i>

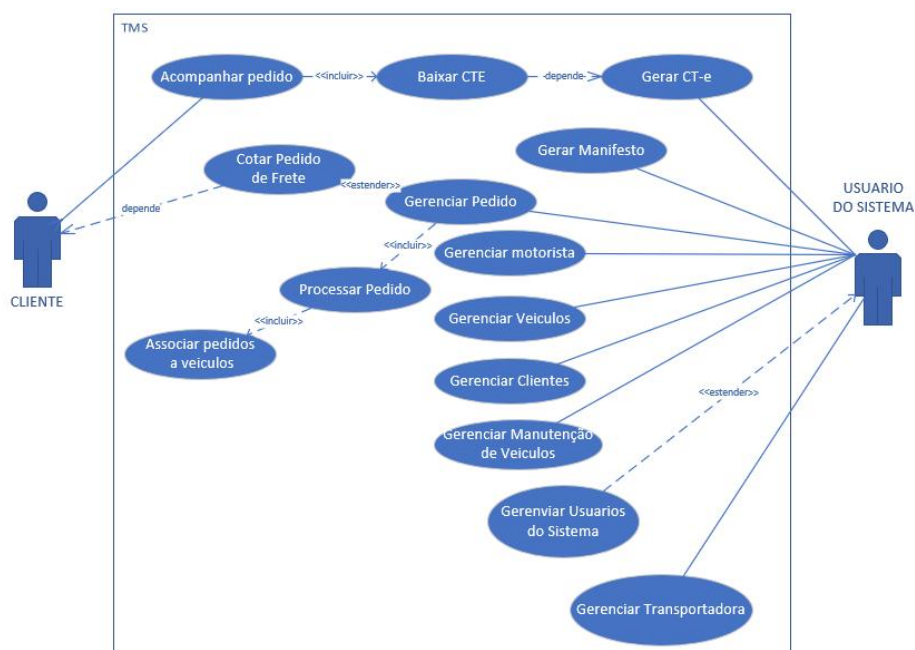
4.8 ESPECIFICAÇÃO DO SOFTWARE

Nesta seção serão apresentados alguns diagramas que trarão mais contexto e entendimento sobre como o software deve funcionar e como o usuário irá interagir com software.

4.8.1 Caso de uso

O caso de uso da Figura 7 a seguir, mostra uma visão geral do funcionamento do software

Figura 7 - Caso de uso do TMS

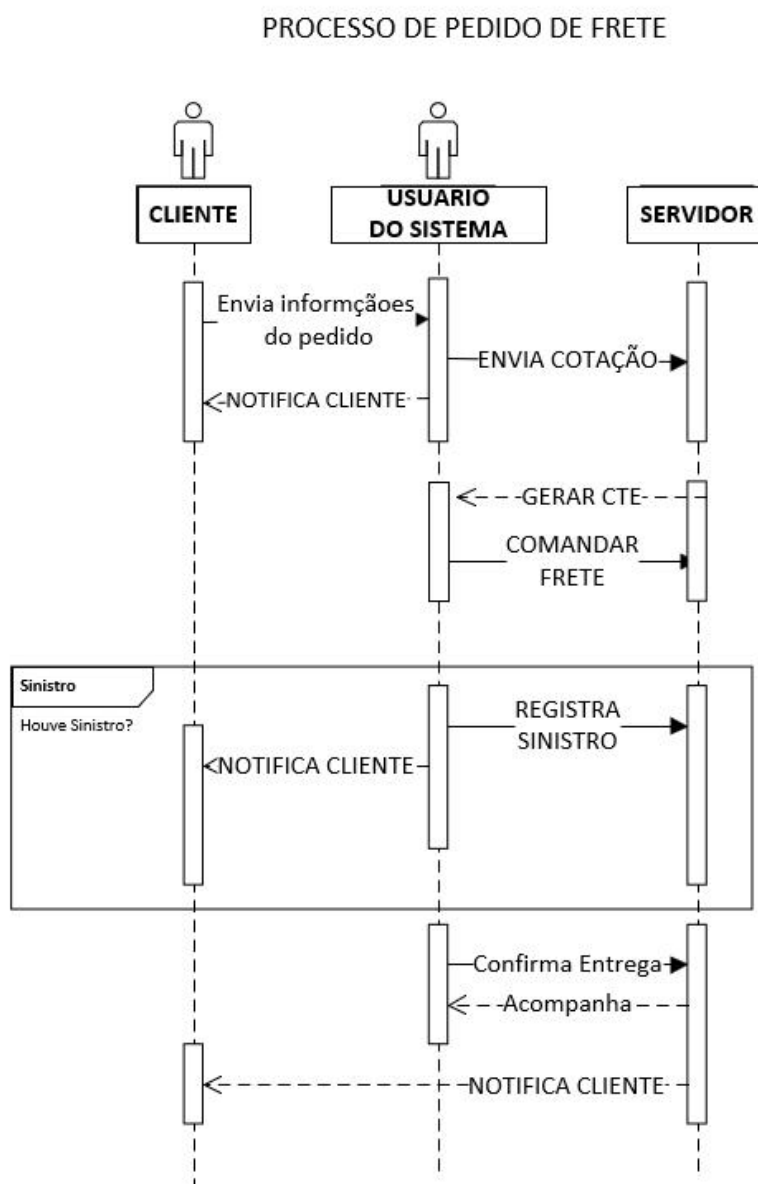


Fonte: Autoria própria, 2024.

4.8.2 Diagrama de Sequência

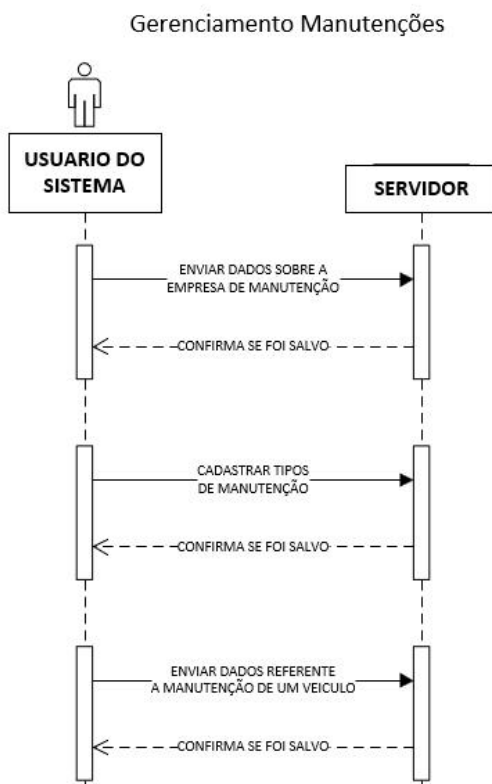
A ideia dos diagramas de sequência desenvolvidos, apresentados nas Figuras 8 a 12, é mostrar de forma mais detalhada cada processo do TMS. Vale ressaltar que os diagramas que possuem a palavra **gerenciamento** significam que o usuário do sistema tem o controle sobre criar, atualizar, deletar e ver todos os dados.

Figura 8 - Diagrama de sequência de pedido de frete



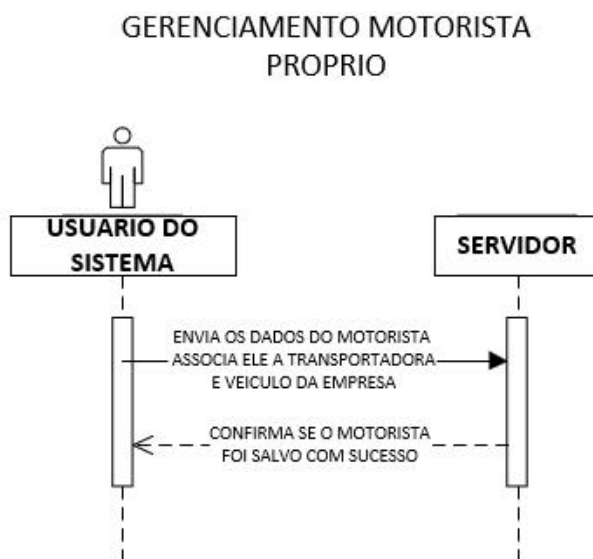
Fonte: Autoria própria, 2024.

Figura 9 - Diagrama de sequência do gerenciamento de manutenção



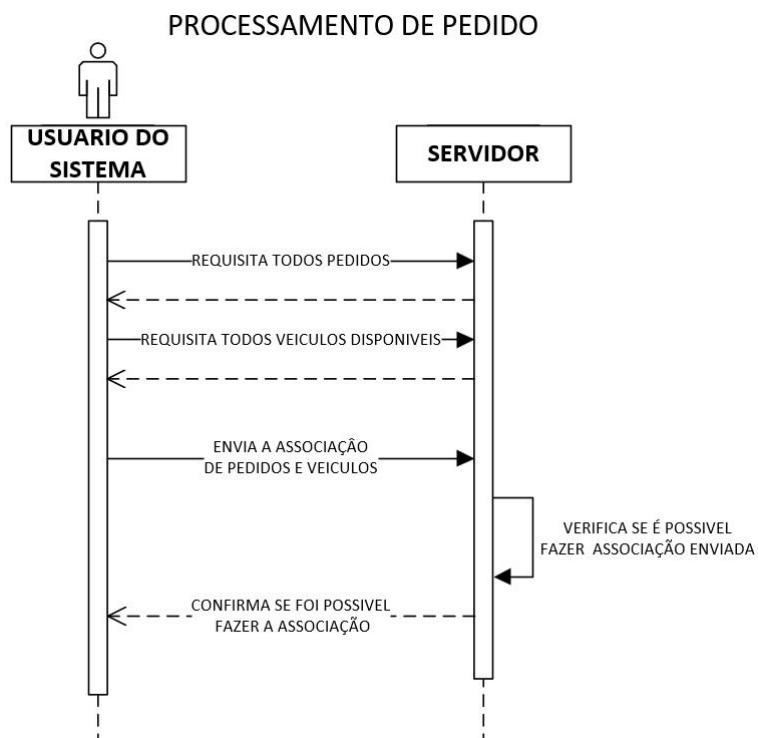
Fonte: Autoria própria, 2024.

Figura 10- Diagrama de sequência do gerenciamento de motorista próprio



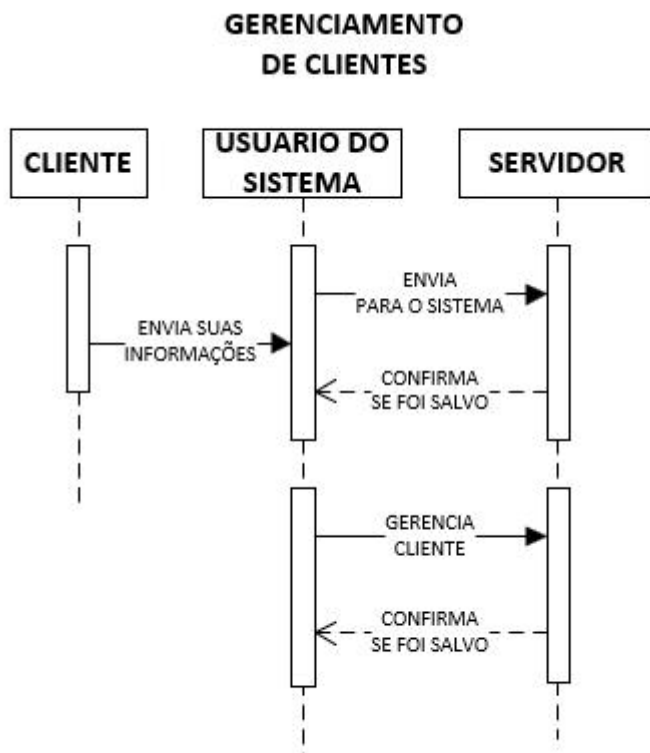
Fonte: Autoria própria, 2024.

Figura 11- Diagrama de seqüência do processamento de pedido



Fonte: Autoria própria, 2024.

Figura 12- Diagrama de seqüência do gerenciamento de clientes



Fonte: Autoria própria, 2024.

