



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

CAUÃ DA SILVA RIBEIRO

SISTEMA DE GERENCIAMENTO DE FROTAS DE VEÍCULOS

**Assis/SP
2024**



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

CAUÃ DA SILVA RIBEIRO

SISTEMA DE GERENCIAMENTO DE FROTAS DE VEÍCULOS

Projeto de pesquisa apresentado ao curso de Análise e Desenvolvimento de Sistemas do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando(a): Cauã da Silva Ribeiro

Orientador(a): Prof. Luiz Carlos Begosso

**Assis/SP
2024**

FICHA CATALOGRÁFICA

Ribeiro, Cauã da Silva

R484s Sistema e gerenciamento de frota / Cauã da Silva
Ribeiro. -- Assis, 2024.

34p. : il.

Trabalho de Conclusão de Curso (Graduação em Análise e
Desenvolvimento de Sistemas) -- Fundação Educacional do
Município de Assis (FEMA), Instituto Municipal de Ensino
Superior de Assis (IMESA), 2024.

Orientador: Prof. Dr. Luiz Carlos Begosso.

1. Eficiência organizacional. 2. Desenvolvimento de software. 3.
Veículos. I Begosso, Luiz Carlos. II Título.

CDD 003

RESUMO

Resumo — O presente trabalho objetivou desenvolver um sistema de gerenciamento de frotas para o controle de uma frota de veículos. Ele oferece alguns recursos para simplificar o gerenciamento diário, melhorar a eficiência operacional e reduzir o tempo. Um sistema de gerenciamento de frotas permite que as empresas acompanhem seus veículos, registrem dados monitorem e gerenciem sua frota.

Palavras-Chave — requisitos; Frotas; Gerenciamento; Desenvolvimento de software

ABSTRACT

Summary — This paper aims to develop a fleet management system for controlling a fleet of vehicles. It offers some features to simplify daily management, improve operational efficiency and reduce downtime. A fleet management system allows companies to track their vehicles, record data, monitor and manage their fleet.

Keywords — requirements; Fleets; Management; Software development

LISTA DE ILUSTRAÇÕES

Figura 1: Mapa mental.....	16
Figura 2 – Diagrama de caso de uso.....	18
Figura 3 –Narrativas dos UC– Manter Funcionário	19
Figura 4 –Narrativas dos UC – Manter Veículo	20
Figura 5 –Narrativas dos UC – Manter Motorista.....	22
Figura 6 –Narrativas dos UC – Manter Departamento	23
Figura 7 –Narrativas dos UC – Manter Manutenções.....	24
Figura 9 – Diagrama de classes.....	26
Figura 11 – Diagrama Entidade de Relacionamento.....	27
Figura 12 – Descrição da Classe Veículo.....	29
Figura 13 – Descrição do Repositório.....	30
Figura 14 – Descrição da Tela Front-End.....	31

LISTA DE TABELAS

Tabela 1 – Narrativa de caso de uso – Manter Funcionário.....	19
Tabela 2 – Narrativa de caso de uso – Manter Veículo.....	20
Tabela 3 – Narrativa de caso de uso – Manter Motorista.....	22
Tabela 4 – Narrativa de caso de uso – Manter Departamento.....	23
Tabela 5 – Narrativa de caso de uso – Manter Manutenções.....	24

SUMÁRIO

1. CAPÍTULO 1 – INTRODUÇÃO	9
1.1 OBJETIVO.....	10
1.2 PUBLICO ALVO.....	10
1.3 JUSTIFICATIVA.....	10
1.4 MOTIVAÇÃO.....	11
1.5 PERSPECTIVAS DE CONTRIBUIÇÃO.....	11
1.6 METODOLOGIA.....	12
2. CAPÍTULO 2 – TECNOLOGIAS UTILIZADAS	13
2.1 JAVA.....	13
2.2 SPRING BOOT.....	13
2.3 ANGULAR.....	13
2.4 MYSQL.....	14
2.5 BOOTSTRAP.....	14
2.6 INTELLIJ.....	15
3. CAPÍTULO 3 – MODELAGEM DO SOFTWARE	16
3.1 MAPA MENTAL.....	16
3.2 DIAGRAMA DE UC.....	17
3.3 NARRATIVAS DOS UC.....	19
3.3.1 UC 01 - Cadastrar Funcionário.....	19
3.3.2 UC 02 - Cadastrar Veiculo.....	20
3.3.3 UC 03 - Cadastrar Motorista.....	22
3.3.4 UC 04 - Manter Departamento.....	23
3.3.9 UC 09 - Manter Manutenções.....	24
3.4 DIAGRAMA DE CLASSES.....	25

3.5 DIAGRAMA ENTIDADE RELACIONAMENTO.....	26
4. DESCRIÇÃO DO SOFTWARE IMPLEMENTADO.....	27
4.1 DEFINIÇÃO DA CLASSE VEICULO E SUAS RELAÇÕES NO BANCO DE DADOS.....	27
4.2 ARQUITETURA MVC.....	28
4.3 ANGULAR.....	30
5. CONCLUSÕES.....	31
REFERENCIAS.....	32

1. CAPÍTULO 1 – INTRODUÇÃO

Atualmente, estamos imersos na era tecnológica, onde técnicas, conhecimentos e métodos são empregados para resolver problemas que antes eram tratados manualmente, muitas vezes apenas com o auxílio de planilhas feitas à mão. Agora, essas tarefas ganham eficientes gerenciadores, conferindo às empresas que os adotam uma vantagem competitiva significativa, independentemente do seu porte. A busca pela redução de custos e o aprimoramento do desempenho são objetivos centrais nesse cenário.

O software desenvolvido trata de um Sistema de Administração de Frotas (SGF), que poderá ser instalado em empresas que disponham de uma frota, ou prefeitura. É muito útil a qualquer empresa, gerenciando as inúmeras informações, manipulando e guardando em seu banco de dados.

Empresas que atuam neste campo, como a GEOTAB (2023) e a SofitView (2024), têm desenvolvido soluções avançadas de SGF. A GEOTAB, por exemplo, utiliza Internet das Coisas (IoT) e veículos conectados para fornecer soluções que visam aumentar a produtividade, otimizar o consumo de combustível e melhorar a manutenção da frota. Por sua vez, a SofitView oferece uma plataforma abrangente que centraliza dados cruciais para a operação de frotas, incluindo controle de pneus, registros de abastecimento e gestão de manutenção.

