INSTITUTO FEDERAL DE SANTA CATARINA

ALANA BEATRIZ ROSTIROLLA MANDIM

Sistema Computacional para Reserva e Controle da Entrada- e-Saída de Veículos

São José - SC

Dezembro/2023

SISTEMA COMPUTACIONAL PARA RESERVA E CONTROLE DA ENTRADA- E-SAÍDA DE VEÍCULOS

Monografia apresentada ao Curso de Engenharia de Telecomunicações do campus São José do Instituto Federal de Santa Catarina para a obtenção do diploma de Engenheira de Telecomunicações.

Orientador: Prof. Arliones Stevert Hoeller Jr., Dr.

São José - SC

Dezembro/2023

Alana Beatriz Rostirolla Mandim

Sistema Computacional para Reserva e Controle da Entrada-e-Saída de Veículos

Este trabalho foi julgado adequado para obtenção do título de Engenheira de Telecomunicações, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

São José - SC, 21 de dezembro de 2023:

Prof. Arliones Stevert Hoeller Jr., Dr.
Orientador
Instituto Federal de Santa Catarina

Prof. Emerson Ribeiro de Mello, Dr.
Instituto Federal de Santa Catarina

Prof. Cleber Jorge Amaral, Dr.
Instituto Federal de Santa Catarina

Este trabalho é dedicado à minha família que sempre me apoiou.

AGRADECIMENTOS

Os agradecimentos principais são direcionados à minha mãe, Ana Beatriz, que abriu mão da sua zona de conforto para vir morar aqui em Florianópolis para que eu pudesse continuar minha faculdade, ao meu esposo, Jeferson Cadorin, que sempre esteve ao meu lado em todos os momentos, muitas vezes assumindo funções que eram minhas para que eu pudesse me dedicar aos estudos, ao meu filho, Henrique Cadorin, por me motivar todos os dias a ser uma mãe melhor, ao meu orientador, Arliones, que foi muito paciente e a todo momento buscou me auxiliar por fim não menos importante a todos os professores que direta ou indiretamente contribuíram para que pudesse chegar até aqui.

Nossas atitudes escrevem nosso destino. Nós somos responsáveis pela vida que temos. Culpar os outros pelo que nos acontece é cultivar a ilusão. A aprendizagem é nossa e ninguém poderá fazê-la por nós, assim como nós não poderemos fazer pelos outros. Quanto mais depressa aprendermos isso, menos sofreremos.

Zíbia Gasparetto

RESUMO

A tecnologia desempenha um papel crucial na simplificação de tarefas cotidianas, oferecendo suporte em diversas atividades. Este estudo propõe o desenvolvimento de um sistema de gestão de frotas empregando a tecnologia da Internet das Coisas (IoT). O objetivo principal é substituir métodos manuais de controle de entrada e saída de veículos por uma solução computacional, tornando informações mais ágeis e acessíveis a todos os envolvidos. A implementação deste sistema, integrando conceitos de gestão de frotas e Internet das Coisas (IoT), busca evitar erros decorrentes da falta de acesso a dados cruciais, bem como prevenir o cancelamento de atividades programadas. O trabalho explora formas de identificação de pessoas e coisas visando a gestão de frotas, incorporando técnicas de *machine learning*. Utiliza o *framework Spring Boot* para tornar mais prático o desenvolvimento do *backend*, a biblioteca *React* com a linguagem *Typescript* no desenvolvimento do *frontend*, e microcontroladores para auxiliar no controle de dispositivos IoT.

Palavras-chave: Gestão de Frotas. Internet das Coisas. *Spring Boot*. *React*.

ABSTRACT

Technology plays a crucial role in simplifying everyday tasks, offering support across various activities. This study proposes the development of a fleet management system utilizing the technology of the Internet of Things (IoT). The primary objective is to replace manual methods of vehicle entry and exit control with a computational solution, making information more agile and accessible to all stakeholders. The implementation of this system, integrating fleet management concepts with IoT, aims to prevent errors due to the lack of access to critical data and to avoid the cancellation of scheduled activities. The work explores methods for identifying people and objects for fleet management, incorporating machine learning techniques. The system leverages the Spring Boot framework to streamline backend development, the React library with Typescript for frontend development, and microcontrollers to assist in the control of IoT devices.

Keywords: Fleet Management. Internet of Things. Spring Boot. React.

LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura de três camadas. 19
Figura 2 – Arquitetura com cinco camadas. 20
Figura 3 – Etapas de modelagem do projeto de banco de dados. 25
Figura 4 – Fluxo Cliente-Servidor através da API. 27
Figura 5 – Diagrama de Casos de Uso. 32
Figura 6 – Diagrama de Implantação. 34
Figura 7 – Diagrama de Entidade Relacionamento. 35
Figura 8 – *Site Map* da interface. 37
Figura 9 – *Site Map* Motorista. 38
Figura 10 – *Site Map* Gestor. 39
Figura 11 – *Site Map* Aprovador. 40
Figura 12 – *Site Map* Chefe. 41
Figura 13 – Tela de Login. 47
Figura 14 – Tela de Registro. 47
Figura 15 – Telas para fazer uma solicitação de veículo. 47
Figura 16 – Telas para gerenciar uma solicitação de veículo. 48
Figura 17 – Diagrama de Sequencia - Visualização e aprovação de solicitações. . . . 49
Figura 18 – Telas para gerenciar uma solicitação de veículo. 50
Figura 19 – Diagrama de Sequência para acompanhamento, inspeção e identificação na saída do veículo. 51
Figura 20 – Diagrama de Sequência para acompanhamento, inspeção e identificação no retorno do veículo. 52