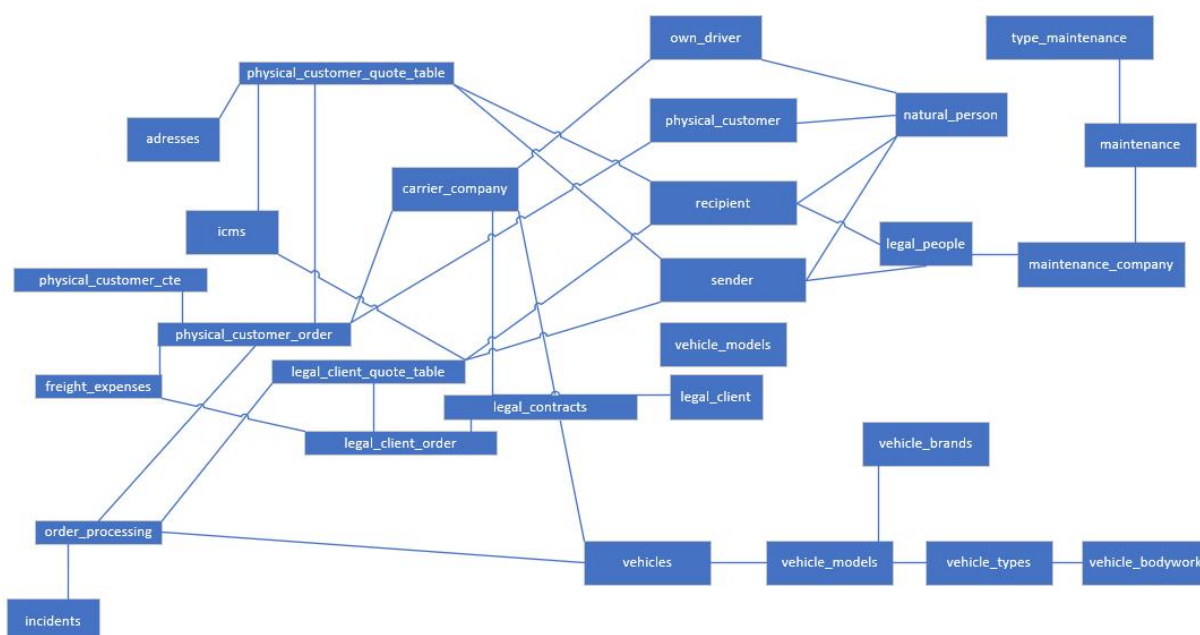
4.9 DIAGRAMA DE ENTIDADE RELACIONAMENTO (DER)

Um Diagrama de Entidade-Relacionamento (DER) é uma representação visual da estrutura de um banco de dados, mostrando as tabelas (entidades), as informações que cada tabela armazena (atributos) e como elas estão relacionadas. Ele ajuda os desenvolvedores a entenderem como os dados estão organizados e como eles se relacionam entre si.

4.9.1 BANCO DE DADOS

Por conta da complexidade e tamanho do banco de dados do TMS, é inviável mostrar todo o banco então optamos por fazer uma imagem que mostra de forma geral as conexões das tabelas, conforme Figura 13.

Figura 13 - Estrutura Geral







Fonte: Autoria própria, 2024.

4.9.2 Estrutura de Dados

Para garantir um certo controle sobre os usuários que estão manipulando os dados, grandes partes das tabelas (Figuras 14 e 15) tem relação direta com um usuário e essas conexões ficaram com o nome de *created_by* (em português: **criado por** que tem o objetivo de relacionar o usuário com o dado que ele criou), *updated_by* (**atualizado por** que relaciona o último usuário que atualizou com o dado), *created_at* (**Criado em** que armazena o momento exato que o dado foi criado) e por fim *updated_at* (**Atualizado em** que armazena o momento exato que o dado foi atualizado).

Figura 14 – Estrutura da Tabela de Usuários (parte 1)

users	
id 	String
name	String NN
username	String NN
email	String NN
role	String NN
avatar_url	String
password 	String NN
created_at 	DateTime NN
updated_at 	DateTime NN
VehicleModelCreatedBy	vehicle_models NN
VehicleModelUpdatedBy	vehicle_models NN
VehicleBrandCreatedBy	vehicle_brands NN
VehicleBrandUpdatedBy	vehicle_brands NN
VehicleTypeCreatedBy	vehicle_types NN
VehicleTypeUpdatedBy	vehicle_types NN
VehicleBodyworkCreatedBy	vehicle_bodyworks NN
VehicleBodyworkUpdatedBy	vehicle_bodyworks NN
OutsourcedVehicleCreatedBy	outsourced_vehicles NN
OutsourcedVehicleUpdatedBy	outsourced_vehicles NN
CompanyVehicleCreatedBy	company_vehicles NN
CompanyVehicleUpdatedBy	company_vehicles NN
OutsourcedDriverCreatedBy	outsourced_drivers NN
OutsourcedDriverUpdatedBy	outsourced_drivers NN
ContractOutsourcedDriverCreatedBy	contract_outsourced_drivers NN
ContractOutsourcedDriverUpdatedBy	contract_outsourced_drivers NN
OwnDriverCreatedBy	own_drivers NN
OwnDriverUpdatedBy	own_drivers NN
CarrierCompanyCreatedBy	carrier_companies NN

Fonte: Autoria própria, 2024.

Figura 15 - Estrutura Tabela de Usuário (parte 2)

PhysicalCustomerCreatedBy	physical_customers NN
PhysicalCustomerUpdatedBy	physical_customers NN
MaintenanceCreatedBy	maintenance NN
MaintenanceUpdatedBy	maintenance NN
MaintenanceCompanyCreatedBy	maintenance_companies NN
MaintenanceCompanyUpdatedBy	maintenance_companies NN
TypeOfMaintenanceCreatedBy	types_of_maintenances NN
TypeOfMaintenanceUpdatedBy	types_of_maintenances NN
PhysicalCustomerOrderUpdatedBy	physical_orders NN
PhysicalCustomerOrderCreatedBy	physical_orders NN
LegalClientCustomerOrderCreatedBy	legal_orders NN
LegalClientCustomerOrderUpdatedBy	legal_orders NN
CiotsForLegalUpdatedByClient	ciots_for_legal_clients NN
CiotsForLegalCreatedByClient	ciots_for_legal_clients NN
LegalClientUpdatedBy	legal_clients NN
LegalClientCreatedBy	legal_clients NN
LegalContractCreatedBy	legal_Contracts NN
LegalContractUpdatedBy	legal_Contracts NN
OutsourcedTransportCompanyUpdatedBy	OutsourcedTransportCompany NN
OutsourcedTransportCompanyCreatedBy	OutsourcedTransportCompany NN
OutsourcedTransportVehicleCreatedBy	outsourced_transport_vehicle NN
OutsourcedTransportVehicleUpdatedBy	outsourced_transport_vehicle NN
OutsourcedTransportCompanyDriverCreatedBy	outsourcedT_transport_company_driver NN
OutsourcedTransportCompanyDriverUpdatedBy	outsourcedT_transport_company_driver NN
OutsourcedTransportCompanyContractCreatedBy	OutsourcedTransportCompanyContract NN
OutsourcedTransportCompanyContractUpdatedBy	OutsourcedTransportCompanyContract NN
RecipientUpdatedBy	recipient NN
RecipientCreatedBy	recipient NN
IcmsUpdatedBy	icms NN
IcmsCreatedBy	icms NN
SenderCreatedBy	sender NN
SenderUpdatedBy	sender NN
LegalClientQuoteTableCreatedBy	legal_client_quote NN
LegalClientQuoteTableUpdatedBy	legal_client_quote NN
PhysicalCustomerQuoteTableCreatedBy	physical_customer_quote NN
PhysicalCustomerQuoteTableUpdatedBy	physical_customer_quote NN
IncidentUpdatedBy	Incident NN
IncidentCreatedBy	Incident NN

Fonte: Autoria própria, 2024.

4.9.3 Tabela relacionadas com Veículos

A entidade de veículo é fundamental para todo TMS pois ela é o núcleo para a maioria dos processos que ele manipula por isso é fundamental que ela seja bem-organizada e criada. Dito isto resolvemos dividir a entidade veículos em algumas tabelas que garante que todos os veículos cadastrados estejam padronizados e organizados.

Assim a entidade *vehicle_bodyworks* (Figura 16) tem como objetivo cadastrar todos os tipos de carrocerias que os veículos podem ter. Já a entidade *Vehicle_types* (Figura 17) serve para linkar e criar tipos de veículos (exemplo: caminhonete, moto etc.) a as carrocerias que esse tipo pode ter, exemplo: o tipo caminhão pode ter a carroceria baú fechado ou baú lonado. A entidade *Vehicle_brands* (Figura 18) serve para cadastrar as marcas de veículos como Ford e Fiat. Enquanto a entidade *Vehicle_models* (Figura 19) serve para criar um modelo de veículo e linkar com uma marca de veículo (*Vehicle_brands*) e um tipo de veículos (*Vehicle_types*), exemplo: o modelo Accelo Euro6 da marca Mercedes do tipo caminhão que contém um baú fechado. Por fim a entidade *vehicles* (Figura 20) serve para criar um veículo único terá seus dados como placa, cor etc.

Figura 16 – Estrutura da tabela de carroceria de veículos

vehicle_bodyworks	
id 	String
name	String NN
axles	Int NN
mass	Float NN
volume	Float NN
created_at 	DateTime NN
updated_at 	DateTime NN
CreatedBy	users
created_by	String NN
UpdatedBy	users
updated_by	String NN
VehicleType	vehicle_types NN




Fonte: Autoria própria, 2024.

Figura 17 – Estrutura da tabela de tipo de veículos

vehicle_types	
id 	String
name	String NN
bodywork 	Boolean NN
created_at 	DateTime NN
updated_at 	DateTime NN
VehicleModel	vehicle_models NN
VehicleBodyWork	vehicle_bodyworks NN
CreatedBy	users
created_by	String NN
UpdatedBy	users
updated_by	String NN

Fonte: Autoria própria, 2024.

Figura 18 – Estrutura da tabela de marcas de veículos

vehicle_brands	
id 	String
name	String NN
created_at 	DateTime NN
updated_at 	DateTime NN
VehicleModel	vehicle_models NN
CreatedBy	users
created_by	String NN
UpdatedBy	users
updated_by	String NN

Fonte: Autoria própria, 2024.

Figura 19 - Estrutura da tabela de modelo de veículo

vehicle_models	
id 	String
name	String NN
weight	Float NN
capacity_max	Float NN
axles	Int NN
capacity_per_axle	Int
created_at 	DateTime NN
updated_at 	DateTime NN
Vehicle	vehicles NN
CreatedBy	users
created_by	String NN
UpdatedBy	users
updated_by	String NN
Brand	vehicle_brands NN
brand_id	String NN
VehicleType	vehicle_types NN
type_id	String NN

Fonte: Autoria própria, 2024.

Figura 20 - Estrutura da tabela de veículos

vehicles	
id 	String
plate	String NN
year	String NN
color	String NN
renavam	String NN
antt	String NN
registration	DateTime NN
is_ipva_paid	Boolean NN
CompanyVehicle	company_vehicles
Maintenance	maintenance NN
Model	vehicle_models NN
model_id	String NN
OrderProcessing	order_processing

Fonte: Autoria própria, 2024.

Por fim criada o veículo (*vehicle*), pode se então relacioná-lo com a transportadora que ele pertence (*company_vehicles*) como mostrado na Figura 21.

Figura 21 – Estrutura da tabela de motorista próprios

company_vehicles	
id	String
created_at	DateTime NN
updated_at	DateTime NN
company_id	String NN
CarrierCompany	carrier_companies NN
CreatedBy	users
created_by	String NN
UpdatedBy	users
updated_by	String NN
Vehicle	vehicles NN
vehicle_id	String NN

Fonte: Autoria própria, 2024.

4.9.4 Tabelas relacionadas com Pessoas Físicas

No TMS é fundamental o cadastro de pessoas físicas visto que inúmeros atores desse tipo atuam no software, como os motoristas, clientes, destinatários etc., assim sendo a Figura 22 a seguir apresenta a estrutura da tabela de pessoa física (*natural_people*).

Figura 22 – Estrutura da tabela de pessoa física

natural_people	
id	String
name	String NN
date_birth	DateTime NN
gender	String NN
cpf	String NN
rg	String NN
cep	String NN
public_place	String NN
address_number	String NN
neighborhood	String NN
complement	String
city	String NN
uf	String NN
first_phone	String NN
second_phone	String
third_phone	String
email	String NN
nationality	String NN
OwnDriver	own_drivers NN
PhysicalCustomer	physical_customers NN
Recipient	recipient
Sender	sender

Fonte: Autoria própria, 2024.

4.9.5 Tabela de Pessoa Jurídica

Outra tabela importante no DB é a de pessoas jurídicas visto que assim como a tabela de pessoa física possuem inúmeros atores como a transportadora, clientes jurídicos, remetentes etc. Na Figura 23 a seguir é mostrado a estrutura da tabela.