Considerando os pontos críticos nas tomadas de decisão relacionadas à gestão da Administração de Frotas, especialmente no que se refere aos sistemas de manutenção e composição da frota de uma empresa, um sistema dedicado pode oferecer uma ampla gama de informações através de seus relatórios. Esses relatórios são essenciais para auxiliar os gestores a alcançar a máxima eficiência operacional

O presente trabalho está organizado em cinco capítulos. A estrutura do artigo é a seguinte: Capítulo 1 Introdução: são apresentadas as áreas pesquisadas que estão sendo executadas no trabalho, contextualizando o campo de pesquisa, destacado a importância da área de Sistemas de Gerenciamento de Frotas e os objetivos do trabalho. No Capítulo 2 são apresentadas as tecnologias implementadas no sistema. O Capítulo 3 aborda a modelagem do software e os artefatos UML utilizados para representar o projeto do

software. O capítulo 4 que descreve o software implementado. Capítulo 5 que descreve conclusão do projeto.

1.1 – OBJETIVO

O objetivo deste projeto foi criar um sistema de controle e gerenciamento de uma frota de veículos com foco em ambiente web.

Para atingir este objetivo, foram desenvolvidas as seguintes funcionalidades no aplicativo:

- Permitir e controlar o cadastro de todos os veículos da frota;
- Apresentar manutenções dos veículos da frota;
- Permitir consultar cadastros gerais do sistema como os de veículos e departamentos.

Este estudo visa contribuir para o avanço do conhecimento na área de SGF, fornecendo *insights*, abordagens e recomendações práticas para aprimorar o gerenciamento de frotas, tanto do ponto de vista operacional quanto estratégico.

1.2 PÚBLICO-ALVO

Empresas e prefeituras que necessitam de um melhor controle de sua frota. Essas soluções aperfeiçoam recursos, reduzem custos operacionais e melhoram a produtividade, resultando em serviços de transporte mais confiáveis para a comunidade.

1.3 JUSTIFICATIVA

Ao notar a crescente demanda por tecnologia no setor automotivo, percebe-se que ainda há poucos softwares disponíveis para o público. Assim, este trabalho pode

contribuir, mesmo que de forma modesta, com inovação e produtividade nessa área ampla.

Observa que muitas ferramentas disponíveis não são fáceis de usar e, na maioria das vezes, suas funções são confusas. A forma como são visualmente organizadas é pouco intuitiva, o que acaba afastando pessoas com menos conhecimento em tecnologia. Por isso, o software apresentado busca atrair indivíduos com pouca familiaridade tecnológica, pois uma de suas propostas é oferecer uma interface mais simples e intuitiva possível, ajudando, de certa forma, na inclusão digital desse setor

1.4 MOTIVAÇÃO

A motivação para a realização desse trabalho está em colaborar com as empresas a administrar seus veículos de forma mais eficiente com objetivo de tornar a gestão de frotas mais fácil, econômica e segura. Através desse sistema, as empresas podem acompanhar seus veículos, organizar e ter um controle melhor da sua frota.

1.5 PERSPECTIVAS DE CONTRIBUIÇÃO

Com este projeto, espera-se oferecer um sistema rápido e fácil de usar, com uma interface simples e limpa para os usuários. Os retiros que usarem essa ferramenta poderão ter um melhor controle e acesso aos dados, o que pode ser útil em momentos em que essas informações forem importantes, além de trazer mais agilidade em algumas ações. Em alguns casos, também poderá ser útil para estudos no desenvolvimento de aplicativos nessa área. Além disso, o uso desse sistema pode melhorar a organização e eficiência no gerenciamento de atividades cotidianas.

1.6 METODOLOGIA

Para atender aos objetivos estabelecidos, foram conduzidos estudos teóricos sobre o sistema de gerenciamento de frotas, controle de viagens, ordens de serviço, alertas de manutenção, entre outros.

A respeito do desenvolvimento do software, foi utilizada a metodologia UML como forma de apresentar o projeto. Foram desenvolvidos os seguintes artefatos UML: diagrama de caso de uso, diagrama de classes e diagrama de sequência. A ferramenta online draw.io, foi utilizada para geração dos diagramas do projeto.

Na fase de implementação, utilizou-se a linguagem Java, com o *framework* Spring Boot, para projetos web, devido ao seu amplo uso no mercado de software e na criação das interfaces de usuário será adotado o framework Angular para usuários web.

CAPÍTULO 2 – TECNOLOGIAS UTILIZADAS

2.1 JAVA

De acordo com Rock (2019), é vastamente utilizada para desenvolver sites e aplicativos. A intenção predominante é possibilitar que os programadores escrevam o código uma única vez e possam executá-lo em qualquer tipo de plataforma, graças a JVM (Java Virtual Machine). A máquina virtual desenvolve pontos entre o aplicativo e o hardware do sistema. A linguagem JAVA consiste em um paradigma orientado a objetos popularmente conhecido como POO (Programação Orientada a Objetos), que permite elaborar diversos tipos de dados utilizados, para se conectarem às suas respectivas funções

2.2 SPRING BOOT

O Spring Boot é um framework de código aberto com uma estrutura de *back-end*, sendo muito utilizado para descomplicar as configurações de uma aplicação, ou seja, agiliza o trabalho de um desenvolvedor. Nesse sentido, o Spring Boot traz facilidade ao direcionar para o usuário, melhores possibilidades de desenvolver uma ação. Sendo assim, o desenvolvedor otimiza seu tempo, podendo se dedicar mais a codificação ao contrário de perder muito tempo configurando aplicações (SILVA, 2023).

2.3 ANGULAR

Segundo o site Treinaweb (2004), atualmente o Angular é um dos frameworks Javascript que dominam o mercado, sendo muito popular nos últimos anos e é utilizado em inúmeros projetos. Por esses motivos, ele vem sendo amplamente requisitado no mercado para desenvolvimento web e mobile, tendo diversas oportunidades no Brasil e no mundo.