LISTA DE TABELAS

Tabela 1 – Tabela Comparativa: QR Code, RFID e Reconhecimento de Placa para Identificação de Veículo. 22
Tabela 2 – Comparativo de Soluções Tecnológicas em Gestão de Frotas. 28
Tabela 3 – Tabela de utilização de requisição GET para o *backend*. 60
Tabela 4 – Tabela de utilização de requisições POST e PUT para o *backend*. . . . 61

LISTA DE ABREVIATURAS E SIGLAS

CRUD *Create, Read, Update, Delete.*

DTCEA-FL Destacamento de Controle do Espaço Aéreo de Florianópolis.

HF *High Frequency.*

HTTP *Hyper Transfer Protocol.*

IoT Internet das Coisas.

Iot-A *Internet of Things Architecture.*

IP *Internet Protocol.*

JPA *Java Persistence API.*

LF *Low Frequency.*

MQTT *Message Queuing Telemetry Transport.*

MVP *Minimum Viable Product.*

QR Code *Quick Response Code.*

RFID Radio Frequency Identification.

SGBDs Sistemas de gerenciamento de banco de dados.

UHF *Ultra High Frequency.*

SUMÁRIO

1 INTRODUÇÃO 13
1.1 Objetivos 14
1.1.1 Objetivo Geral 14
1.1.2 Objetivos Específicos 14
1.2 Estrutura do trabalho 15

2 FUNDAMENTAÇÃO TEÓRICA 16
2.1 Gestão de frotas 16
2.2 Internet das coisas 17
2.2.1 Arquiteturas e componentes da IoT 18
2.3 Identificação de coisas e pessoas 20
2.3.1 Tecnologias para identificação do veículo 20
2.3.2 Tecnologias para identificação de pessoas 23
2.4 Banco de dados 24
2.4.1 Níveis de abstração de banco de dados 25
2.4.2 Acesso ao BD via RESTful 26
2.5 Trabalhos relacionados 28

3 DESENVOLVIMENTO 30
3.1 Levantamento de requisitos 30
3.1.1 Análise de requisitos e casos de uso 30
3.2 Arquitetura da solução 33
3.2.1 Backend do Gerenciador do Sistema 34
3.2.2 Frontend 36
3.2.3 API RESTful 42
3.2.4 Backend do Subsistema de Identificação 42
3.2.5 Dispositivos Conectados ao Sistema 43
3.2.5.1 Câmera IP 44
3.2.5.2 Hardware da bomba de combustível 44
3.2.5.3 Hardware da cancela 45

4 RESULTADOS 46
4.1 Interações do Usuário com o Sistema 46
4.1.1 Etapas para saída do veículo 46

5 CONCLUSÃO 54

5.1 Trabalhos futuros 54

REFERÊNCIAS 56

APÊNDICES 58

APÊNDICE A – API RESTFUL 59

1 INTRODUÇÃO

A gestão de frotas é essencial em uma que empresa utiliza veículos para a sua atividade fim. Em razão disso, é importante ter atenção a segurança dos motoristas que atuam no deslocamento dos veículos, aos custos operacionais, a manutenção e a logística de transporte. Para um serviço adequado, se faz necessário buscar por soluções que auxiliem nesta tarefa.

O gestor responsável por administrar e controlar a frota automotiva enfrenta diversos desafios no cotidiano, e várias informações importantes para que o controle seja mantido não estão disponíveis com facilidade ou são dependentes de terceiros. Desse modo, o controle da entrada e saída da frota para utilização em serviços ou para manutenções programadas dos veículos acaba sendo uma dificuldade enfrentada no gerenciamento de frotas (VALENTE et al., 2016).

A Internet das Coisas (IoT) é uma tecnologia capaz de beneficiar a gestão de frotas, pois permite a coleta de dados em tempo real com relação aos veículos, as condições de tráfego e dos motoristas que estão em atuação. Para um controle mais eficiente das frotas podem ser usados sensores e dispositivos IoT, por exemplo, para a geração de informações precisas com relação as frotas em atividade.

A IoT, segundo Jara et al. (2014), tem em sua composição objetos inteligentes cuja diferenciação se dá pelo fato de serem dispositivos físicos e restritos no que diz respeito a sua capacidade de computação, memória, comunicação e autonomia de energia. Formados tanto por códigos quanto por etiquetas de identificação, esses dispositivos possibilitam que algo em específico seja identificado de maneira única.

Vermesan et al. (2011) explica que a internet das coisas consiste em uma infraestrutura de rede global e dinâmica, cuja capacidade é de autoconfiguração com base nos protocolos de comunicação interoperáveis e padronizados em que as “coisas” virtuais e físicas possuem personalidades virtuais, atributos físicos, identidades, utilizam interfaces inteligentes e são totalmente integradas à rede de informação.

Empresas que dependem diariamente da utilização de veículos enfrentam a necessidade de designar um responsável para coordenar o uso desses recursos, assegurando que somente indivíduos autorizados tenham acesso aos mesmos. Adicionalmente, surge a exigência de identificar o condutor associado a infrações de trânsito, bem como de reservar ou indisponibilizar um veículo para uso exclusivo de determinada pessoa durante um período predefinido.