Figura 23 - Estrutura da tabela de pessoa jurídica

legal_people	
id	String
fantasy_name	String NN
cnpj	String NN
state_registration	String NN
cep	String NN
corporate_name	String NN
public_place	String NN
address_number	String NN
neighborhood	String NN
complement	String
city	String NN
uf	String NN
first_phone	String NN
second_phone	String
third_phone	String
email	String NN
created_at	DateTime NN
updated_at	DateTime NN
CorporateClient	legal_clients NN
MaintenanceCompany	maintenance_companies
CarrierCompany	carrier_companies NN
Recipient	recipient
Sender	sender

Fonte: Autoria própria, 2024.

4.9.6 Tabela de Motorista Próprio

A tabela de motorista (Figura 24) tem como fundamento linkar com um registro na tabela de pessoa física e colocar os dados relacionados ao motorista como a CNH, a categoria etc.

Figura 24 - Estrutura da tabela de veículos próprio

own_drivers	
id	String
cnh	String NN
cnh_category	String NN
cnh_expiration	DateTime NN
course_mopp	Boolean NN
created_at	DateTime NN
updated_at	DateTime NN
CreatedBy	users
created_by	String NN
UpdatedBy	users
updated_by	String NN
NaturalPerson	natural_people NN
natural_person_id	String NN
OrderProcessing	order_procesing NN

Fonte: Autoria própria, 2024.

4.9.7 Tabela de Transportadora

A tabela tem como fundamento criar as transportadoras responsáveis pela prestação de serviço do frete, sendo assim ela terá um link com a tabela de pessoa jurídica e o campo RNTRC⁵ da empresa, como demonstrado na Figura 25.

⁵ O RNTRC (Registro Nacional de Transportadores Rodoviários de Cargas) é um documento obrigatório emitido pela ANTT, criado com objetivo de identificar e cadastrar os transportadores rodoviários de cargas, incluindo veículos. Fonte: <https://www.totvs.com/blog/gestao-logistica/rntrc-o-que-e/>

Figura 25 - Estrutura da tabela de transportadoras

carrier_companies	
id	String
LegalPerson	legal_people NN
legal_person_id	String NN
LegalContract	legal_Contracts NN
CompanyVehicle	company_vehicles NN
rntrc	String NN
created_at	DateTime NN
updated_at	DateTime NN
CreatedBy	users
created_by	String NN
UpdatedBy	users
updated_by	String NN
LegalClientOrder	legal_orders NN
PhysicalCustomerOrder	physical_orders NN

Fonte: Autoria própria, 2024.

4.9.8 Tabelas de clientes

Os clientes são responsáveis por realizar o pedido de frete, sendo fundamental que estejam devidamente cadastrados no sistema. É importante ressaltar que existem dois tipos de clientes: o cliente físico e o cliente jurídico. Ambos têm a possibilidade de contratar uma transportadora para a realização de múltiplos fretes. A Figura 26 mostra a estrutura da tabela de clientes físicos enquanto a Figura 27 mostra a estrutura da tabela de cliente jurídico.

Figura 26 - Estrutura da tabela de cliente físico

physical_customers	
id	String
branch	String
NaturalPerson	natural_people NN
natural_person_id	String NN
CreatedBy	users NN
created_by	String NN
UpdatedBy	users NN
updated_by	String NN
created_at	DateTime NN
updated_at	DateTime NN
PhysicalCustomerOrder	physical_orders NN

Fonte: Autoria própria, 2024.

Figura 27 - Estrutura da tabela de Cliente Jurídico

legal_clients	
id	String
branch	String NN
LegalPerson	legal_people NN
legal_person_id	String NN
CreatedBy	users NN
created_by	String NN
UpdatedBy	users NN
updated_by	String NN
LegalContract	legal_Contracts NN

Fonte: Autoria própria, 2024.

4.9.9 Tabela de Destinatário e Remetente

O cliente pode não querer realizar o frete para si mesmo. Portanto, durante o processo de cotação de frete, ele pode especificar quem será o destinatário e o

remetente do pedido. Por essa razão, são necessárias tabelas específicas para destinatário e remetente (Figura 28) e destinatário (Figura 29).

Vale ressaltar que a construção dessas tabelas é um pouco diferente. Ao contrário das tabelas de cliente físico e cliente jurídico, que são vinculadas às tabelas de pessoa física e pessoa jurídica, respectivamente, decidimos criar de forma conjunta os tipos de remetente físico e jurídico, bem como destinatário físico e jurídico.

Figura 28 - Estrutura de dados da tabela de remetente

recipient	
id	String
legal_person_id	String
LegalPerson	legal_people
natural_person_id	String
NaturalPerson	natural_people
created_at	DateTime NN
updated_at	DateTime NN
CreatedBy	users NN
created_by	String NN
updated_by	String NN
UpdatedBy	users NN
LegalClientQuoteTable	legal_client_quote NN
PhysicalCustomerQuoteTable	physical_customer_quote NN

Fonte: Autoria própria, 2024.

Figura 29 – Estrutura de dados da tabela de destinatário

sender	
id	String
legal_person_id	String
LegalPerson	legal_people
natural_person_id	String
NaturalPerson	natural_people
created_at	DateTime NN
updated_at	DateTime NN
CreatedBy	users NN
created_by	String NN
updated_by	String NN
UpdatedBy	users NN
LegalClientQuoteTable	legal_client_quote NN
PhysicalCustomerQuoteTable	physical_customer_quote NN

Fonte: Autoria própria, 2024.

4.9.10 Tabela de ICMS

Antes de iniciar qualquer processo de frete é necessário cadastrar os ICMS de cada estado para simular, mesmo que de forma superficial, as cobranças de imposto que incidem no processo de frete, por essa razão criamos essa tabela (Figura 28) para permitir um certo realismo nesse processo.

Figura 30 - Estrutura da Tabela de ICMS

Column Name	Data Type	Constraints
id	String	PK
state_origin	String	NN
recipient_state	String	NN
aliquot	Float	NN
effective_date	DateTime	NN
created_at	DateTime	NN
updated_at	DateTime	NN
CreatedBy	users	NN
created_by	String	NN
updated_by	String	NN
UpdatedBy	users	NN
PhysicalCustomerQuoteTable	physical_customer_quote	NN
LegalClientQuoteTable	legal_client_quote	NN

Fonte: Autoria própria, 2024.

4.9.11 Tabelas de Cotação

A ideia da criação das tabelas de cotação é armazenar as informações iniciais do frete, sendo, portanto, de suma importância para o processo de transporte. Essas tabelas contêm dados dos remetentes, destinatários, destino da carga, dados da nota fiscal, entre outros. Assim sendo temos as Figuras 31 e 32 que mostra a estrutura da tabela de cotação para clientes jurídicos e as Figuras 33 e 34 para cotação para clientes físicos.

Figura 31 - Estrutura da tabela de cotação de clientes jurídicos (parte 1)

legal_client_quote	
id	String
cod_quote	String NN
kind_service	String NN
form_payment	String NN
recipient_id	String NN
Recipient	recipient NN
Sender	sender NN
sender_id	String NN
who_pays	String NN
adress_origin_id	String NN
AdressOrigin	adresses NN
AdressDestiny	adresses NN
adress_destiny_id	String NN
type_merchandise	String NN
amount	Int NN
description	String NN

Fonte: Autoria própria, 2024.

Figura 32 - Estrutura da tabela de cotação de clientes jurídicos (parte 2)

mass	Float NN
volume	Float NN
nf_value	Float NN
nf_serie	String NN
nf_number	String NN
digital_signature	String NN
LegalClientOrder	legal_orders NN
created_at	DateTime NN
updated_at	DateTime NN
CreatedBy	users NN
created_by	String NN
UpdatedBy	users NN
updated_by	String NN
icms_id	String NN
icms	icms NN

Fonte: Autoria própria, 2024.

Figura 33 - Estrutura de dados da Tabela de Cotação de clientes Físicos (Parte 1)

physical_customer_quote	
id 	String
cod_quote	String NN
kind_service	String NN
form_payment	String NN
recipient_id	String NN
Recipient	recipient NN
Sender	sender NN
senderId	String NN
who_pays	String NN
adress_origin_id	String NN
AdressOrigin	addresses NN
adress_destiny_id	String NN
AdressDestiny	addresses NN
type_merchandise	String NN
amount	Int NN
description	String NN

Fonte: Autoria própria, 2024.

Figura 34 - Estrutura de dados da Tabela de Cotação de clientes Físicos (Parte 2)

mass	Float NN
volume	Float NN
nf_value	Float NN
nf_serie	String NN
nf_number	String NN
digital_signature	String NN
created_at 	DateTime NN
updated_at 	DateTime NN
CreatedBy	users NN
created_by	String NN
updated_by	String NN
UpdatedBy	users NN
PhysicalCustomerOrder	physical_orders NN
icms_id	String NN
lcms	icms NN

Fonte: Autoria própria, 2024.

4.9.12 Contrato de Frete para Clientes Jurídicos

A tabela contrato de frete para clientes Jurídicos foi criada com o intuito de simular de forma mais precisas o funcionamento de frete para empresas, visto que nesses casos pode se criar um contrato passando informações importantes referente ao frete, como as condições de entrega, a data de entrega entre outras informações importantes. A Figura 35 demonstra a sua estrutura.

Figura 35 - Estrutura de dados da tabela de contratos de frete para clientes jurídicos

legal_Contracts	
id	String
contract_number	String NN
LegalClient	legal_clients NN
legal_client_id	String NN
CarrierCompany	carrier_companies NN
carrier_company_id	String NN
observations	String
delivery_conditions	String NN
effective_date	DateTime NN
LegalClientOrder	legal_orders NN
Ciot	ciots_for_legal_clients NN
created_at	DateTime NN
updated_at	DateTime NN
CreatedBy	users NN
created_by	String NN
UpdatedBy	users NN
updated_by	String NN

Fonte: Autoria própria, 2024.

4.9.11 Tabela de Gastos de frete

Como o nome sugere, a ideia dessa tabela (Figura 36) é armazenar todos os gastos ocorridos em um frete.

Figura 36 - Estrutura da Tabela Gasto de Frete

freight_expenses	
id 	String
expense_name	String NN
value	Float NN
legal_client_order_id	String
physical_customer_id	String
LegalClientOrder	legal_orders
PhysicalCustomerOrder	physical_orders

Fonte: Autoria própria, 2024.

4.9.11 Tabelas de Pedidos

Existem duas tabelas de pedidos sendo que elas seguem o mesmo estilo, porém uma é para clientes jurídicos (Figura 37) e outra para clientes físicos (Figura 38), sendo que o principal objetivos delas é reunir as informações referentes aos fretes como os gastos gerados que podem acontecer durante o processo de frete.

Figura 37 - Estrutura da Tabela de Pedidos de clientes jurídicos

legal_orders	
id	String
order	String NN
created_at	DateTime NN
updated_at	DateTime NN
total_shipping_cost	Float NN
total_receivable	Float NN
total_tax_payable	Float NN
CreatedBy	users NN
created_by	String NN
UpdatedBy	users NN
updated_by	String NN
LegalContract	legal_Contracts NN
legal_contract_id	String NN
quote_table_id	String NN
QuoteTable	legal_client_quote NN
order_processing_id	String
OrderProcessing	order_procesing
carrier_id	String NN
CarrierCompany	carrier_companies NN
completed_orders_id	String
LegalClientCte	legal_client_cte
FreightExpenses	freight_expenses NN

Fonte: Autoria própria, 2024.

Figura 38 - Estrutura da Tabela de Pedidos para Clientes Físicos


physical_orders	
id 	String NN
order	String NN
updated_by	String NN
physical_customer_id	String NN
total_shipping_cost	Float NN
total_receivable	Float NN
total_tax_payable	Float NN
PhysicalCustomer	physical_customers NN
created_at 	DateTime NN
updated_at 	DateTime NN
CreatedBy	users NN
created_by	String NN
UpdatedBy	users NN
carrier_id	String NN
CarrierCompany	carrier_companies NN
order_processing_id	String
OrderProcessing	order_procesing
completedOrdersId	String
PhysicalCustomerQuoteTable	physical_customer_quote NN
quote_table_id	String NN
PhysicalCustomerCte	physical_customer_cte
FreightExpenses	freight_expenses NN

Fonte: Autoria própria, 2024.

4.9.12 Tabela de Incidentes

A tabela de Incidentes (Figura 39) serve para registrar todas as ocorrências que podem acontecer durante o processo de frete como por exemplo atrasar a entrega por causa que o veículo quebrou.

Figura 39 - Estrutura da Tabela de Ocorrências

Incident	
id 	String
description	String NN
date_incident	DateTime NN
order_process_id	String NN
OrderProcess	order_procesing NN
date_resolved	DateTime
CreatedBy	users NN
created_by	String NN
UpdatedBy	users NN
updated_by	String NN
created_at	DateTime NN
updated_at	DateTime NN

Fonte: Autoria própria, 2024.