Produtividade aliada à performance: consegue oferecer suporte ao desenvolvimento rápido de aplicações através de uma API simples, bem estruturada e bem documentada, o que acaba trazendo bastante produtividade

2.4 MYSQL

O MySQL foi escolhido como o banco de dados para o projeto. Em conformidade com Cerqueira (2021), é uma plataforma de gerenciamento de bancos de dados relacional de código aberto, amplamente utilizado (RDBMS). Pode ser executado em sistemas operacionais famosos, incluindo Windows, Linux e macOS, além de oferecer a capacidade de armazenar, gerenciar e organizar dados em diversos aplicativos, desde 17 pequenos sites até grandes aplicações. Ademais, é compatível com muitas linguagens de programação e para consultas é baseado em linguagem de pesquisa estruturada (SQL). A Oracle Corporation é a responsável pelas manutenções e desenvolvimento do MySQL.

2.5 BOOTSTRAP

Para Noletto (2022), trata-se de um framework normalmente utilizado junto com JavaScript e CSS no Front-End, para aprimorar as páginas somando com diversas aplicabilidades, provendo bem mais que apenas uma bela aparência ao site, pois sucede implementar responsividade, controles de paginação, formulários e muitas outras funções disponíveis, com o objetivo de favorecer a reutilização de código reduzindo o tempo de desenvolvimento de várias linhas do código.

2.5 INTELLIJ

Segundo o site Awari(2024), o IntelliJ oferece tudo o que os desenvolvedores precisam para criar aplicativos de software de forma rápida e eficiente. Ele possui um editor de código poderoso que permite escrever, editar e depurar o código de forma rápida e intuitiva. Além disso, ele oferece um depurador integrado que ajuda os desenvolvedores a encontrar e corrigir problemas no código. Outro recurso importante do IntelliJ é a integração com várias ferramentas de controle de versão, como Git, SVN e Mercurial. Isso torna mais fácil para os desenvolvedores gerenciar o código fonte do projeto

CAPÍTULO 3 – MODELAGEM DO SOFTWARE

Para garantir que o desenvolvimento de software atenda aos padrões de excelência e eficiência, a documentação precisa ser clara e bem organizada, facilitando o entendimento por quem não participou do processo de desenvolvimento. Uma documentação detalhada é essencial para assegurar a qualidade e também para tornar sua manutenção e uso mais simples. Além disso, é importante incluir o Mapa Mental como parte desse processo.

3.1. MAPA MENTAL:

Figura 1 ilustra o mapa mental que abrange os principais cadastros de dados e as consultas relacionadas à cada cadastro já realizada movimentação dentro do sistema.

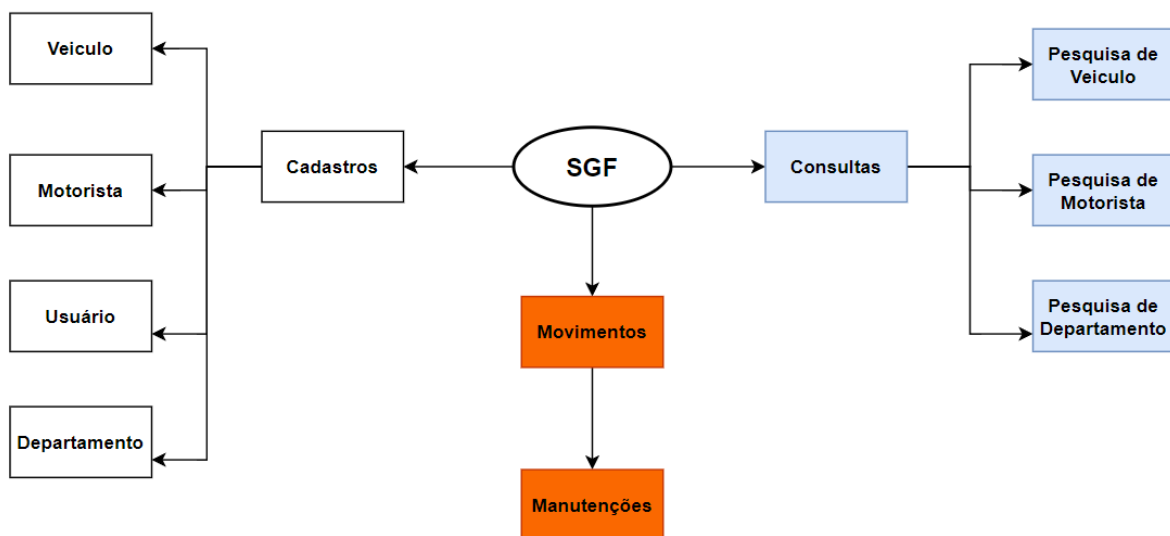


Figura 1 – Mapa mental

3.2. DIAGRAMA DE UC:

O diagrama de caso de uso tem como objetivo resumir as interações e os principais aspectos do sistema com os atores, documentando cenários, casos de uso e funcionalidades. Sua finalidade é apresentar uma visão geral sem entrar em muitos detalhes. A seguir, está o diagrama de caso de uso do sistema:

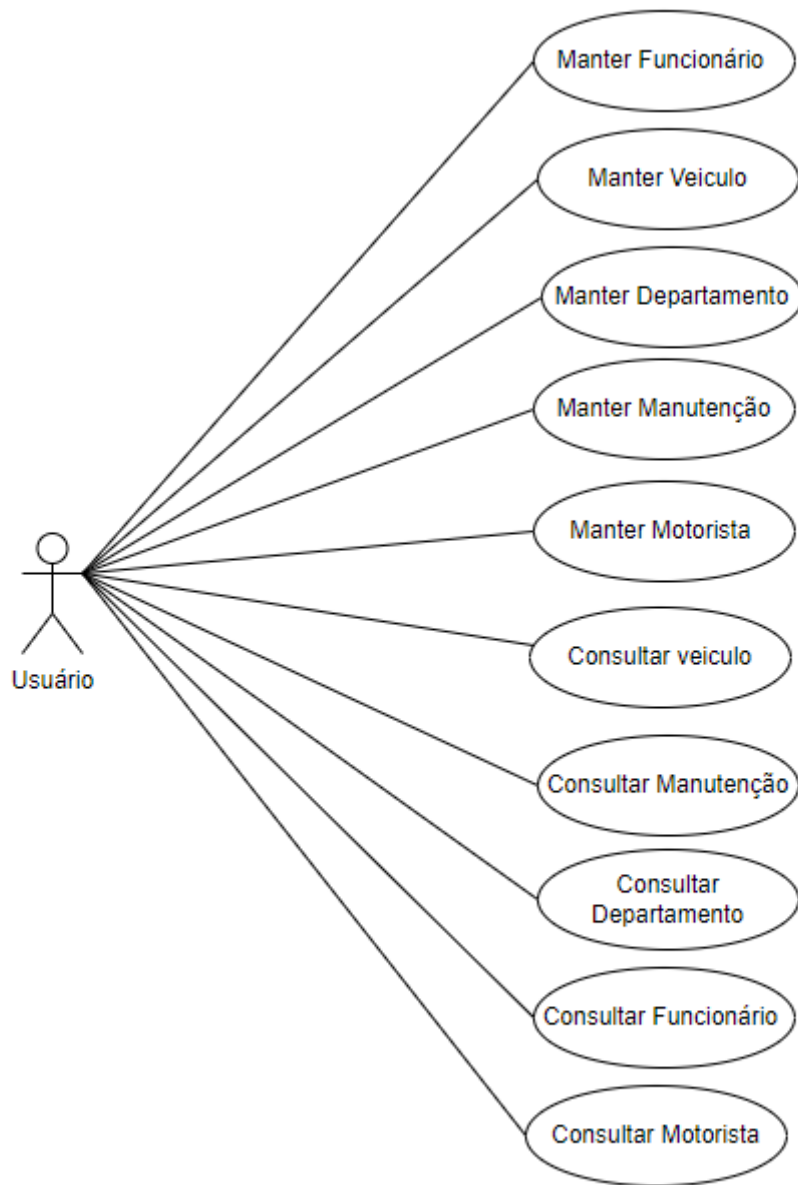


Figura 2 – Diagrama de caso de uso

3.3 NARRATIVAS DOS UC:

3.3.1 UC 01 – Cadastrar Funcionário:

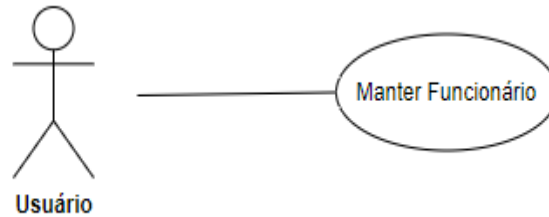


Figura 3 – Diagrama de caso de uso – Manter Funcionário

Finalidade/Objetivo	Cadastrar, pesquisar e editar um funcionário.
Atores	Gestor
Evento Inicial	O ator deve estar logado ao sistema.
Fluxo Principal	<p>a) O caso de uso é iniciado quando o Ator acessa o sistema e seleciona e entra com “Usuário” para menu principal.</p> <p>b) O sistema apresenta a interface “Cadastro de Usuários”. [A1], [A2].</p> <p>c) O sistema solicita confirmação de inclusão.</p> <p>d) O sistema apresenta a mensagem “Operação realizada com sucesso”.</p> <p>e) O caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Editar Funcionário</p> <p>a) O ator altera os dados cadastrais desejados e seleciona a opção “Salvar”.</p> <p>c) O ator confirma a alteração selecionando a opção “SIM”.</p> <p>d) O sistema retorna para o passo “e” do fluxo principal.</p> <p>A2 – Excluir Funcionário</p> <p>a) O ator seleciona a opção “Excluir”. [E4].</p>

	<p>b) O sistema solicita a confirmação da exclusão.</p> <p>c) O ator confirma a exclusão selecionando a opção “SIM”.</p> <p>d) O sistema retorna para o passo “e” do fluxo principal.</p>
Fluxo de Exceção	<p>E1 – Já Existente</p> <p>a) O usuário na qual deseja cadastrar já existe;</p> <p>b) Sistema encaminha para o passo “b” do fluxo principal.</p> <p>E2 – Nome</p> <p>O Sistema apresenta a mensagem “Nome inválido”.</p> <p>E3 – Campos Obrigatórios</p> <p>O Sistema apresenta a mensagem “Campo obrigatório não preenchido, favor verificar”.</p>

Tabela 1 – Narrativa de caso de uso – Manter Funcionário

3.3.2 UC 02 – Cadastrar Veículo:

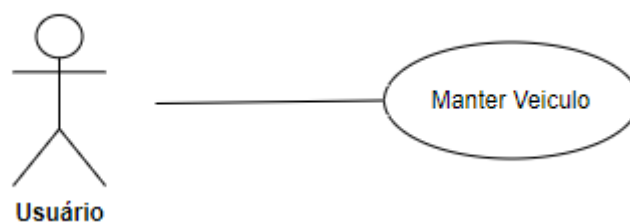


Figura 4 – Diagrama de caso de uso – Manter Veículo

Finalidade/Objetivo	Cadastrar, pesquisar e editar um carro.
Atores	Gestor
Evento Inicial	O ator deve estar logado ao sistema.
Fluxo Principal	<p>a) O caso de uso é iniciado quando o Ator acessa o sistema e seleciona a opção “Veículo” no menu principal.</p> <p>b) O sistema apresenta a interface “Cadastro</p>

	<p>de Veículos”. [A1], [A2].</p> <p>c) O ator preenche todos os campos solicitados “salvar”. [E2], [T1], [T2].</p> <p>d) O sistema solicita confirmação de inclusão.</p> <p>e) O ator confirma a inclusão selecionando a opção “SIM”.</p> <p>f) O sistema apresenta a mensagem “Operação realizada com sucesso”.</p> <p>g) O caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Editar Veículo</p> <p>a) O ator altera os campos desejados e seleciona a opção “Salvar”.</p> <p>b) O sistema solicita a confirmação da alteração. [E3].</p> <p>c) O ator confirma a alteração selecionando a opção “SIM”.</p> <p>d) O sistema retorna para o passo “g” do fluxo principal.</p> <p>A2 – Excluir Veículo</p> <p>a) O ator seleciona a opção “Excluir”. [E4].</p> <p>d) O sistema retorna para o passo “g” do fluxo principal.</p>
Fluxo de Exceção	<p>E1 – Já Existente</p> <p>a) O veículo na qual deseja cadastrar já existe;</p> <p>b) O sistema informa se deseja editar o veículo existente;</p> <p>c) Sistema encaminha para o passo “b” do fluxo principal [A1].</p> <p>E2 – Nome Inválido</p> <p>O Sistema apresenta a mensagem “Nome inválido”.</p> <p>E3 – Campos Obrigatórios</p> <p>O Sistema apresenta a mensagem “Campo obrigatório não preenchido, favor verificar”.</p>