Essas demandas refletem os desafios operacionais inerentes à gestão de frotas, uma

atividade complexa que envolve não apenas a operacionalização dos veículos, mas também a implementação de políticas e procedimentos eficazes para garantir o uso adequado e seguro dos recursos da empresa. A necessidade de coordenação e controle torna-se evidente, especialmente em ambientes empresariais onde múltiplos veículos são utilizados regularmente para diversas finalidades.

A designação de um responsável pela gestão da frota e a implementação de sistemas de controle e monitoramento são estratégias fundamentais para lidar com esses desafios. Através da utilização de tecnologias avançadas, como sistemas de identificação biométrica, bloqueio em tempo real com o uso de IoT e identificação de veículos, é possível aumentar a eficiência e a segurança das operações de frota, ao mesmo tempo em que se reduz o risco de abusos ou irregularidades.

Neste contexto, este trabalho visa atender estas demandas de modo que seja implementado um sistema de gestão de frotas integrado a dois sistemas de identificação (pessoa e veículo) juntamente a um sistema de controle do veículo contando com a tecnologia IoT no qual será empregado uma EPS 32 para que seja possível realizar o bloqueio dos veículos de forma remota. A autora utilizará o Destacamento de Controle do Espaço Aéreo de Florianópolis (DTCEA-FL), seu local de trabalho, como caso de uso para o desenvolvimento do sistema.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um sistema computacional que permita reservas e controle da entrada e saída de veículos, visando a resolução de problemas relacionados a gestão de frotas.

1.1.2 Objetivos Específicos

- Identificar as tecnologias que contribuem para o acesso com maior rapidez a informações de gestão de frotas;
- Especificar e validar mecanismo para identificação de pessoas e veículos viáveis para gestão de frotas;
- Especificar e validar um sistema computacional para reserva e controle da entrada e saída de veículos.

1.2 Estrutura do trabalho

Esse trabalho seguirá com os seguintes capítulos. O Capítulo 2 que é a fundamentação teórica, apresentando toda informação necessária para a compreensão da tecnologia que será aplicada e trará uma pesquisa de mercado para verificar soluções existentes. Em seguida, serão apresentadas no Capítulo 3 a concepção e implementação do sistema, incluindo componentes de *hardware* e *software*. No Capítulo 4 serão apresentados todos os resultados obtidos do desenvolvimento do sistema, e por fim o Capítulo 5 apresenta as conclusões do projeto, bem como sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Gestão de frotas

A gestão de frotas tem o papel importante de gerenciar os custos de combustível, de manutenção, de aquisição e renovação dos veículos, assegurando que todos os seus processos aconteçam de forma harmoniosa com as solicitações dos usuários, não ocorrendo atrasos ou mesmo a falta de atendimento.

De acordo com Valente, Passaglia e Novaes (1997), a gestão de frotas é importante para o planejamento, o gerenciamento e a execução de atividades relacionadas aos veículos de uma instituição, sendo exigido do gestor um conhecimento teórico e prático das tarefas que englobam os equipamentos, as tecnologias, os veículos e a mão de obra existentes para o auxílio do controle e da execução desta gestão.

Na gestão são utilizados alguns métodos, técnicas e ferramentas variadas, como softwares, que permitem que as empresas eliminem e/ou minimizem riscos inerentes ao investimento de seus veículos, ampliando a eficiência e a produtividade dos seus serviços (SALLES, 2011).

O controle de forma informatizada viabiliza o encontro de dados gerenciais com confiabilidade, porém poderá ter como resultado um fracasso informatizado caso o sistema seja ineficiente (CAMPOS; BELHOT, 1994). Contudo, podem acontecer vários problemas na gestão, como a depreciação e a manutenção dos veículos, o preço dos combustíveis, os seguros dos veículos, os pagamentos dos motoristas, a concorrência, e os custos de transporte elevados. Além disso, ainda existe a falta de zelo pelos condutores, aumentando o desgaste natural dos veículos, usados por diversas horas seguidas por várias vezes (CLEMENTE, 2013).

COTTI (1989) explica que uma gestão de frotas eficaz não requer processos sofisticados ou complexos, porém, por diversas vezes se tornam necessárias decisões e análises conduzidas pela preocupação real e o bom senso para a melhoria do rendimento de cada dia para que alcance excelentes resultados.

O cenário passar por modificações e se adapta as inovações que trazem facilidade para o dia a dia do gestor. Para ajudar no trabalho de gestão, existem softwares que fazem a contabilização da manutenção, do abastecimento, da demanda dos serviços, dos seus itinerários, dos tipos de utilização dos veículos e da depreciação dos veículos, viabilizando um controle maior das demandas.

Compreende-se que a gestão de frotas consiste na atividade de gerenciar, administrar

ou reger um conjunto de ativos, em específico, os veículos de uma organização ou empresa. A tarefa é bastante ampla e passa por vários serviços, como dimensionamento, cálculo de custos, manutenção mecânica ou roteirização, por exemplo. A tarefa de fazer o cálculo preciso de um modelo de participação da frota nos custos e no patrimônio das instituições não é algo considerado simples, visto que essa parcela tem variação de caso para caso e depende dos propósitos da empresa, por exemplo, se a carga é de passageiros, se é própria, de terceiros; da maneira como são contabilizados e inseridos os veículos na frota, por meio de *leasing*, aluguel ou financiamento; e da natureza da operação (VALENTE; PASSAGLIA; NOVAES, 1997).