4.9.13 Tabela de Processamento de Pedido

A tabela de Processamento de pedido (Figura 40) é responsável pelo frete em si, já que é nela que é relacionada os pedidos, veículo e motorista responsável pela entrega.

Figura 40 - Estrutura da Tabela de Processamento de Pedido

order_processing	
id	String
order_processing_number	String NN
vehicle_id	String NN
Vehicle	vehicles NN
total_distance	Float NN
total_spend_liters	Float NN
total_spending_money	Float NN
status	String NN
start_at	DateTime NN
end_at	DateTime
updated_at	DateTime NN
created_at	DateTime NN
created_by	String NN
updated_by	String NN
driver_id	String NN
OwnDriver	own_drivers
PhysicalCustomerOrder	physical_orders NN
LegalClientOrder	legal_orders NN
Incident	Incident NN
Manifest	Manifest NN

Fonte: Autoria própria, 2024.

5 RESULTADOS E DISCUSSÕES

Nesta sessão iremos apresentar os resultados do desenvolvimento do programa bem como as telas de criações e funcionalidades do programa.

5.1 COMPONENTES DE INTERFACE

Uma funcionalidade que terá na maioria das telas é um *data table* que permite os usuários atualizarem alguns campos mais simples da entidade e permite selecionar um determinado cadastro para então atualizá-lo ou excluí-lo, como demonstrado na Figura 41 a seguir:

Figura 41 - Data Table

ID	E-mail	Nome	Username	Permissão	Criado em
2a722bcd-ef7c-46ae-8f2...	ddrjudr@gmail.com	jdtjkd	dggh	USER	25 de junho de 2024 às
fc0ddb17-e998-4e9d-b8c...	gabrielrguedess@gmail.c...	Gabriel Guedes	gabrielguedess	ADMIN	8 de junho de 2024 às

Mostrando 1 de 1 página e 2 registros

Previous 1 Next

Fonte: Autoria própria, 2024.

Outro elemento de interface é as combos box que combina as funcionalidades de uma caixa de texto e um menu suspenso (*drop-down*), permitindo que o usuário selecione itens pré-cadastrados ou selecionados de forma mais rápida e dinâmica já que realiza um filtro conforme o usuário vai digitando o texto.

Figura 42 - Exemplo de Combo Box

Pessoa Física

Selecione a Pessoa Física

Q Pesquise a Pessoa Física...

12345678942 - MARCOS

12345678941 - Tssa

Fonte: Autoria própria, 2024.

Outros dois componentes fundamentais para aplicação, é o componente de criação de pessoa jurídica (Figura 43) e de criação de pessoa Física (Figura 44).

Figura 43 - Componente de criação de pessoa jurídica(empresa).

Não possui nenhuma Empresa? Crie abaixo

Nome Fantasia	Nome Corporativo	E-Mail	Celular	CNPJ
CEP	Cidade	Bairro	Rua	Numero
Estado Estado ▾	Registrado no Estado			

✕

Fonte: Autoria própria, 2024.

Figura 44 - Componente de Criação de Pessoa Física

Não possui nenhuma Pessoa Física? Crie abaixo

Nome	E-Mail	RG	Gênero Gênero ▾	Celular
Nacionalidade	CPF	Data de Nascimento 12/06/2024 📅	CEP	Cidade
Bairro	Rua	Numero	Estado Estado ▾	

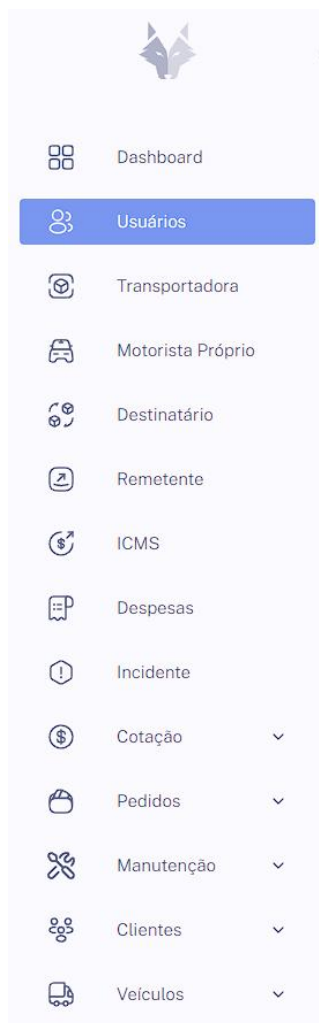
✕

Fonte: Autoria própria, 2024.

O menu lateral (Figura 45) permite o acesso as entidades do programa bem como o CRUD ⁶delas

Figura 45 - Componente de Interface - Menu Lateral

⁶ CRUD é um acrônimo que representa as quatro operações básicas de um banco de dados: Create (criar), Read (ler), Update (atualizar) e delete (excluir).

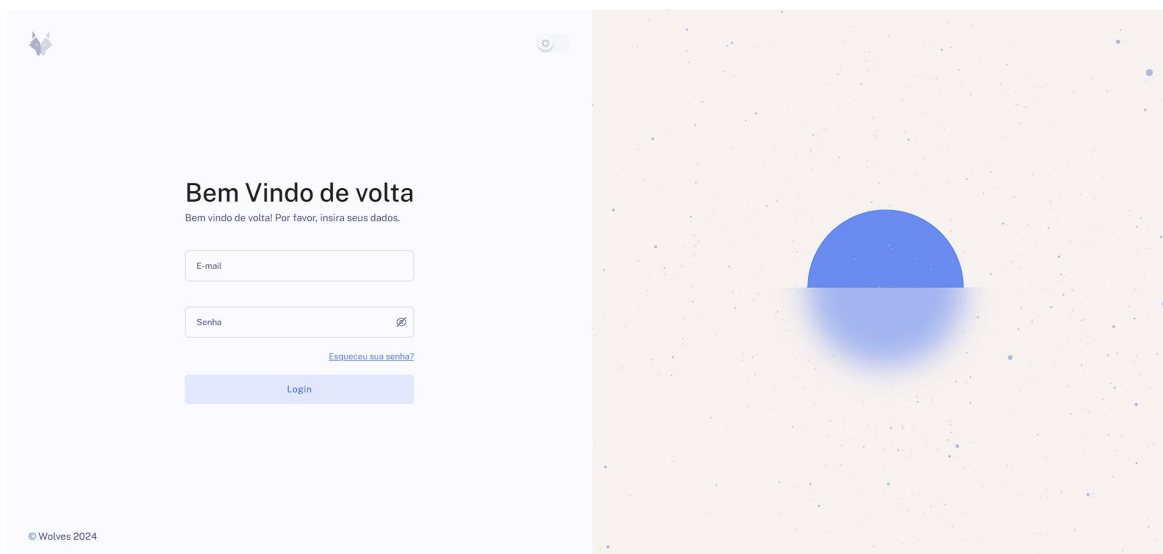


Fonte: Autoria própria, 2024.

5.2 INTERFACES GRAFICAS

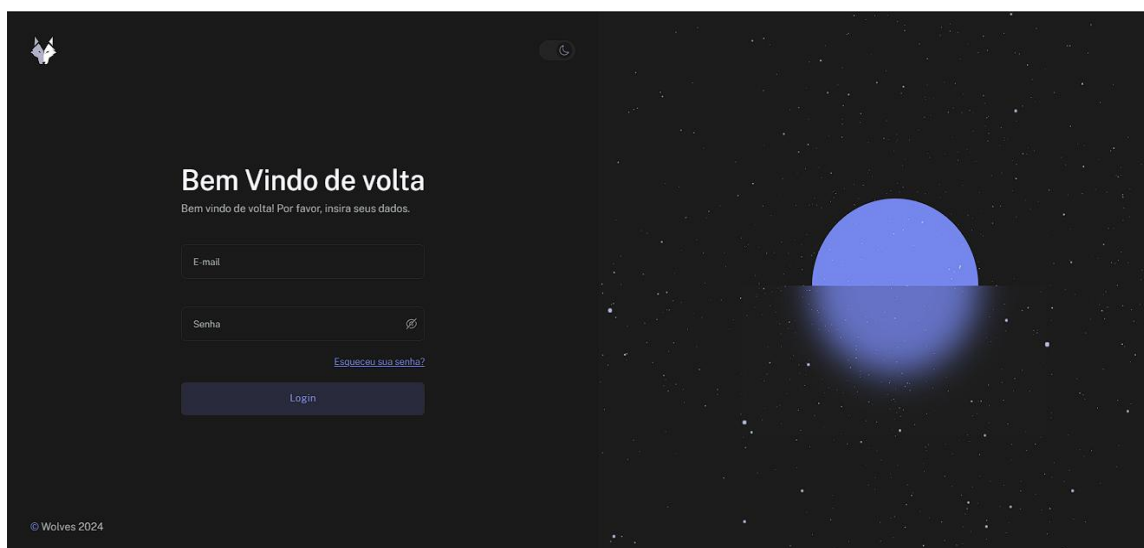
O software oferece dois modos de interface: *Light Mode* (modo claro), conforme demonstrado na Figura 46, e *Dark Mode* (modo escuro), mostrado na Figura 47. Esses modos permitem ao usuário escolher aquele que considera mais confortável. É importante destacar que a troca entre os modos pode ser feita a qualquer momento e em qualquer tela que o usuário desejar.

Figura 46 - Interface *Light mode*



Fonte: Aatoria própria, 2024.

Figura 47 - Interface *Dark Mode*



Fonte: Aatoria própria, 2024.

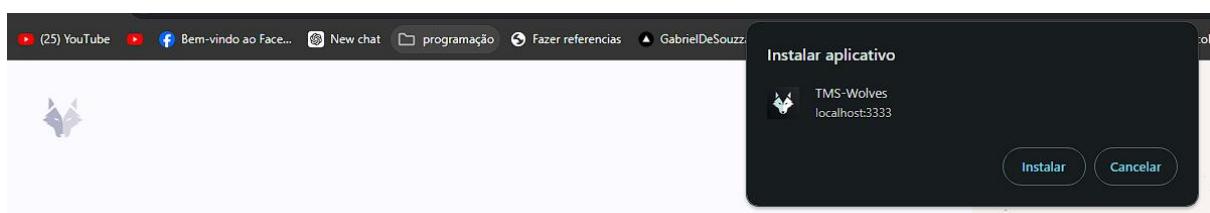
Além da versão web o projeto conta com uma versão mobile e desktop criada a partir do PWA (Progressive Web App) que é uma aplicação web que utiliza tecnologias modernas da web para oferecer uma experiência similar à de um aplicativo nativo. PWAs são projetados para serem rápidos, responsivos e funcionarem em diferentes dispositivos e navegadores. Caracterizam-se por:

- Responsividade: Adaptam-se a qualquer tamanho de tela.
- Conectividade: Funcionam mesmo sem conexão ou com conexão instável.

- Instalação: Podem ser instalados na tela inicial do dispositivo como um aplicativo tradicional.
- Atualizações: Atualizam-se automaticamente.

Para Instalar a versão desktop é muito simples, basta entrar no site da aplicação e clicar no botão apontado pela seta na Figura 48. Ao instalar a versão *desktop* o usuário tem uma maior facilidade de acesso e conforto visto que não vai mais precisar abrir o navegador para acessar o sistema.

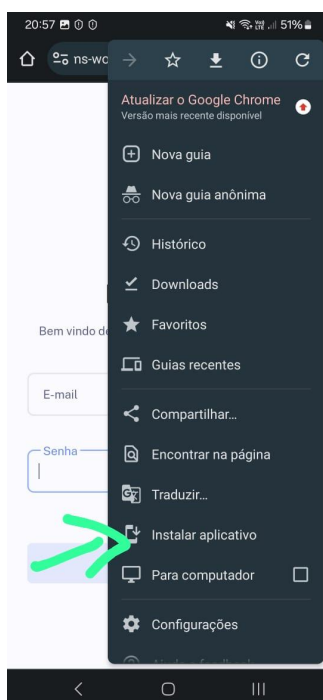
Figura 48 - Instalação da versão *desktop* do TMS



Fonte: Autoria própria, 2024.

Já para instalação da versão *mobile*, é necessário entrar no site e então apertar no três pontinho do seu navegador e ir à opção instalar aplicativo, como demonstrado na Figura 49 a seguir.