Testes	<p>T1 – Testar Placa do Veículo</p> <p>a) Sistema verifica se a Placa é válida;</p> <p>b) Retorna para o passo “e” do fluxo principal;</p>
--------	---

Tabela 2 – Narrativa de caso de uso – Manter Veículo

3.3.3 UC 03 – Cadastrar Motorista:

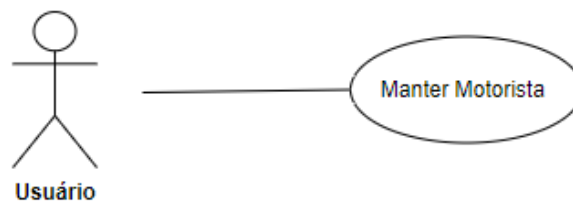


Figura 5 – Diagrama de caso de uso – Manter Motorista

Finalidade/Objetivo	Cadastrar, pesquisar e editar um Motorista.
Atores	Gestor
Evento Inicial	O ator deve estar logado ao sistema.
Fluxo Principal	<p>a) O caso de uso é iniciado quando o Ator acessa o sistema e seleciona a opção “Veículo” no menu principal.</p> <p>b) O sistema apresenta a interface “Cadastro de Motoristas”. [A1], [A2].</p> <p>c) O ator preenche todos os campos solicitados “salvar”. [E2], [T1], [T2].</p> <p>b) O ator confirma a inclusão selecionando a opção “SIM”.</p> <p>e) O sistema apresenta a mensagem “Operação realizada com sucesso”.</p> <p>g) O caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Editar Veículo</p> <p>a) O ator altera os campos desejados e seleciona a opção “Salvar”.</p> <p>b) O sistema retorna para o passo “g” do fluxo</p>

	<p>principal.</p> <p>A2 – Excluir Veículo</p> <p>a) O ator seleciona a opção “Excluir”. [E4].</p> <p>b) O sistema retorna para o passo “g” do fluxo principal.</p>
Fluxo de Exceção	<p>E1 – Já Existente</p> <p>a) O usuário na qual deseja cadastrar já existe;</p> <p>b) Sistema encaminha para o passo “b” do fluxo principal [A1].</p> <p>E2 – Campos Obrigatórios</p> <p>O Sistema apresenta a mensagem “Campo obrigatório não preenchido, favor verificar”.</p>

Tabela 3 – Narrativa de caso de uso – Manter Motorista

3.3.4 UC 04 – Manter Departamento:



Figura 6 – Diagrama de caso de uso – Manter Departamento e Subdepartamento

Finalidade/Objetivo	Responsável por cadastrar Departamento.
Atores	Usuário

Evento Inicial	O ator deve estar logado ao sistema.
Fluxo Principal	<p>a) O usuário escolhe no menu a opção "cadastro de departamentos".</p> <p>c) O Software direciona o usuário para uma tela de cadastro de departamentos e usuário preenche os campos solicitados e clica em "salvar". [E1].</p> <p>d) Se todos os dados estiverem corretos o cadastro é salvo no banco de dados</p> <p>e) O caso de uso é encerrado.</p>
Fluxo de Alternativo	<p>E1 – Não a veículos</p> <p>a) Se a departamento já estiver cadastrada o software emite a mensagem "Registro já Cadastrado".</p>

Tabela 4 – Narrativa de caso de uso – Manter Departamento e Subdepartamento

3.3.9 UC 09 – Manter Manutenções:

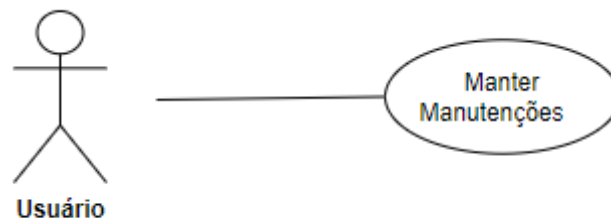


Figura 7 – Diagrama de caso de uso – Manter Manutenções

Finalidade/Objetivo	Responsável pelos registros de manutenções
Atores	Usuário

Evento Inicial	O ator deve estar logado ao sistema.
Fluxo Principal	<p>a) O usuário escolhe no menu a opção “Informar a manutenção”.</p> <p>b) O usuário preenche todos os campos solicitados.</p> <p>c) O Software direciona o usuário para uma tela pesquisa se realmente necessitar de cadastrar clique em “Salvar”.</p> <p>d) Se todos os dados estiverem corretos o cadastro é salvo no banco de dados</p> <p>e) O caso de uso é encerrado.</p>

Tabela 5 – Narrativa de caso de uso – Manter Manutenções

3.4. DIAGRAMA DE CLASSES:

De acordo com LUCIDCHART (2023), diagrama de classe é a representação estática de um sistema, eles descrevem cada classe com seus atributos, métodos e objetos, além das conexões entre elas, esse diagrama é utilizada para definir como será a implementação de um sistema orientado a objetos. A seguir está o diagrama de classe do sistema:

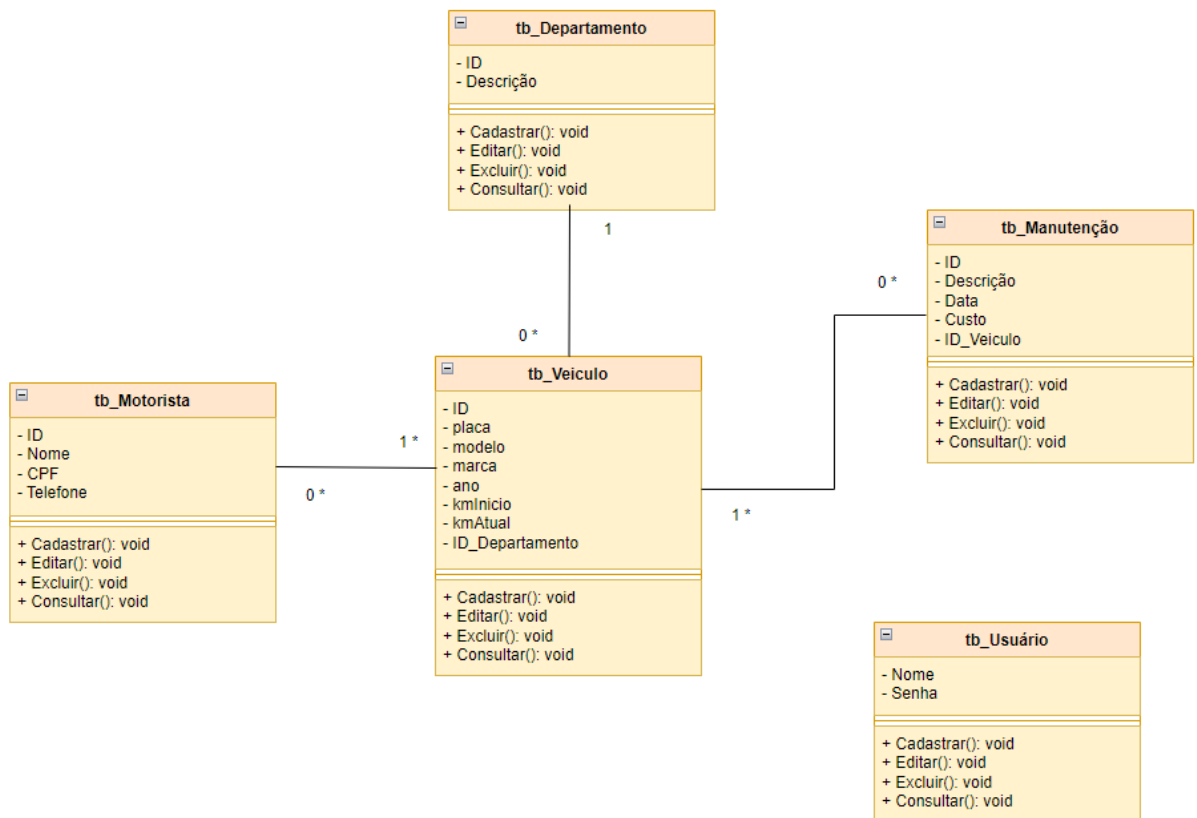


Figura 6– Diagrama de Classes

3.5. DIAGRAMA ENTIDADE RELACIONAMENTO:

Conforme LUCIDCHART (2023), o diagrama entidade relacionamento é um fluxograma para expressar todas as entidades de um sistema, sejam elas pessoas, objetos, etc., em geral esses diagramas são empenhados na projeção de bases de dados,

ele é representado por uma série de símbolos, entidades e os relacionamentos entre as entidades. A seguir é apresentado o DER do sistema de frota:

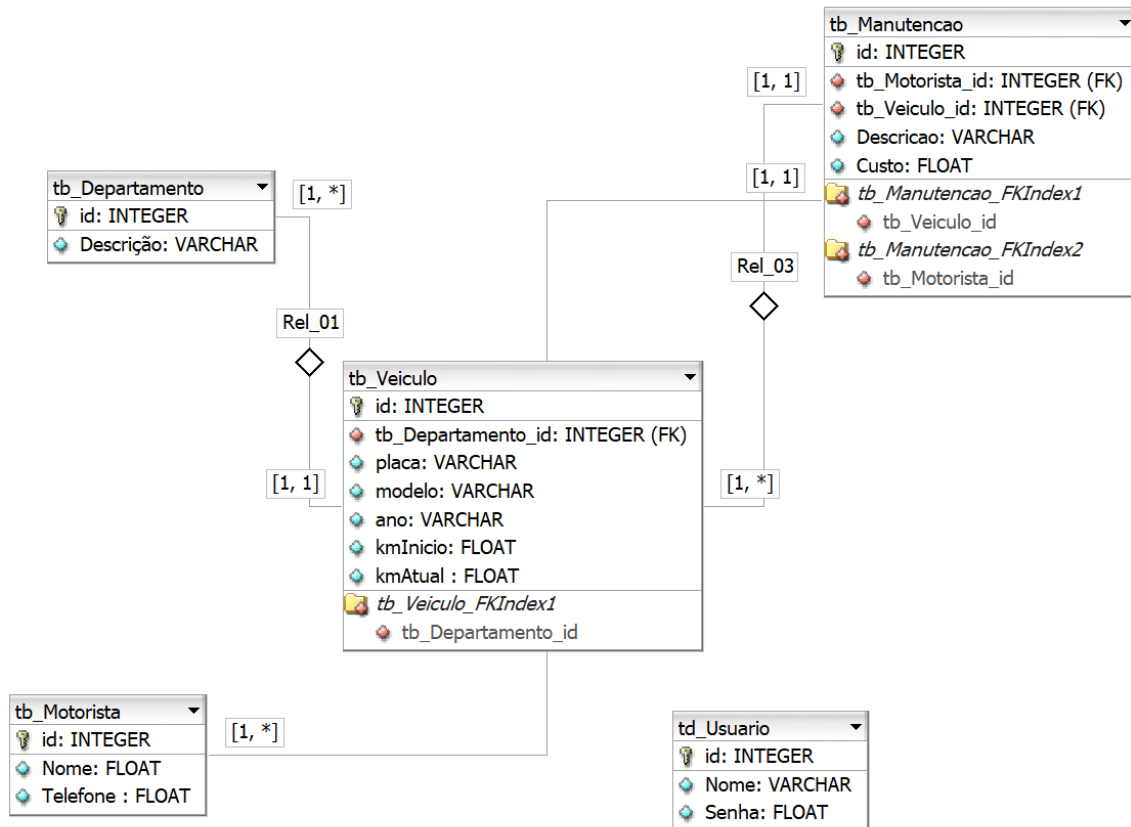


Figura 9– Diagrama Entidade de Relacionamento

CAPITULO 4 – DESCRIÇÃO DO SOFTWARE IMPLEMENTADO

Nesse capítulo, será abordada a fase final do trabalho, foram descritas as tecnologias e metodologias utilizadas na construção do software, destacando os processos principais que foram utilizados e as estratégias de desenvolvimento. Serão

exibidos trechos de código da implementação demonstrando diferentes partes do sistema e será discutida a maneira como se desenvolveu a solução. O objetivo das próximas seções deste capítulo é fornecer uma visão da execução desse projeto.