De acordo com Bertaglia (2017), as empresas que possuem envolvimento com o serviço de transporte tanto com veículos próprios quanto com terceirizados, precisam fazer o controle e ter conhecimento de modo efetivo dos seus ativos e seguir práticas de negócios importantes, visto que é a melhor maneira de manter a competitividade e diminuir seus custos logísticos e, consequentemente, o preço do produto final, ficando clara a relevância da administração correta. Um dos elementos considerados mais relevantes na administração de transporte é a gestão das frotas, especialmente se for a sua atividade final da cadeia de abastecimento. Em diversas situações, a movimentação de carga possui um peso considerável na qualidade do serviço prestado e na formação dos custos logísticos. No decorrer dos anos, as empresas que tinham frotas menores tiveram um crescimento e passaram a ampliar a quantidade de veículos. Entretanto, isso não significa que a sua operação continuou sendo feita de modo mais eficiente, o que pode, em diversos casos, aumentar os custos de modo desnecessário e excessivo, ou mesmo perder a qualidade na prestação do serviço (BERTAGLIA, 2017).

A gestão de frotas tem vários aspectos importantes que necessitam receber atenção, como a inutilização da carga, indicadores de produtividade e desempenho, eficiência da frota, planejamento de atividades administrativas, estoque de peças de reposição, manutenção mecânica, consumo de combustível, mão de obra, administração dos ativos, dentre outros. A combinação desses fatores justifica a utilização da tecnologia da informação para administrar os transportes. Com o progresso da tecnologia, várias ferramentas importantes foram elaboradas para o fornecimento de suporte para as tarefas de processo de transporte, como a telemática e os sistemas de otimização de rastreamento de frota e de rotas, por exemplo (BERTAGLIA, 2017).

2.2 Internet das coisas

A internet trouxe mudanças para o cotidiano do mundo todo, possibilitando o acesso rápido a informações que não eram acessíveis até aquele momento. Porém, se a internet “das pessoas” pode ser vista como uma revolução, a internet “das coisas” (IoT)

pode permitir muito mais (SANTOS et al., 2016).

A IoT é considerada um conceito onde os objetos e os dispositivos do cotidiano são equipados com sensores que possuem a capacidade de manterem uma comunicação entre si de modo inteligente. (SANTOS et al., 2016) explicam que os sensores possuem responsabilidade pela coleta de informações com relação ao contexto no qual os objetos estão localizados, fazendo o armazenamento dos dados e podendo realizar a manipulação do ambiente ou ter uma reação conforme os dados lidos. Estes dispositivos monitoram o ambiente em que está localizado o objeto fazendo a captura dos valores das grandezas físicas como presença, pressão, umidade e temperatura. Nessas aplicações há diversos sensores distintos com capacidade de captura dessas grandezas, além de poder realizar a produção de alguma ação, com o acionamento de comandos que podem ser mecânicos, elétricos ou manuais.

Em sua essência, a IoT é somente um ambiente que faz a reunião de informações de diversos dispositivos (semáforos, smartphones, veículos, computadores, dentre outros) e de aplicações (desde um sistema de controle de tráfego a uma aplicação de mídia social)(SANTOS et al., 2016).

2.2.1 Arquiteturas e componentes da IoT

A definição de Uviase e Kotonya (2018) descreve a arquitetura *Internet of Things Architecture* (Iot-A) como uma estrutura que engloba a especificação dos componentes físicos conectados em rede, a organização e configuração funcional desses componentes, bem como os princípios e procedimentos operacionais associados. Além disso, a arquitetura *Internet of Things Architecture* (Iot-A) também estabelece os formatos de dados utilizados na operação desses componentes. Essa definição destaca a importância de uma arquitetura sólida e bem definida visando garantir a interoperabilidade e a integração adequada dos dispositivos conectados em uma rede IoT. Ela é responsável por fornecer diretrizes e padrões para a organização e comunicação entre os dispositivos, bem como para a estruturação dos dados que são trocados e processados.

Para compor os diversos modelos de Iot-A existem 5 tipos de camadas diferentes (SATYABRATA, 2022) são elas:

- Camada de Percepção: Nesta camada é onde encontram-se sensores e aturadores, sua principal função é obter informações dos arredores e passar os dados para outra camada como suporte para ações que possam ser tomadas (Radio Frequency Identification (RFID)).
- Camada de Rede: é a camada que recebe os dados vindos da camada de aplicação e envia para a camada *middleware* através de rede 3G, 4G, 5G, WiFI, entre outros. É nela que há a garantia de transferência de dados de forma segura.

Figura 1 – Arquitetura de três camadas.

Fonte:Maschietto et al. (2021)

- Camada de *middleware*: Responsável por armazenar os conjuntos de dados com base no nome e endereço do dispositivo, composta de recursos de tomada de decisão com base em cálculos, processamento e computação.
- Camada de Aplicação: É a camada responsável pelo gerenciamento de todos os processos de aplicativo com base nas informações passadas pelo *middleware*.
- Camada de Negócios: Trata-se de de uma camada responsável por fazer melhor uso dos dados, trazendo fluxogramas, gráficos, análise de resultados, etc.