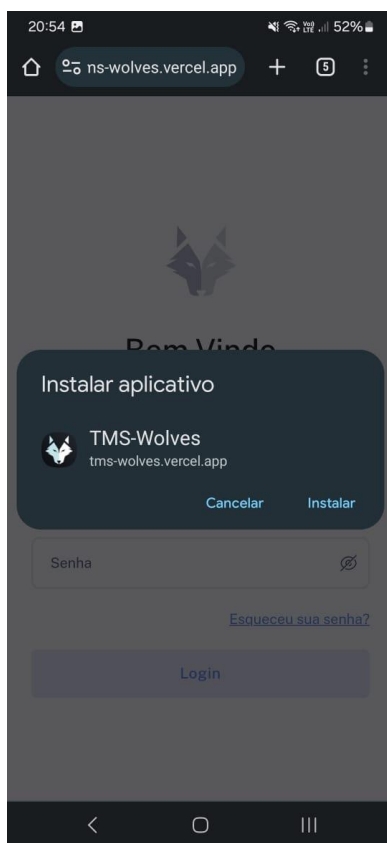
Figura 49 - Instalação da versão *mobile*



Fonte: Autoria própria, 2024.

Ao apertar na opção **instalar aplicativo** irá aparecer a tela de permissão de instalação, como mostrado na Figura 50:

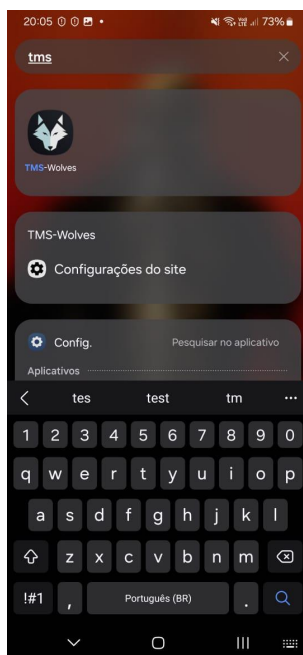
Figura 50 - Tela de Permissão de Instalação da versão mobile



Fonte: Autoria própria, 2024.

E ao apertar instalar, o aplicativo será instalado no seu celular, permitindo uma forma rápida, flexível e simples de acessar o sistema. Como Exemplificado na Figura 51, 52 e 53.

Figura 51 - Mostrando aplicativo instalado no celular



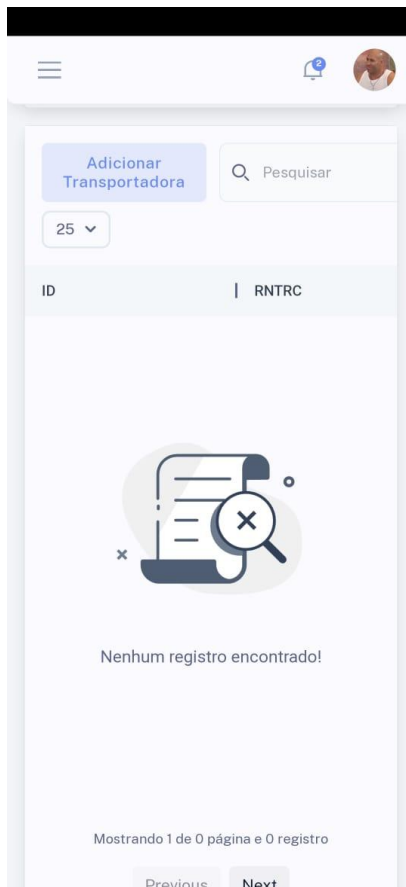
Fonte: Autoria própria, 2024.

Figura 52 - Tela de login da versão mobile



Fonte: Autoria própria, 2024.

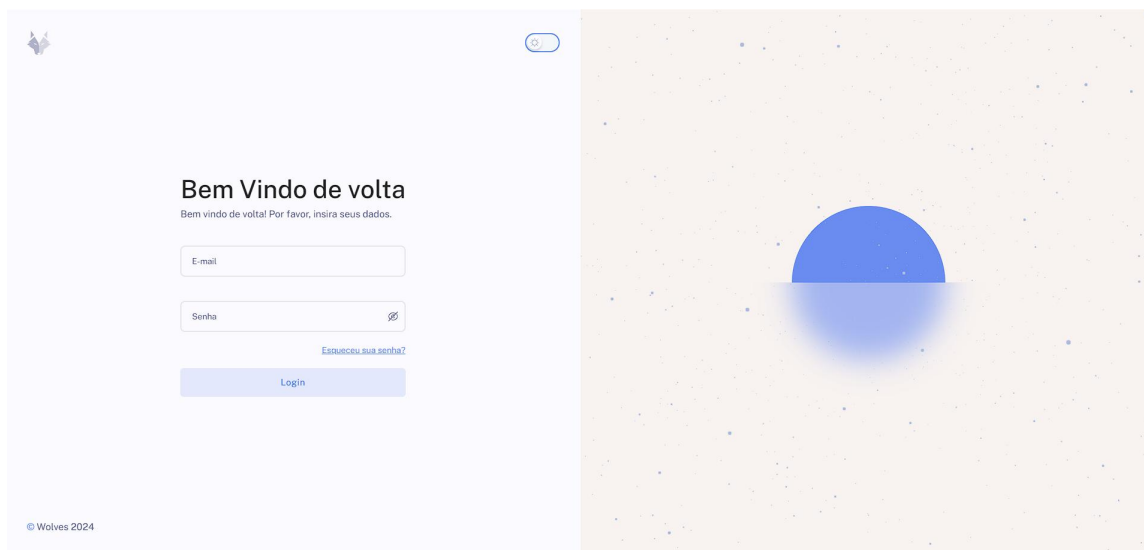
Figura 53 - Exemplo de tela da versão mobile



Fonte: Autoria própria, 2024.

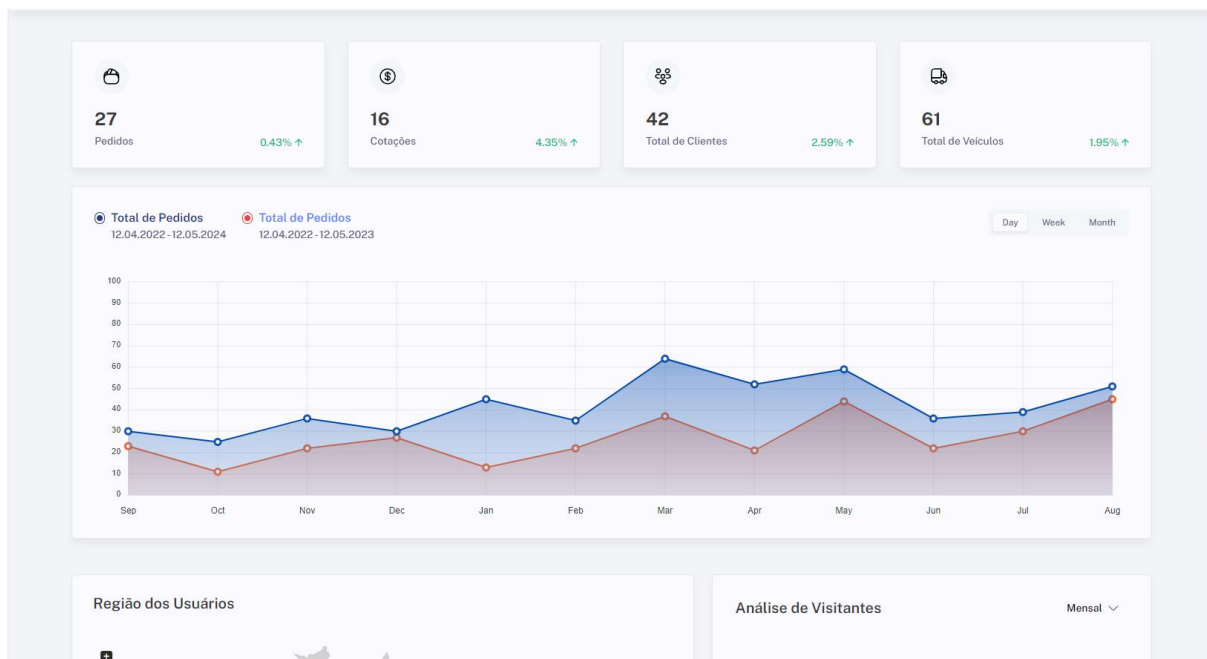
Assim que o usuário entra no site do TMS será solicitado o login, onde o usuário entrara com seu e-mail e senha para entrar no sistema, como demonstrado na Figura 54 a seguir:

Figura 54 - Tela de Login



Fonte: Autoria própria, 2024.

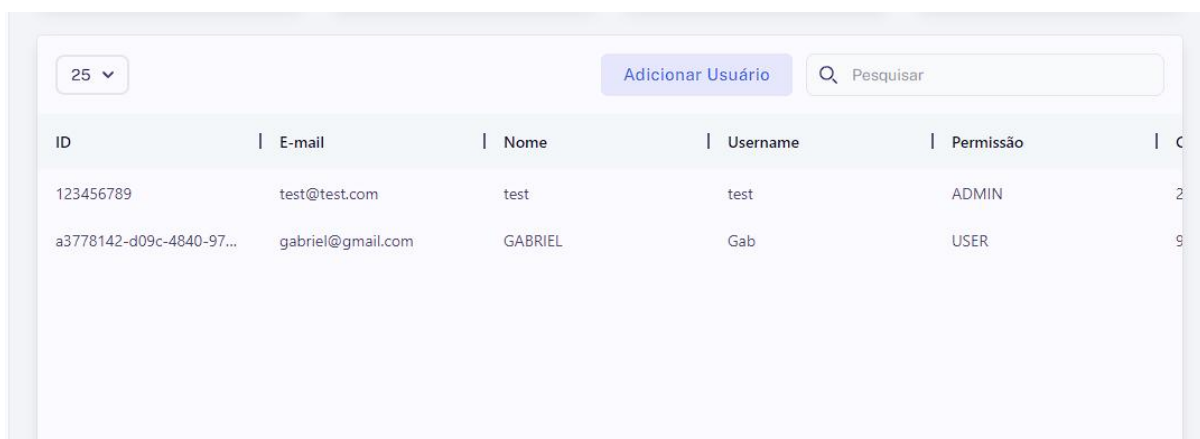
Assim que o usuário entrar no sistema, verá o dashboard (Figura 55), onde estarão as principais informações, como a quantidade de pedidos já realizados e os estados com maior demanda, entre outros.

Figura 55 - Tela de *Dash Board*

Fonte: Autoria própria, 2024.

A próxima tela é a de usuários, acessível apenas para administradores. Nela, o usuário poderá visualizar (Figura 56), atualizar e criar usuários (Figura 57) que utilizarão o sistema.

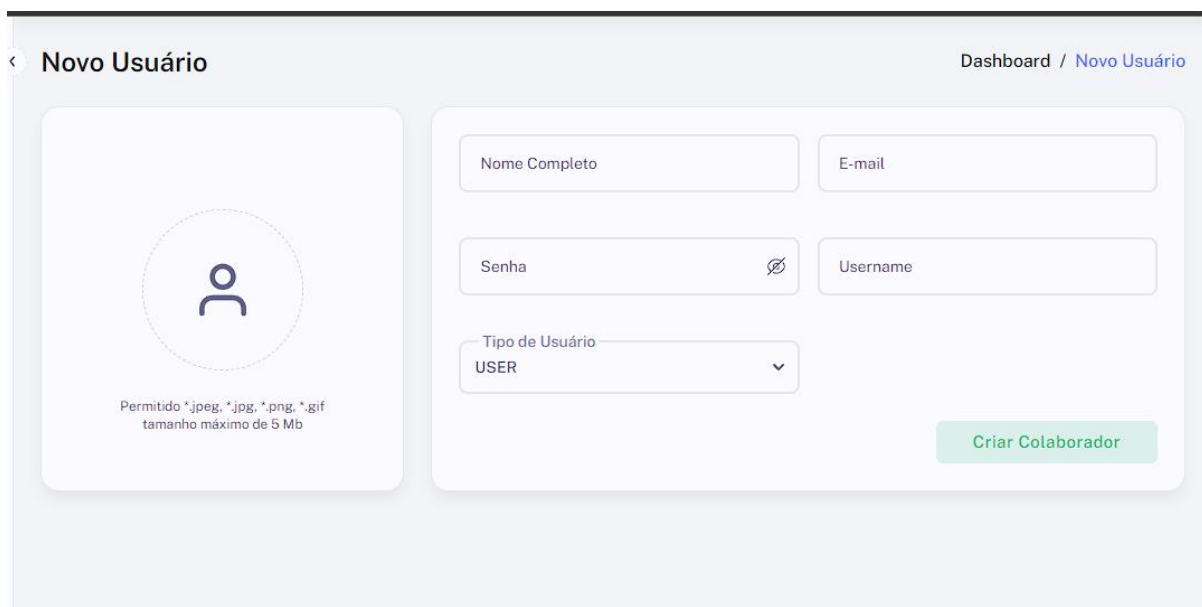
Figura 56 - Tela de Controle de Usuários



ID	E-mail	Nome	Username	Permissão	C
123456789	test@test.com	test	test	ADMIN	2
a3778142-d09c-4840-97...	gabriel@gmail.com	GABRIEL	Gab	USER	9

Fonte: Autoria própria, 2024.