4.1. DEFINIÇÃO DA CLASSE VEICULO E SUAS RELAÇÕES NO BANCO DE DADOS

A figura 10 ilustra o código que cria uma classe chamada "Veiculo", que está ligada à tabela "veiculo" no banco de dados. Essa classe é uma "entidade", o que significa que ela representa uma tabela no banco de dados. O campo "id_Veiculo" é a chave principal da tabela e seu valor é gerado automaticamente. As colunas da tabela são definidas na classe com informações como o nome da coluna, o tamanho máximo do valor e se pode ficar vazia. Além disso, a classe "Veiculo" tem uma ligação com a classe "Departamento", onde um veículo pertence a um departamento, e essa relação é representada pela chave estrangeira "departamento_id".

```

@Entity
@Table(name = "veiculo")
public class Veiculo {

    3 usages
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id_Veiculo")
    private Long id_Veiculo;

    3 usages
    @Column(name = "placa", length = 30, nullable = false)
    private String placa;

    3 usages
    @Column(name = "modelo", length = 30, nullable = false)
    private String modelo;

    3 usages
    @Column(name = "marca", length = 30, nullable = false)
    private String marca;

    3 usages
    @Column(name = "ano", nullable = false)
    private String ano;

    3 usages
    @Column(name = "kmInicio", nullable = false)
    private Long kmInicio;

    3 usages
    @Column(name = "kmAtual", nullable = false)
    private Long kmAtual;

    2 usages
    @ManyToOne
    @JoinColumn(name = "departamento_id", nullable = false)
    private Departamento departamento;

```

Figura 10 – Descrição da Classe Veículo

4.2. FUNÇÕES DE VEICULO REPOSITORY PARA GERENCIAR VEÍCULOS NO BANCO DE DADOS

O código cria uma interface chamada "VeiculoRepository", que é responsável por acessar os dados de veículos no banco de dados. Ela já vem com métodos prontos para adicionar, buscar, atualizar e remover veículos. Além disso, tem métodos personalizados para verificar se um veículo existe pela placa, encontrar um veículo pela placa e listar

veículos de um determinado departamento. Conforme ilustrado pela figura 11.

```
package github.com.cauarb.tccSGF.domain.veiculo.repository;

import github.com.cauarb.tccSGF.domain.veiculo.entity.Veiculo;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import java.util.List;
import java.util.Optional;

@Repository
public interface VeiculoRepository extends JpaRepository <Veiculo, Long> {
    1 usage
    boolean existsByPlaca(String placa);
    1 usage
    Optional<Veiculo> findByPlaca(String placa);
    1 usage
    List<Veiculo> findByDepartamentoNome(String nome);
}
```

Figura 11 – Descrição do Repositório

4.2. ARQUITETURA MVC

Durante o desenvolvimento e idealização do software, foram aplicados conceitos de arquitetura MVC (*Model View Controller*) no projeto, de maneira que ele ficou em boa parte dividido em classes de modelo contendo informações sobre os objetos que consiste na camada de dados, classes de Controle que contém as regras para processar os dados corretamente e classes de *View* que representam as telas e a parte visual do sistema. A

partir da adoção de algumas práticas desse modelo, foi possível tornar o software mais fácil de se construir e trabalhar, além de trazer uma organização melhor para o código.

4.3. ANGULAR

Agora, focando na parte visual do sistema, foram criadas várias telas usando o Angular. Para isso, foram feitos componentes e módulos de forma organizada e eficiente. Cada componente foi construído com HTML para o layout, CSS para o visual e JavaScript junto com TypeScript para a lógica. Essa separação ajudou a deixar o design adaptável e o código mais fácil de manter. Além disso, boas práticas de design foram seguidas para garantir uma experiência simples e agradável para o usuário

A figura 12 ilustra um exemplo de tela de CRUD da Tela de Cadastro de Veículo:

The screenshot shows a web application interface for vehicle registration. At the top, there is a dark navigation bar with the text 'Frota de Veículos' and a hamburger menu icon. Below this is a light gray header with the title 'Cadastros de Veiculos'. A subtitle below the header reads 'Nesta opção você pode realizar seu cadastros de veiculos'. The main content area contains a form with the following fields:

ID:	Placa: *
Modelo: *	Marca: *
Ano: *	KM Início: *
KM Atual: *	Departamento: *

Below the form is a green button with a white document icon and the text 'Salvar'. At the bottom of the page, there is a footer with 'Copyright © SFG 2024' on the left and 'Privacy Policy - Terms & Conditions' on the right.

Figura 12 – Descrição da tela front-end

CAPITULO 5 – CONCLUSÕES

Este projeto representou um grande desafio pessoal para o autor. Ao longo dessa experiência foi possível explorar uma área de desenvolvimento com a qual o presente autor tinha tido pouco contato. Com base em extensa pesquisa e dedicação, foi possível compreender o *framework* Angular e as ferramentas associadas.

Durante o desenvolvimento, foram aplicadas habilidades obtidas no curso de forma a criar uma aplicação utilizando conceitos modernos e amplamente utilizados no mercado. Embora o objetivo principal de criar uma aplicação completa para o sistema de frotas de veículos não tenha sido totalmente alcançado, o projeto ofereceu valiosas perspectivas sobre a melhor abordagem para o desenvolvimento de software nesta área.

Acredita-se na possibilidade de que futuros trabalhos que possam aprimorar este software, adicionando novas funcionalidades e melhorando aspectos que ainda podem ser otimizados.

Em conclusão, esta experiência foi extremamente satisfatória. A jornada de criação deste projeto de monografia e software proporcionou novos conhecimentos e habilidades que serão de grande valor tanto pessoal quanto profissional.