Segundo (GLUHAK et al., 2011) a maioria dos experimentos em IoT seguem uma estrutura de duas ou três camadas. A arquitetura de duas camadas é composta pela a camada de percepção e de aplicação. Já a de três camadas, conforme Figura 1, é acrescentado a camada de *middleware* visando a garantida da segurança dos dados. Destaca-se o fato de que essa arquitetura possui uma comunicação restrita e altamente interligada com as camadas de percepção e *middleware*. Sua dependência direta dessas camadas, sem uma camada de rede desacoplada, acaba impactando negativamente a eficiência e escalabilidade das comunicações na IoT.

Observa-se que quanto mais camadas são adicionadas à arquitetura, mais complexa vai se tornando. Em contra partida oferece ao cliente um programa de maior qualidade. Na arquitetura de quatro camadas surge a camada de rede, aumentando o leque de possibilidades de implantação tendo em vista que passa a se ter a Internet como meio de envio de dados, ou seja, a estrutura se baseia na comunicação da camada de percepção com a de rede, em seguida passado para a de *middleware* e ao final a de aplicação.

Por fim, temos a arquitetura de cinco camadas na qual é adicionada a camada de negócio, que é uma camada criada com o intuito de personalizar mais a aplicação, visando utilizá-la para, através das análises de dados obtidas, sugerir melhoras por exemplo. Vale ressaltar que alguns autores colocam essa camada depois da de aplicação e outros antes conforme a Figura 2.

Figura 2 – Arquitetura com cinco camadas.

Fonte: (MASCHIETTO et al., 2021)

2.3 Identificação de coisas e pessoas

A identificação de coisas e pessoas é um fator importante em determinadas áreas, como controle de acesso, autenticação de usuários e controle de estoque. Em cenários não digitalizados tudo é feito de forma manual, o que torna os processos lentos e muitas vezes ineficazes. Nos últimos anos, a tecnologia trouxe avanços significativos que contribuíram na área de controle de acesso, por exemplo, tornando-os capazes de auxiliar na identificação de forma eficiente e confiável. As tecnologias *Quick Response Code* (QR Code), Radio Frequency Identification (RFID), reconhecimento da placa, facial, digital e de íris são exemplos dessas tecnologias.

2.3.1 Tecnologias para identificação do veículo

O *Quick Response Code* (QR Code) foi inventado por um empresa japonesa chamada Denso-Wave em 1994. Naquela época surgiu a necessidade de que a catalogação de peças fossem realizadas de forma mais rápida. Com isso, desenvolveram uma versão bidimensional do código de barras capaz de armazenar caracteres numéricos, alfanuméricos, especiais e pontuações. Para realizar a leitura dos QR Codes basta um aparelho celular com câmera. Atualmente, muitos dispositivos já vem com o leitor no próprio celular, caso contrário, basta baixar um dos vários aplicativos disponíveis nos sistemas Android e iOS.

Os QR Codes se dividem em dois tipos, o estático que armazenam informações "fixas" e permanentes, no qual as informações são codificadas diretamente no código e não podem ser modificadas posteriormente sem que seja gerado um novo código; e dinâmico,

que é um código que contém informações que podem ser alteradas/atualizadas sempre que necessário sem a necessidade de ser gerado um novo código. O uso de cada um deles depende do que se deseja. O estático é voltado para rótulos de produtos com links para informações estáticas, ou quando se quer deixar uma informação não muito extensa para ser compartilhada com um público de modo que ela não precise ser alterada, já o dinâmico por ter a flexibilidade em seu conteúdo e não necessidade de reemissão de um novo código, é voltado ao uso em redes sociais, cupom de descontos, chaves pix entre outros. A viabilidade do QR Code é possibilitada através de plataformas de gerenciamento de QR Codes dinâmicos, elas possuem ferramentas necessárias desde a criação, atualização até seu rastreamento de desempenho e análise, essas duas ultimas ferramentas, são muito úteis pois permite saber a quantidade de vezes que o código foi escaneado.

Um outro dispositivo muito útil na identificação de coisas é o RFID, considerado como uma variação do código de barras, mas que dispensa o contato visual para identificação já que ela é feita por radiofrequência. Os dados são codificados em etiquetas RFID e, para realizar a identificação, as antenas capturam as ondas de rádio.

O RFID trabalha com quatro faixas de frequência: *Low Frequency* (LF), *High Frequency* (HF) (13,56MHz), *Ultra High Frequency* (UHF) que possuem as seguintes aplicações nas bandas respectivamente, de 100 a 500KHz, 10 a 15MHz, 850 a 950MHz e 2,4 a 5,8 GHz. Cada uma delas tem um uso específico sendo levado em consideração suas vantagens e desvantagens.Para o monitoramento de veículo em trânsito, o mais indicado é a UHF devido a sua larga faixa de leitura e alta velocidade de leitura, tal fato se dá pelo uso de frequências mais altas Teleco (2023).

Além das tecnologias citadas anteriormente, tem-se também, o sistema de reconhecimento automático de placas, denominado subconjunto do sistema de identificação veicular. Eles combinam software e hardware para realizar diversas funções essenciais. Esses sistemas têm como objetivo identificar a passagem de veículos, registrar imagens dos veículos capturando suas placas, reconhecer e converter o conteúdo da placa em caracteres processáveis, tomar decisões em com base nos dados coletados, transmitir dados e imagens, e tratar as informações de maneira adequada. Com isso, esses sistemas dependem não apenas das tecnologias utilizadas para identificar e processar as imagens, mas também da qualidade e funcionalidade do conjunto de soluções adotadas para implementá-los (BERNARDI et al., 2015).