Figura 57 - Tela de Criação/Atualização de usuários



Novo Usuário Dashboard / Novo Usuário

Nome Completo E-mail

Senha Username

Tipo de Usuário

Permitido *.jpeg, *.jpg, *.png, *.gif
tamanho máximo de 5 Mb

Fonte: Autoria própria, 2024.

O Próxima tela de cria/atualização é a de transportadora onde o usuário pode selecionar uma empresa já cadastrada no sistema ou criar uma nova como demonstrado na Figura 58.

Figura 58 - Tela de Criação/Atualização de Transportadora

Nova Transportadora Dashboard / Nova Transportadora

RNTRC Empresa

Não possui nenhuma Empresa? Crie abaixo

+

[Criar Transportadora](#)

Fonte: Autoria própria, 2024.

Cadastrada a transportadora é de suma importância cadastrar os motoristas dessa transportadora. A Figura 59 mostra a tela de cadastro/atualização de motorista.

Figura 59 - Exemplo de tela da versão mobile

Novo Motorista Próprio Dashboard / Novo Motorista Próprio

CNH Categoria da CNH Expiração de CNH Veiculo da Empresa? Curso MOOP

Pessoa Física

Não possui nenhuma Pessoa Física? Crie abaixo

+

[Criar Motorista](#)

Fonte: Autoria própria, 2024.

Ainda temos um conjunto de cadastros que permitem criar um controle sólido sobre os veículos. Primeiramente, temos a tela de criação/atualização de marca de veículo (Figura 60). Em seguida, há a tela de carroceria (Figura 61), onde o usuário pode criar uma carroceria. Após isso, o usuário pode acessar a tela de tipo de veículo (Figura 62), na qual ele cria o tipo e seleciona as carrocerias associadas a esse tipo. Em seguida, há a tela de modelo (Figura 63), onde o usuário insere os dados do modelo e o associa a um tipo de veículo Figura (64). Por fim, o veículo é criado e associado a uma transportadora.

Figura 60 - Tela de Criação/Atualização de marca de veículo

A interface apresenta o título "Nova Marca" no canto superior esquerdo e o caminho de navegação "Dashboard / Nova Marca" no canto superior direito. O formulário principal contém um campo de entrada de texto rotulado "Nome". No canto inferior direito do formulário, há um botão verde com o texto "Criar Marca de Veículo".

Fonte: Autoria própria, 2024.

Figura 61 - Tela de Criação/Atualização de Carroceria

A interface apresenta o título "Nova Carroceria" no canto superior esquerdo e o caminho de navegação "Dashboard / Nova Carroceria" no canto superior direito. O formulário principal contém quatro campos de entrada de texto rotulados "Nome", "Massa", "Eixos" e "Volume". No canto inferior direito do formulário, há um botão verde com o texto "Criar Carroceria de Veículo".

Fonte: Autoria própria, 2024.

Figura 62 - Tela de Criação/Atualização de Tipo de veículo

A interface apresenta o título "Novo Tipo de Veículo" no canto superior esquerdo e o caminho de navegação "Dashboard / Novo Tipo de Veículo" no canto superior direito. O formulário principal contém um campo de entrada de texto rotulado "Nome", um menu suspenso rotulado "Carroceria" com o valor "Não" selecionado, e um campo de entrada de texto rotulado "Carroceria" com o texto "Selecione a Carroceria" dentro dele. No canto inferior direito do formulário, há um botão verde com o texto "Criar Tipo de Veículo".

Fonte: Autoria própria, 2024.

Figura 63 - Tela de Criação/Atualização de modelo de veículo

The screenshot shows a web form titled "Novo Modelo" with a breadcrumb "Dashboard / Novo Modelo". The form contains several input fields: "Nome", "Capacidade Maxima", "Maxima por Eixo", "Peso", and "Eixos" in the top row; and "Marca" (with subtext "Selecione a Marca") and "Tipo" (with subtext "Selecione a Tipo") in the bottom row. A green button labeled "Criar Modelo de Veiculo" is positioned at the bottom right of the form area.

Fonte: Autoria própria, 2024.

Figura 64 - Tela de Criação/Atualização de Veículo

The screenshot shows a web form titled "Novo Veiculo de Empresa" with a breadcrumb "Dashboard / Novo Veiculo de Empresa". The form contains two selection fields: "Transportadora" (with subtext "Selecione a Transportadora") and "Veiculo" (with subtext "Selecione a Veiculo"). A green button labeled "Criar Veiculo de Empresa" is located on the right side of the form. Below the form, there is a message "Não possui nenhuma Veiculo? Crie abaixo" and a circular button with a plus sign (+).

Fonte: Autoria própria, 2024.

Na tela de cadastro de destinatário (Figura 65 e 66) e remetente (Figura 67 e 68), o usuário pode optar por cadastrar uma pessoa física, selecionando um CPF já registrado no sistema ou criando um cadastro. Alternativamente, o usuário pode escolher cadastrar uma empresa, inserindo um CNPJ existente ou criando uma empresa.

Figura 65 - Cadastro de destinatário com CPF já criado no sistema

The screenshot shows the 'Novo Destinatário' form. At the top left, there is a dropdown menu for 'Física ou Jurídica?' with 'Física' selected. To its right is a button labeled 'Pessoa Física' with the text 'Selecione a Pessoa Física' below it. Below this button is a search input field with a magnifying glass icon and the placeholder text 'Pesquise a Pessoa Física...'. A dropdown menu is open below the search field, displaying two results: '12345678942 - MARCOS' and '12345678941 - Tssa'. In the center of the form is a large circular button with a plus sign '+'. At the bottom right, there is a green button labeled 'Criar Remetente'. The breadcrumb 'Dashboard / Novo Destinatário' is visible in the top right corner.

Fonte: Autoria própria, 2024.

Figura 66 - Criação de destinatário com CNPJ já cadastrado sistema

The screenshot shows the 'Novo Destinatário' form. At the top left, there is a dropdown menu for 'Física ou Jurídica?' with 'Física' selected. To its right is a button labeled 'Pessoa Física' with the text 'Selecione a Pessoa Física' below it. Below this button is a search input field with a magnifying glass icon and the placeholder text 'Pesquise a Pessoa Física...'. Below the search field is the text 'Não possui nenhuma Pessoa Física? Crie abaixo'. The form contains several input fields: 'Nome', 'E-Mail', 'RG', 'Gênero' (with a dropdown menu showing 'Gênero'), 'Celular', 'Nacionalidade', 'CPF', 'Data de Nascimento' (with a calendar icon and the value '11/06/2024'), 'CEP', 'Cidade', 'Bairro', 'Rua', 'Numero', and 'Estado' (with a dropdown menu showing 'Estado'). In the center of the form is a large circular button with a plus sign '+'. At the bottom right, there is a green button labeled 'Criar Remetente'. The breadcrumb 'Dashboard / Novo Destinatário' is visible in the top right corner.

Fonte: Autoria própria, 2024.

Figura 67 - Criação de destinatário com CNPJ já cadastrado no sistema

A interface de usuário para a criação de um destinatário. No topo, há o título "Novo Destinatário" e o caminho de navegação "Dashboard / Novo Destinatário". O formulário contém:

- Um menu suspenso "Física ou Jurídica?" com a opção "Jurídica" selecionada.
- Um campo de busca "Empresa" com o texto "Selecione a Empresa".
- Uma barra de pesquisa "Pesquise a Empresa..." com uma lupa.
- Uma lista de sugestões de empresas com seus respectivos CNPJs: "54769765000110 - Ssada", "14512451145112 - URSNIHO CARILHOSOS", "54765769000110 - WOLFE" e "14714725849635 - VOARR LTDA".
- Um botão "Criar Remetente" no canto inferior direito.
- Um ícone de "+" no centro da tela.

Fonte: Autoria própria, 2024.

Figura 68 - Criação de empresa na tela de destinatário

A interface de usuário para a criação de um destinatário, focada na criação de uma nova empresa. No topo, há o título "Novo Destinatário" e o caminho de navegação "Dashboard / Novo Destinatário". O formulário contém:

- Um menu suspenso "Física ou Jurídica?" com a opção "Jurídica" selecionada.
- Um campo de busca "Empresa" com o texto "Selecione a Empresa".
- Uma barra de pesquisa "Pesquise a Empresa..." com uma lupa.
- Um texto "Não possui nenhuma Empresa? Crie abaixo".
- Campos de entrada para: "Nome Fantasia", "Nome Corporativo", "E-Mail", "Celular", "CNPJ", "CEP", "Cidade", "Bairro", "Rua", "Número".
- Um menu suspenso "Estado" com a opção "Estado" selecionada.
- Um campo de entrada "Registrado no Estado".
- Um botão "Criar Remetente" no canto inferior direito.
- Um ícone de "X" no centro da tela.

Fonte: Autoria própria, 2024.

Além do destinatário tem a criação de clientes, eles que são responsáveis por realizar a cotação do pedido. Sendo que pode ser criado cliente jurídico (Figura 69 e cliente físico (Figura 70).

Figura 69 - Tela de Criação/Atualização de Jurídico

The screenshot shows a web form titled "Novo Cliente Jurídico" with a breadcrumb "Dashboard / Novo Cliente Jurídico". The form contains a "Ramo" dropdown menu and an "Empresa" selection box with the text "Selecione a Empresa". Below these fields, there is a message: "Não possui nenhuma Empresa? Crie abaixo" and a large circular button with a "+" sign. A green "Criar Cliente" button is located at the bottom right of the form.

Fonte: Autoria própria, 2024.

Figura 70 - Tela de Criação/Atualização de cliente Jurídico

The screenshot shows a web form titled "Novo Cliente Físico" with a breadcrumb "Dashboard / Novo Cliente Físico". The form contains a "Ramo" dropdown menu and a "Pessoa Física" selection box with the text "Selecione a Pessoa Física". Below these fields, there is a message: "Não possui nenhuma Pessoa Física? Crie abaixo" and a large circular button with a "+" sign. A green "Criar Cliente" button is located at the bottom right of the form.

Fonte: Autoria própria, 2024.

A tela de cadastro e atualização do ICMS (Figura 71) o usuário poderá cadastrar, alterar e excluir os dados de ICMS que serão usados para calcular os pedidos.

Figura 71 - Tela de Cadastro/Atualização de ICMS



Fonte: Autoria própria, 2024.

Enquanto a tela de despesa de frete (Figura 72), assim como o nome sugeri serve para cadastrar os custos de frete como pagamento de pedágio.

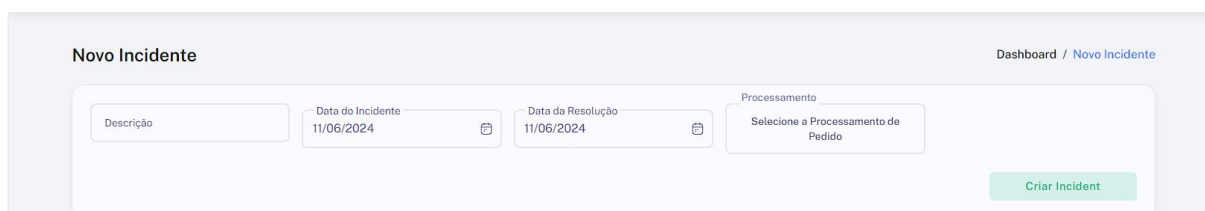
Figura 72 - Tela de Criação/Atualização de Despesa de Frete



Fonte: Autoria própria, 2024.

Já a tela de incidentes (Figura 73) serve para cadastrar/atualizar sinistros que podem ocorrer durante uma viagem.

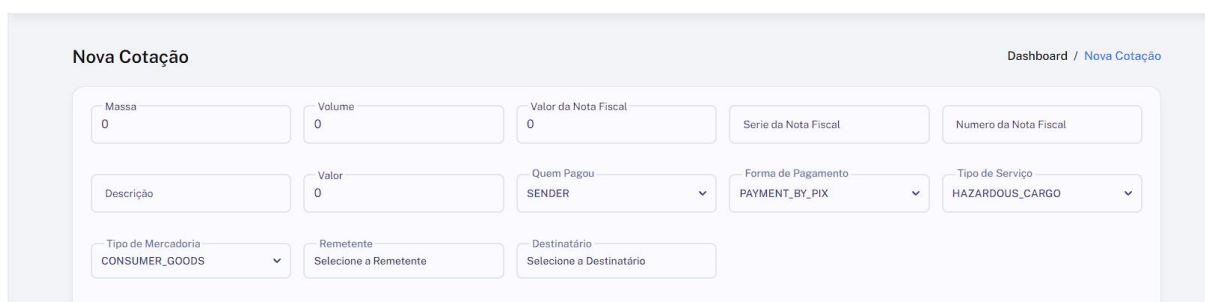
Figura 73 - Tela de Cadastro/Atualização de incidentes de frete (Sinistro)



Fonte: Autoria própria, 2024.