REFERÊNCIAS

AWS, Amazon Web Services, Inc. O que é Java. Local. Disponível em:<
<https://aws.amazon.com/pt/what-is/java/>>. Acesso em: 30 mar e 2024.

BERTAGLIA, Paulo Bertaglia. Logística e Gerenciamento da Cadeia de Abastecimento . Local. Disponível em:<
https://www.academia.edu/34295016/PAULO_ROBERTO_BERTAGLIA_AUTOR_DOS_LIVROS_LOGISTICA_E_GERENCIAMENTO_DA_CADEIA_DE_ABASTECIMENTO_EDITORA_SARAIVA_SERVI%C3%87O_A_CLIENTE ESTRAT%C3%89GIAS_GLOBAIS_DE_LOG%C3%8DSTICA_FUNDADOR_E_CEO_DA_BERTHAS>. Acesso em: 30 mar e 2024.

CENTER INFORMATICA, Marcio Marques. Softwares de Gestão. Local. Disponível em:<
<https://centerinformatica.com.br/solucoes/sqf-sistema-de-gerenciamentode-frotas> >. Acesso em: 30 mar e 2024.

COBIL, Paulo Silva. Sistema de Gerenciamento de frotas. Local. Disponível em:<
https://www.cobli.co/?internal_campaign=blog_direct&internal_content=navbar_site>. Acesso em: 30 mar e 2024.

ENCICLOPÉDIA SIGNIFICADOS, Adriano Padilha. Mapa Mental. Local. Disponível em:
<<https://www.significados.com.br/mapamental/#:~:text=Um%20mapa%20mental%20%C3%A9%20um,outras%20informa%C3%A7%C3%B5es%20a%20ela%20relacionadas> >. Acesso em: 30 mar e 2024.

GEOTAB, Geotab inc. Título do Trabalho. Local. Disponível em:<
<https://www.geotab.com/pt-br/blog/gestao-de-frotas-guia-completo/>>. Acesso em: 30 mar e 2024.

GOOGLE CLOUD, Edgar F. Codd. O que é o MySQL. Local. Disponível em. Acesso em: 30 mar e 2024.

HOSTINGER, Bruna Vidanya. O Que É Angular: O Framework Que Transformará Seu Código. Local. Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-e-angular>>. Acesso em: 30 mar e 2024.

IBM, IBM Corporation. O que é Java Spring Boot. Local. Disponível em: <<https://www.ibm.com/br-pt/topics/java-spring-boot>>. Acesso em: 30 mar e 2024.

IBM, IBM Corporation. Diagramas de Classes. Local. Disponível em: <<https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams>>. Acesso em: 30 mar e 2024.

LUCIDCHART, Peter Chen, Charles Bachman. O que é um diagrama entidade relacionamento. Local. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-entidade-relacionamento>>. Acesso em: 30 mar e 2024.

LUCIDCHART, Dave Grow. O que é um diagrama UML. Local. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-uml#:~:text=A%20UML%20%C3%A9%20uma%20combina%C3%A7%C3%A3o,e%20mais%20f%C3%A1cil%20de%20usar>>. Acesso em: 30 mar e 2024.

SOFIT. Eduardo Canas. Sofit, o Canal do Gestor de Frotas. Local. Disponível em: <<https://www.sofit4.com.br>>. Acesso em: 30 mar e 2024.

VALENTE. Amir Mattar Valente, Antônio Galvão Novaes, Eunice Passaglia e Heitor Vieira. Gerenciamento de Transporte e Frotas. Local. Disponível em: <https://issuu.com/cengagebrasil/docs/gerenciamento_de_transporte_e_frota>. Acesso em: 30 mar e 2024.

CARVALHO, Andreyana. Entenda o que é Diagrama de Entidade e Relacionamento (DER). Coodesh, 2022. Disponível em: <<https://www.devmedia.com.br/mer-e-der-modelagemde-bancos-de-dados/14332>>. Acesso em: 08 de set. de 2024.

CERQUEIRA, Aléxis. MySQL: o que é e como usar o sistema. Tecmundo, 2021. Disponível em: <<https://www.tecmundo.com.br/software/223924-mysql-usar-osistema.htm>>. Acesso em: 08 de set. de 2024.

ROCK, Redator. O que é Java? Conheça as particularidades dessa linguagem de programação. Rockcontent, 2019. Disponível em: <<https://rockcontent.com/br/blog/o-quee-java>>. Acesso em: 05 de mar. de 2023.

SILVA, Gizela. O que é Spring Boot? Coodesh,2023. Disponível em: <<https://coodesh.com/blog/dicionario/o-que-e-spring-boot/>>. Acesso em: 08 de set. de 2024.

TREINAWEB, Marylene Guedes. Angular: O que é o Angular e para que serve? Disponível em: < <https://www.treinaweb.com.br/blog/o-que-e-o-angular-e-para-que-serve>>. Acesso em: 08 de set. de 2024.

NOLETO, Cairo. Bootstrap: o que é, como usar e para que serve esse framework. Betrybe, 2022. Disponível em: <<https://blog.betrybe.com/bootstrap/>>. Acesso em: 08 de set. de 2024.

FABFORCE. DBDesigner 4 fabFORCE. 2003. Disponível em < <https://www.fabforce.net/dbdesigner4/index.php>>. Acesso em: 08 fev.2024.

LUCIDCHART. O que é UML? Lucidchart. 2023. Disponível em < [https://www.lucidchart.com/pages/pt/o-que-e-uml#:~:text=um%20diagrama%20UML-O%20que%20%C3%A9%20UML%3F,tanto%20estruturalmente%20quanto%20para%20,comportamentos./](https://www.lucidchart.com/pages/pt/o-que-e-uml#:~:text=um%20diagrama%20UML-O%20que%20%C3%A9%20UML%3F,tanto%20estruturalmente%20quanto%20para%20,comportamentos./>)>. Acesso em: 08 set.2024

LUCIDCHART. Conceito de diagrama entidade relacionamento Lucidchart. 2023. Disponível em <<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-entidaderelacionamento>>. Acesso em: 08 set.2024

AWARI. Conhecendo o IntelliJ. Disponível em < <https://awari.com.br/intelli-j/>>. Acesso em: 08 set.2024