A Tabela 1 fornece uma análise comparativa das tecnologias potenciais para a identificação de veículos, considerando diversos critérios, como distância média de leitura, segurança, velocidade de leitura, custo e durabilidade. Com base nas necessidades específicas do projeto, que incluem uma leitura em distância média para otimizar a dinâmica do processo de identificação, bem como a prioridade de garantir uma boa segurança, a escolha mais indicada é o reconhecimento de placa. Além de oferecer a

Tabela 1 – Tabela Comparativa: QR Code, RFID e Reconhecimento de Placa para Identificação de Veículo.

QR Code	**RFID**	**Reconhecimento de Placa**
	Princípio de Funcionamento	
Codificação de dados em código de barras bidimensional.	Comunicação sem fio por radiofrequência entre tags e leitores.	Captura e processamento de informações a partir da imagem da placa do veículo.
	Distância de Leitura	
Curta distância, até alguns centímetros.	Pode variar de centímetros a metros, dependendo do tipo de RFID.	Dependente da qualidade da câmera, geralmente média a curta distância.
	Velocidade de Leitura	
Rápida, pois basta escanear o código com a câmera.	Rápida, a leitura é instantânea quando a tag está na faixa de alcance.	Pode variar, mas geralmente é rápida o suficiente para aplicações de identificação veicular.
	Custo	
Baixo custo para implementação de QR Codes, a depender do material utilizado.	Moderado, dependendo do tipo de RFID utilizado.	Pode variar, dependendo do sistema de câmeras e software utilizado.
	Durabilidade	
Sensível a danos físicos, rasuras ou sujeira.	As tags RFID podem ser mais robustas, mas o leitor pode ser danificado.	Sujeita a condições climáticas e desgaste das placas.
	Segurança	
Relativamente menor em comparação com RFID.	Segurança razoável, risco de clonagem	Seguro, porém não identifica placa clonada.
	Aplicações Comuns	
Bilhetagem em transportes públicos, pagamentos móveis, etc.	Controle de acesso, gestão de estoques, pedágios, etc.	Controle de estacionamentos, segurança pública, monitoramento de tráfego, etc.

Fonte: Do Autor.

capacidade desejada de leitura em distâncias médias, o reconhecimento de placa destaca-se pela velocidade eficiente na captura e processamento de informações. Embora o custo possa ser comparável a algumas soluções, a durabilidade associada ao reconhecimento de placa é notável, especialmente quando consideramos a exposição às condições climáticas e o desgaste natural das placas dos veículos. Adicionalmente, a tecnologia de reconhecimento de placa apresenta um nível satisfatório de segurança e a flexibilidade para implementação

de algoritmos avançados, agregando valor à solução global proposta para a identificação de veículos neste contexto específico.

2.3.2 Tecnologias para identificação de pessoas

O conceito de biometria tem suas raízes etimológicas na língua grega, onde *bios* significa vida e *metron* se refere a medida. No contexto dos sistemas informatizados, a biometria se refere ao uso de características exclusivas de um indivíduo para autenticar e/ou identificar sua presença perante um sistema pertencente a uma organização (MATTOS; BEGOSSO, 2018).Essas características são utilizadas para criar um perfil biométrico que permite reconhecer e verificar a identidade de uma pessoa de forma precisa e confiável. Existem diversas formas de reconhecimento por biometria, seja ela através da digital, reconhecimento facial, reconhecimento por voz, pela íris, entre outros. O fato é que esse tipo de estudo é antigo e vem cada vez mais sendo utilizado no dia-a-dia, seja no controle de acesso ou na segurança pública utilizando sistemas de busca facial. A questão é que as vastas possibilidades abrem margem a dúvidas sobre qual seria a melhor opção.

O reconhecimento facial se baseia em um software que faz a captura do rosto da pessoa e analisa em 2D. Em seguida ele observa a distância entre os olhos, a profundidade das órbitas oculares, a distância entre a testa e o queixo – dados que se julgam particulares de cada pessoa. Após a captura dos dados, o programa realiza uma espécie de extração onde converte a imagem em um código único, para que quando o usuário for se autenticar, ele realizar uma comparação com a imagem obtida na autenticação com o banco de dados. A grande questão deste tipo de reconhecimento é que ele não é 100% preciso. Segundo uma matéria divulgada pelo site Digital (2018), em uma partida da *Champions League* o sistema de reconhecimento facial da polícia errou em 92% ao reconhecer potenciais ameaças, ou seja, ele identificou pessoas inocentes.

O reconhecimento através da íris possui algumas etapas. Inicialmente é feita a captura da imagem da íris, em seguida os dados são processados visando separar a imagem da íris das outras imagens do resto do olho. Após o processamento, ocorre a codificação, onde a íris capturada gera um código único que corresponde a ela. Para realizar o reconhecimento, é utilizado um teste de independência estatística entre dois códigos de íris diferentes. A probabilidade de erro deste tipo de reconhecimento é muito baixa, porém possui limitações: ele não consegue fazer um reconhecimento de um fluxo grande de pessoas de forma ágil; e a distância do identificador para os olhos é curta(VALADÃO, 2008).