A Próxima tela a ser apresentada é a da cotação que apesar de ser uma para cliente jurídico e outra para cliente físico, os campos são o mesmo, a diferença é que o cliente jurídico faz a cotação através de um contrato e o cliente físico faz de forma direta. Inicialmente o usuário de deve digitar os dados relacionado a cotação (Figura 74), e então ele pode digitar o endereço de origem do frete (Figura 75) e por fim o endereço de destino do frete (Figura 76).

Figura 74 - Tela de Cadastro/Atualização de cotação – Dados da cotação

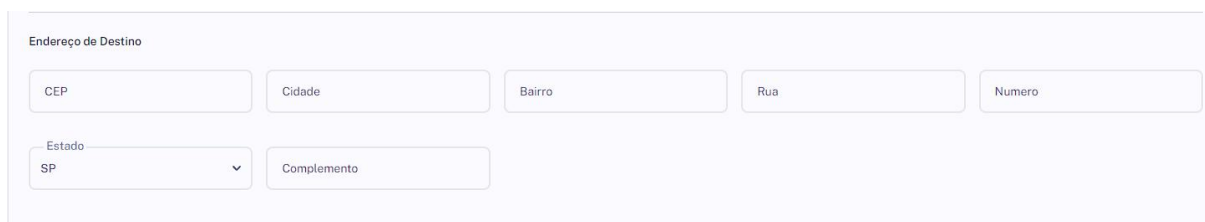


The screenshot shows a form titled "Nova Cotação" with a breadcrumb "Dashboard / Nova Cotação". The form contains several input fields and dropdown menus:

- Massa: 0
- Volume: 0
- Valor da Nota Fiscal: 0
- Serie da Nota Fiscal
- Numero da Nota Fiscal
- Descrição
- Valor: 0
- Quem Pagou: SENDER
- Forma de Pagamento: PAYMENT_BY_PIX
- Tipo de Serviço: HAZARDOUS_CARGO
- Tipo de Mercadoria: CONSUMER_GOODS
- Remetente: Selecione a Remetente
- Destinatário: Selecione a Destinatário

Fonte: Autoria própria, 2024.

Figura 75 - Tela de Cadastro/Atualização de Cotação – Dados da origem



The screenshot shows a form titled "Endereço de Destino" with the following input fields:

- CEP
- Cidade
- Bairro
- Rua
- Numero
- Estado: SP
- Complemento

Fonte: Autoria própria, 2024.

Figura 76 - Tela de Cadastro/Atualização de Cotação – Dados de Destino



The screenshot shows the same "Endereço de Destino" form as in Figure 75, but with a green "Criar Cotação" button located at the bottom right corner.

Fonte: Autoria própria, 2024.

Após realizar a cotação, o usuário pode iniciar o pedido, que pode ser físico (Figura 77) ou jurídico (Figura 78). A principal diferença entre eles é que o pedido do cliente jurídico requer um contrato para ser criado, enquanto o pedido do cliente físico pode ser feito diretamente.

Figura 77 - Tela de Cadastro/Atualização de Pedido de Cliente Físico

Novo Pedido Físico Dashboard / Novo Pedido Físico

Transportadora
Seleção a Transportadora

Client Físico
Seleção a Client Físico

Cotação Física
Seleção a Cotação Física

Despesas

Nome

Valor
0

+

Criar Pedido Físico

Fonte: Autoria própria, 2024.

Figura 78 - Tela de Criação/Atualização de Pedido de Cliente Jurídico

Novo Pedido Jurídico Dashboard / Novo Pedido Jurídico

Transportadora
Seleção a Transportadora

Contrato Jurídico
Seleção a Contrato Jurídico

Cotação Jurídica
Seleção a Cotação Jurídica

Despesas

Nome

Valor
0

+

Criar Pedido Jurídico

Fonte: Autoria própria, 2024.

O como mencionado para fazer o pedido do cliente jurídico e é feito através da tela mostrada na Figura 79:

Figura 79 - Tela de Criação/Atualização de Contrato de Cliente jurídico

The screenshot shows a web form titled "Novo Contrato" with a breadcrumb "Dashboard / Novo Contrato". The form contains several input fields: "Observações", "Condições de Entrega", "Data Efetiva" (with the value "12/06/2024" and a calendar icon), "Transportadora" (with a dropdown menu and the text "Selecione a Transportadora"), and "Cliente Jurídico" (with a dropdown menu and the text "Selecione a Cliente Jurídico"). A green "Criar Contrato" button is located at the bottom right of the form.

Fonte: Autoria própria, 2024.

A tela de Manutenção (Figura 80) tem como objeto registrar das manutenções referentes aos veículos da empresa, junto ao tipo da manutenção, exemplo troca preventiva de pneus.

Figura 80 - Tela de Criação/Atualização de Manutenção

The screenshot shows a web form titled "Nova manutenção" with a breadcrumb "Dashboard / Nova manutenção". The form contains three input fields: "Tipo de Manutenção" (with a dropdown menu and the text "Selecione a Tipo de Manutenção"), "Veiculo" (with a dropdown menu and the text "Selecione o Veiculo"), and "Empresa de Manutenção" (with a dropdown menu and the text "Selecione o Empresa de Manutenção"). A green "Criar Manutenção" button is located at the bottom right of the form.

Fonte: Autoria própria, 2024.

Além de castrar a manutenção é necessária cadastrar a empresa que vai realizar a manutenção (Figura 81).

Figura 81 - Tela de Criação/Atualização de empresa de manutenção

The screenshot shows a web form titled "Nova Empresa de Manutenção". At the top right, there is a breadcrumb trail: "Dashboard / Nova Empresa de Manutenção". The form contains two input fields: "Manutenção Especializada Em" and "Empresa" (with a sub-label "Selecione a Empresa"). Below these fields, a message reads "Não possui nenhuma Empresa? Crie abaixo" centered above a large circular button with a plus sign. In the bottom right corner of the form, there is a green button labeled "Criar Empresa".

Fonte: Autoria própria, 2024.

A tela de cadastro de processamento de pedido (Figura 82) tem como objetivo registrar o início do frete propriamente dito, vinculando os pedidos (tanto físicos quanto jurídicos) ao motorista e ao veículo, incluindo o status e o início e fim do frete.

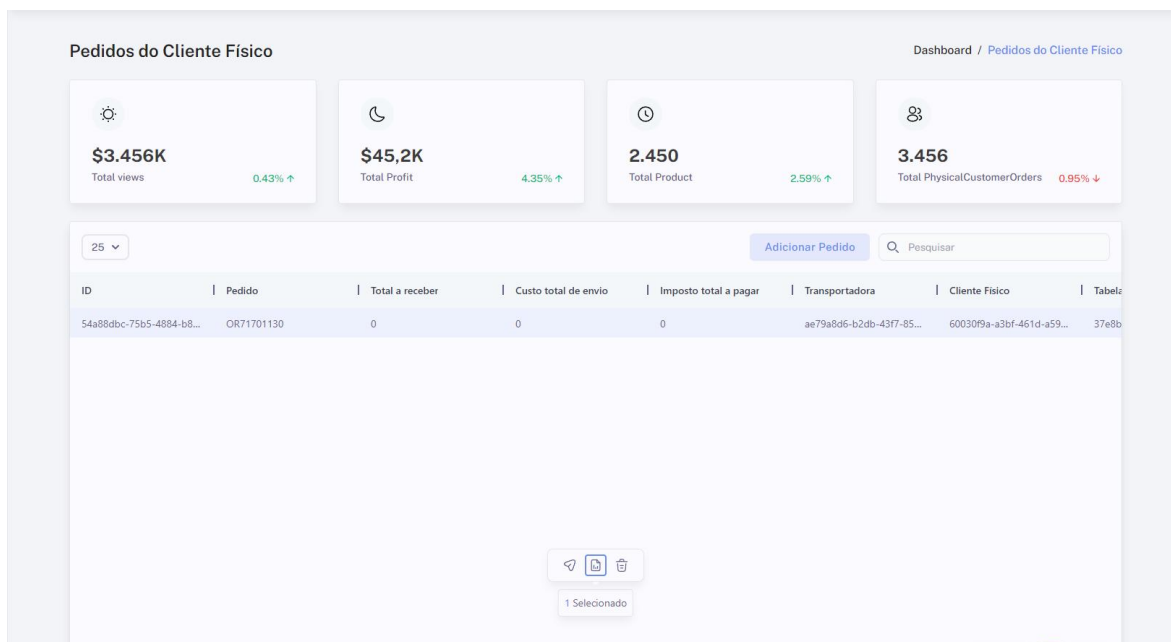
Figura 82 - Tela de cadastro/atualização processamento de pedido

The screenshot shows a web form titled "Novo Processamento". At the top right, there is a breadcrumb trail: "Dashboard / Novo Processamento". The form contains several input fields: "Distancia Total", "Total de Litros Gastos", "Total de Gastos", "Data de Inicio" (with value "12/06/2024" and a calendar icon), and "Data de Finalização" (with value "12/06/2024" and a calendar icon). Below these are three more fields: "Status" (with a dropdown menu showing "CREATED"), "Veiculo" (with sub-label "Selecione a Veiculo"), and "Motorista" (with sub-label "Selecione o Motorista"). The form is divided into two sections for selecting requests: "Selecione os Pedidos Físicos para esse processamento" and "Selecione os Pedidos Jurídicos para esse processamento". Each section has a sub-label "Pedido Físico" or "Pedido Jurídico" and a text input field "Selecione o Pedido Físico" or "Selecione o Pedido Jurídico", followed by a large circular button with a plus sign. In the bottom right corner of the form, there is a green button labeled "Criar Processamento".

Fonte: Autoria própria, 2024.

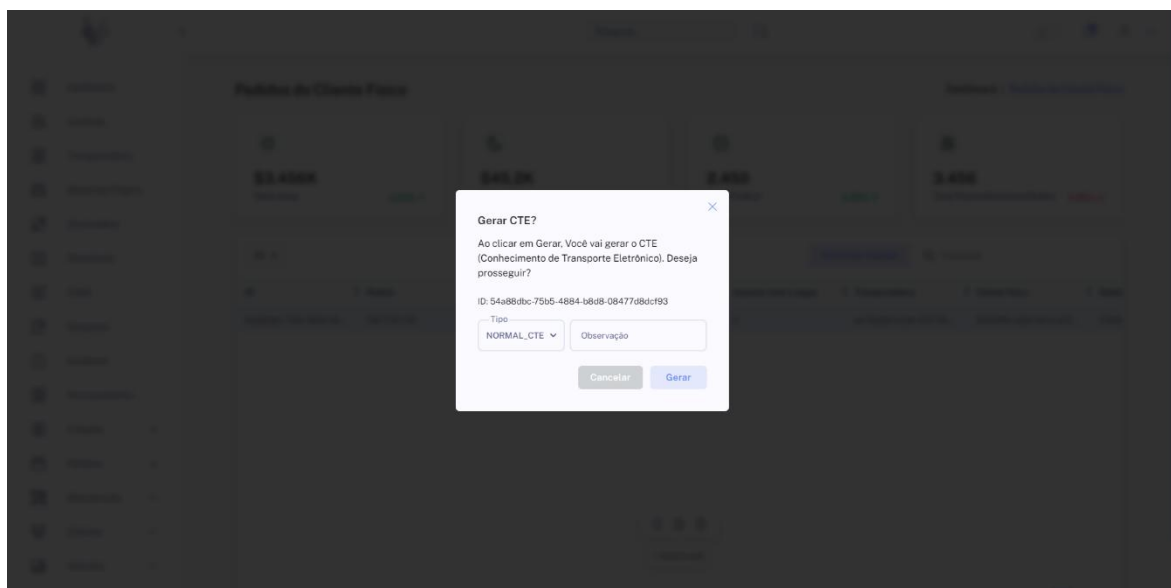
E, por fim, a emissão do CTE (Figura 83) é realizada no dashboard de pedidos, selecionando a linha do pedido para o qual você deseja gerar o CTE. Em seguida, clique no botão “1 selecionado” e depois no ícone de arquivo localizado no centro para abrir um modal com as opções de geração de CTE (Figura 84) por fim será gerado um arquivo PDF do documento (Figura 85).

Figura 83 - Tela de menu de opções do pedido




Fonte: Autoria própria, 2024.