A impressão digital é formada pela o conjunto de elevações da pele, conhecida como papilas, presentes nos dedos. Foi um dos primeiros tipos de autenticação utilizados devido a sua unicidade. Antes feito por peritos por inspeção visual, atualmente são utilizados software que contam com um amplo banco de dados para armazenar essas informações. O

processo de reconhecimento digital segue exatamente as mesmas etapas que os demais: obtenção dos dados, criação de um código correspondente, armazenamento em banco de dados. Para a validação é feito uma verificação da informação obtida com a presente no banco. Esse método é considerado um dos mais seguras. Em contra partida, suas desvantagens são que sujeiras, dedos machucados, ou alguma condição que afete a pele dos dedos acabam por atrapalhar o reconhecimento. Por exemplo, no desbloqueio de celulares, acontecem casos de ter que repetir a autenticação por algum erro na leitura.

O RFID é uma tecnologia versátil para identificação e na autenticação do usuário de um sistema de informação, utilizando *tags* em formatos variados, como chaveiros, cartões, botões, moedas, discos e pulseiras, que são lidos por leitores RFID. Essa tecnologia oferece diversas vantagens nessa aplicação, sendo conveniente ao eliminar a necessidade de senhas ou cartões físicos. Além disso, os leitores RFID podem capturar as informações das *tags* de forma ágil, tanto em curtas distâncias como em movimento, normalmente pare este tipo de aplicação usa-se as bandas *Low Frequency* (LF).

Após um breve estudo sobre os tipos de identificação que podem estar sendo utilizados, a opção escolhida para implementação foi o reconhecimento facial. Essa escolha foi fundamentada em vários fatores cruciais para as necessidades do projeto.

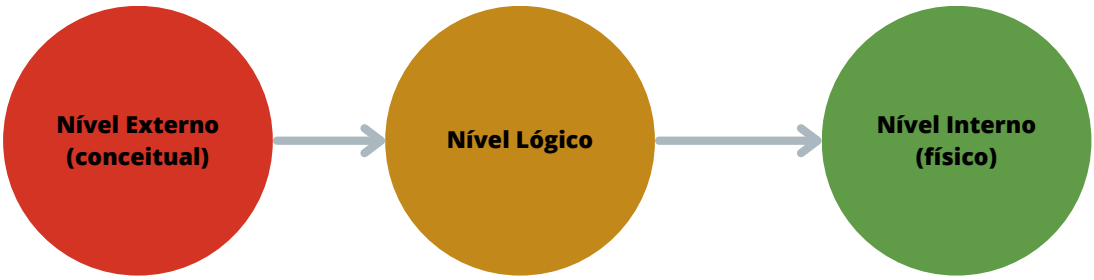
Em primeiro lugar, o custo moderado associado ao reconhecimento facial torna essa tecnologia uma opção atrativa, proporcionando uma solução eficaz sem comprometer significativamente o orçamento. Além disso, o reconhecimento facial oferece uma camada adicional de segurança, contribuindo para evitar fraudes e burlas ao sistema, uma vez que é mais desafiador replicar características faciais únicas em comparação com outros métodos. Outro ponto relevante é a conveniência proporcionada pelo reconhecimento facial, eliminando a necessidade de carregar objetos adicionais, como cartões ou dispositivos biométricos. Isso simplifica o processo de identificação e melhora a experiência do usuário, promovendo uma abordagem mais natural e acessível.

Em resumo, a escolha do reconhecimento facial é respaldada pela sua eficácia, relação custo-benefício favorável, resistência à falsificação e pela praticidade inerente ao método. Essa tecnologia emerge como uma solução abrangente e equilibrada para as demandas específicas do projeto de identificação de pessoas.

2.4 Banco de dados

Os bancos de dados surgiram como uma solução para lidar com a crescente necessidade de armazenar dados de maneira permanente e organizada. Com o contínuo aumento no volume de dados gerados, tornou-se essencial ter um sistema capaz de gerenciar essas informações de forma eficiente. Foi nesse contexto que os Sistemas de gerenciamento de banco de dados (SGBDs) entraram em cena, desempenhando um papel fundamental na

Figura 3 – Etapas de modelagem do projeto de banco de dados.



Fonte: Do Autor.

extração de informações relevantes para a tomada de decisões. Esses sistemas garantem a integridade dos dados, possibilitam a recuperação rápida e eficiente das informações armazenadas e oferecem recursos avançados para consulta, análise e manipulação dos dados. Dessa forma, os Sistemas de gerenciamento de banco de dados (SGBDs) se tornaram peças essenciais na gestão de dados em diversas áreas, contribuindo para a eficiência e o sucesso das organizações no mundo moderno.Atualmente, existem banco de dados do tipo: relacional, orientado a documentos, em memória, orientada a objetos, onde cada um tem uma característica específica.

2.4.1 Níveis de abstração de banco de dados

Os níveis de abstração de banco de dados são conceitos utilizados para descrever diferentes perspectivas e camadas de um banco de dados. Eles representam diversos níveis de detalhamento e complexidade nas operações e interações com o banco de dados. Esses níveis são utilizados durante o desenvolvimento de um projeto de banco de dados e existem três deles (HEUSER, 2011), conforme pode ser visto na Figura 3.