Figura 84 - Modal geração de CTE



Fonte: Autoria própria, 2024.

Figura 85 - CTE gerado

	Transportadora: WOLFE CNPJ: 54765769000110 Inscrição Estadual: 147851254741 Endereço: Rua: sagfd Número: Bairro: dasdsa CEP: 07142090 Cidade: Guarulhos Estado: SP Telefone: 119801741171		D A C T E		AUTORIZAÇÃO 27/05/2024
	TIPO DO CTE Normal		TIPO DO SERVIÇO Normal		NATURUREZA DA PRESTAÇÃO Transporte Rodoviário de Cargas
Origem: SP / SP	Destino: GUA / SP	Emitido por WOLFE		Documento Auxiliar do Conhecimento de Transporte Eletrônico	
REMETENTE: Tssa END: dfdsd 1071 asdsad MUN: GUARULHOS - SP CEP: 14725836 FONE: 1452141474		DESTINATÁRIO: MARCOS END: dfdsd 1071 asdsad MUN: GUARULHOS - SP CEP: 14725836 CPF: 12345678942 FONE: 1452141474		SERIE 1451	
RECEBEDOR: URSNIHO CARILHOSOS LTDA END: sagfd 2515 dasdsa MUN: Guarulhos - SP CEP: 07142090 CNPJ: 14512451145112 IE: 147851254742 FONE: 119801741171		TOMADOR: MARCOS END: dfdsd 1071 asdsad MUN: GUARULHOS - SP CEP: 14725836 CPF: 12345678942 FONE: 1452141474		NUMERO 145	
Observações: SdpsdjoaSaas		NR: CTE15020023		MODELO 57	
TOMADOR SERVIÇO: RECIPIENT		COBRAR PAGAMENTO A VISTA		Nº PROTOCOLO A fazer	
				Chave de acesso para consulta de autenticidade no site www.cte.fazenda.gov.br 0010101131000053340200011501104013700034	
				COMONENTES DE FRETE(R\$)	
				GASOLINA 60	
				PROD PREDOMIN 60	
				ESPECIE: Containerizadas	
				VALOR MERCADORIA(R\$) 541	
				QTDE PARES/VOLUMES 5	
				CUBAG(m3) 841	
				PESO CALCULO (Kg) 145,11	
				ICMS(R\$)	
				SITUAÇÃO TRIBUTARIA	
				BASE CALCULO 60	
				ALIQ DIFAL/ICMS(%) 12	
				VALOR ICMS 7,2	
				DIFAL ICMS ORIG/DEST 0	
				CRED PRES/ICMS ST 0	
				FRETE TOTAL(R\$) 60	
				VALOR A RECEBER(R\$) 72,75	
				DESTAQUE DE TRIBUTOS (Lei 12.741/2012) - Em R\$	
				PIX	
				ICMS/ISS 7,2 PIS: COFINS: 4,56 TOTAL: 12,75	
				CHAVES NF-E/CTE	

Fonte: Autoria própria, 2024.

6 CONSIDERAÇÕES FINAIS

Ao final do desenvolvimento deste projeto, foram alcançados importantes avanços na concepção e implementação de um *Transportation Management System* (TMS) visto que tivemos que estudar a fundo o assunto e entender o funcionamento. Este sistema foi projetado para servir de ferramenta para os professores explicarem na prática, para os alunos, de como o TMS funciona, fornecendo ferramentas eficazes para o acompanhamento e controle de operações de frete.

As principais funcionalidades implementadas incluem a gestão de pedidos, controle de veículos, monitoramento de entregas e uma interface intuitiva para o usuário. A flexibilidade do sistema é assegurada pela utilização de uma arquitetura cliente-servidor, permitindo uma clara separação entre o *front-end* e o *back-end*, o que facilita a manutenção e escalabilidade da aplicação.

O uso de tecnologias modernas, como o *GraphQL* para a construção da API, possibilitou um desempenho eficiente e uma comunicação ágil entre o cliente e o servidor. Este modelo arquitetônico trouxe benefícios significativos, como a redução de problemas de *over-fetching* e *under-fetching*, comuns em outras abordagens como *REST*. A capacidade de filtrar e recuperar dados de forma precisa e personalizada melhora significativamente a experiência do usuário e de desenvolvimento das interfaces gráficas bem como a eficiência do sistema.

Adicionalmente, as interfaces gráficas projetadas oferecem ao usuário a opção de alternar entre modos claro e escuro, promovendo uma melhor usabilidade conforme as preferências individuais. A integração de componentes de interface dinâmicos, como *data tables* e *combo boxes*, também contribui para uma navegação mais fluida e intuitiva. E a possibilidade de instalar o sistema no computador ou celular deixa o sistema mais adaptativo e responsivo as necessidades dos usuários além de ter um certo papel de inclusão, visto que em uma turma de alunos muito grande, onde não tem computadores para todos, os alunos poderão usar o celular para participar da aula e ter a mesma experiência que os demais.

Este trabalho não apenas apresentou um TMS funcional, mas também explorou as melhores práticas de desenvolvimento de software e gestão de dados. A adoção de uma metodologia de desenvolvimento ágil permitiu ajustes contínuos e melhorias incrementais, garantindo a entrega de um produto de alta qualidade.

Em resumo, o projeto do TMS cumpriu seus objetivos principais, oferecendo uma solução tecnológica eficiente para criação de um TMS. As lições aprendidas e as tecnologias aplicadas neste desenvolvimento abrem caminhos para futuras inovações e aprimoramentos, consolidando a importância de sistemas de informação bem projetados para o aprendizado mais eficaz dos alunos.

REFERÊNCIAS

AMAZON AWS. **Qual é a diferença entre o MySQL e o PostgreSQL?** Disponível em: <https://aws.amazon.com/pt/compare/the-difference-between-mysql-vs-postgresql/#:~:text=O%20PostgreSQL%20oferece%20suporte%20aos,procedimentos%20armazenados%20em%20v%C3%A1rias%20linguagens.&text=O%20MySQL%20oferece%20suporte%20a,e%20hora%2C%20especiais%20e%20JSON>. Acesso em: 27 fev. 2024.

C. MARTIN, Robert *et al.* **Código Limpo: Habilidades Práticas do Agile Software**. Rio de Janeiro: Alta Books Editora, 2011. 456 p. ISBN 978-85-508-1148-2.

CARVALHO, L. Análise das potencialidades e vantagens do uso da simulação computacional em operações logísticas complexas, como ferramenta de auxílio à tomada de decisões: estudo de caso em uma organização industrial. Dissertação (Mestrado) – Universidade Federal da Bahia: Salvador, 2006.

CAXITO, Fabiano. **Logística: um enfoque prático**. São Paulo: Editora Saraiva, 2011. 328p.

DIAS, MARCO AURÉLIO. **Logística, Transporte e Infraestrutura: Armazenagem Operador Logístico Gestão via TI Multimodal**. SÃO PAULO: EDITORA ATLAS S.A, 2012. 354 p. ISBN 978-85-224-7427-1.

EVANS, Eric. **Domain-Driven Design: Tackling Complexity in the Heart of Software**. 1. ed. Boston: Addison-Wesley, 2003. ISBN 978-0321125217. Disponível em: [https://github.com/KAYOKG/BibliotecaDev/blob/main/LivrosDev/Domain%20Driven%20Design%20Rapido%20-%20Autor%20\(Eric%20Evans\).pdf](https://github.com/KAYOKG/BibliotecaDev/blob/main/LivrosDev/Domain%20Driven%20Design%20Rapido%20-%20Autor%20(Eric%20Evans).pdf). Acesso em: 28 fev. 2024.

FOWLER, Martin. **A website on building software effectively**. Disponível em: <https://martinfowler.com/>. Acesso em: 27 fev. 2024.

https://www.ilos.com.br/DOWNLOADS/PANORAMAS/Nova_Brochura%20_CustosLog2017.pdf. Acesso em: 3 abr. 2023

ILOS (RJ). **Custos Logísticos no Brasil**. Rio de Janeiro: Panorama ILOS, 2017. 15 disponível em:

INVERSION of Control Containers and the Dependency Injection pattern. [S. l.], 23 jan. 2004. Disponível em: <https://martinfowler.com/articles/injection.html>. Acesso em: 2 mar. 2024.

LAUDON, Kenneth C.; LAUDON, Jane Price Laudo. **Sistemas de Informação Gerenciais**. 7. ed. São Paulo: Person, 2007. 478 p. ISBN 9788576050896.

LIMA, Welton Dias de. Sistemas de Informação: e-business global e colaboração. E-business Global e Colaboração. 2021. Disponível em: <https://dspace.uniceplac.edu.br/bitstream/123456789/1376/1/Sistemas%20de%20informa%C3%A7%C3%A3o%20-%20e-business%20global%20e%20colabora%C3%A7%C3%A3o.pdf>. Acesso em: 10 out. 2023.

LÓSSIO, Rodrigo. Mercado Livre adquire catarinense Axado por R\$ 26 milhões: Investimento amplia portfólio de soluções para logística da gigante do comércio eletrônico da América Latina. **Mercado Livre adquire catarinense Axado por R\$ 26 milhões**, Santa Catarina., p. n.p, 7 jun. 2016. Disponível em: <http://tisc.com.br/mercado-livre-adquire-axado/>. Acesso em: 22 mar. 2023.

MARTIN, Robert C. **Design Principles and Design Patterns**. 2000. Disponível em: https://staff.cs.utu.fi/staff/jouni.smed/doos_06/material/DesignPrinciplesAndPatterns.pdf. Acesso em: 29 fev. 2024.

MELO, Mariana Pereira. Tecnologia da Informação Aplicada à Logística. **Curso Técnico em Logística Educação a Distância**, Pernambuco: Rede e-Tec Brasi, ano 2018, ago. 2018. Disponível em: https://efivest.com.br/wp-content/uploads/2021/03/Tec._informacao_Aplicada_a_Logistica.pdf. Acesso em: 22 mar. 2023.

MORELLI, Daniel *et al.* Sistemas de Informação em Gestão da Cadeia de Suprimento. **Information systems in Supply Chain Management**, [s. l.], v. 17, ed. 33, p. 25-38, 24 mar. 2023. DOI <https://doi.org/10.15600/2238-1252/rct.v17n33p25->

38. Disponível em: <https://www.metodista.br/revistas/revistas-unimep/index.php/cienciatecnologia/article/view/954/675>. Acesso em: 24 mar. 2023.

PAIXÃO, João. O que é SOLID: O guia completo para você entender os 5 princípios da POO. *In*: PAIXÃO, João. **O que é SOLID: O guia completo para você entender os 5 princípios da POO**. [S. l.], 6 jan. 2019. Disponível em: <https://medium.com/desenvolvendo-com-paixao/o-que-%C3%A9-solid-o-guia-completo-para-voc%C3%AA-entender-os-5-princ%C3%ADpios-da-poo-2b937b3fc530>. Acesso em: 2 mar. 2024.

ROVEDA, Ugo. **TYPESCRIPT: O QUE É, COMO COMEÇAR E QUAIS SÃO AS VANTAGENS?** 2023. Disponível em: <https://kenzie.com.br/blog/typescript/>. Acesso em: 03 mar. 2024.

SANTOS, Robson dos. **JavaScript no Back-end: Tudo que Você Precisa Saber**. 2023. Disponível em: <https://www.brasilcode.com.br/javascript-no-back-end-tudo-que-voce-precisa-saber/>. Acesso em: 27 fev. 2024.

SANTOS, Robson dos. **JavaScript no Back-end: Tudo que Você Precisa Saber**. 2023. Disponível em: <https://www.brasilcode.com.br/javascript-no-back-end-tudo-que-voce-precisa-saber/>. Acesso em: 27 fev. 2024.

SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. São Paulo: Pearson, 2011. ISBN 978-85-7936-108-1.

TIPOS de Sistemas de Informação Empresariais - Parte I. Youtube: Prof. Ricardo Said, 2019. Disponível em: <https://www.youtube.com/watch?v=f07A5MGpqMU>. Acesso em: 3 abr. 2023.

VIDANYA, Bruna. **GraphQL vs. REST: Qual o Melhor Para o Desenvolvimento de API?** 2021. Disponível em: <https://www.hostinger.com.br/blog/graphql-vs-rest-qual-o-melhor-para-o-desenvolvimento-de-api/>. Acesso em: 04 mar. 2023.

VIDANYA, Bruna. **GraphQL vs. REST: Qual o Melhor Para o Desenvolvimento de API?** 2021. Disponível em: <https://www.hostinger.com.br/blog/graphql-vs-rest-qual-o-melhor-para-o-desenvolvimento-de-api/>. Acesso em: 04 mar. 2023.