O nível externo, também conhecido como conceitual, é onde o usuário expõe suas necessidades e define como os dados serão armazenados e relacionados. O nível lógico é o próximo passo no desenvolvimento do projeto, no qual é criado, com base no modelo conceitual, o tipo de banco de dados a ser utilizado, como o Oracle, por exemplo. E o outro nível é o interno, também conhecido como físico, onde são definidos os tipos de dados a serem armazenados e a implementação da lógica do SGBD.

Ao iniciar o desenvolvimento de um novo sistema, é essencial realizar um levantamento dos requisitos necessários para construir o produto final. Essa análise permite identificar os objetos envolvidos, suas responsabilidades, atributos e seus relacionamentos. Com base nessas informações, surge o modelo conceitual, que é utilizado como guia no desenvolvimento e garante o cumprimento dos requisitos e funcionalidades desejadas.

Após a fase conceitual é necessário avançar para o nível lógico, onde encontramos a modelagem relacional. Em seguida vem o projeto físico onde são levadas em consideração

particularidades do SGBD que será utilizado na implementação do banco de dados (ELMASRI; NAVATHE, 2011).

2.4.2 Acesso ao BD via RESTful

Antes de conhecer um pouco sobre o que é a API *RESTful*, serão abordados alguns conceitos e nomenclaturas importantes para entender a importância da API no desenvolvimento de aplicações *Web*.

Quando o programador desenvolve um *software* é comum dividi-lo em duas partes denominadas *frontend* e *backend*, onde cada uma tem características e responsabilidades distintas, porém são dependentes uma da outra para o correto funcionamento da aplicação.

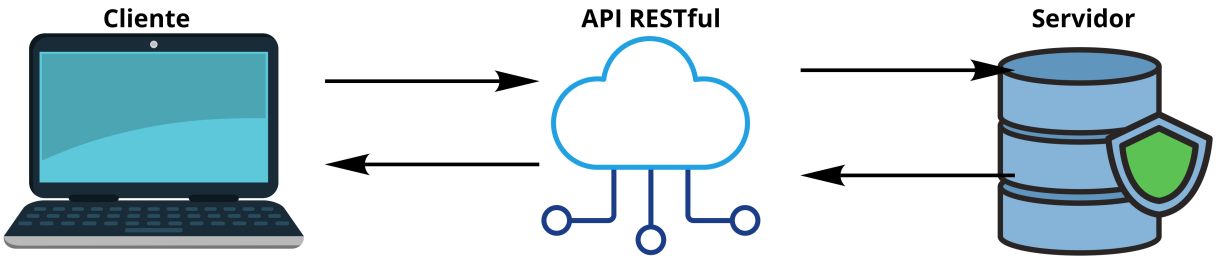
Define-se *frontend* como sendo a parte da aplicação voltada para a interação direta do usuário. É a parte responsável por exibir e gerenciar a interface gráfica. Quanto mais intuitiva for, maior a facilidade do usuário na utilização do programa. O foco dessa parte é o *design*, interação, dinamismo e outros fatores que afetem diretamente o usuário. Já o *backend* é a parte da aplicação responsável por desenvolver a lógica, o processamento e a comunicação com o servidor, incluindo todas as requisições feitas pela usuário, autenticação, segurança, regras de negócio e manipulação do banco de dados.

A API *RESTful* desempenha um papel fundamental na integração de dois sistemas, sendo que eles a usam para trocar informações através da internet de modo seguro. Grande parte das aplicações necessitam se comunicar com outras aplicações externas. Ela define as regras de comunicação entre os *software* (Amazon Web Services, 2023). A API *REST* define um conjunto de princípios e diretrizes para a criação de serviços web baseados no *Hyper Transfer Protocol* (HTTP). Ela utiliza os verbos HTTP, como GET, POST, PUT e DELETE, para realizar operações de criação, leitura, atualização e exclusão de dados (*Create, Read, Update, Delete* (CRUD)).

Analisando o cenário cliente-servidor conforme a Figura 4, o *frontend*, que nesse caso é o cliente, faz requisições à API *REST* para obter ou enviar informações ao *backend*. Tais requisições podem ser feitas através de URLs definidas pela API ao qual correspondem a recursos específicos da aplicação. Já o servidor (*backend*), primeiramente autentica o cliente e dá uma resposta para o cliente informando que ele está autorizado a realizar requisições. Então, o *backend* recebe as requisições, processa-as e retorna as respostas contendo informações indicando se a solicitação foi ou não atendida. Esses respostas vão desde uma consulta a um banco de dados à formatação de resposta de acordo com o formato solicitado, e variam de acordo com o o que foi projetado a API.

Conforme o site Amazon Web Services (2023), existem três benefícios das APIs *RESTful*, que são:

Figura 4 – Fluxo Cliente-Servidor através da API.



Fonte: Do Autor, utilizando Canva.

- Escalabilidade: Capacidade de um sistema ou aplicação lidar com o aumento na demanda de recursos ou usuários, mantendo o desempenho, confiabilidade e qualidade do serviço. A API *REST* otimiza as interações entre cliente e servidor. A ausência de estado reduz o ônus sobre o servidor, pois ele não precisa armazenar informações de solicitações anteriores dos clientes. Além disso, um cache bem gerenciado pode eliminar parcial ou totalmente certas interações entre cliente e servidor. Esses recursos combinados possibilitam escalabilidade sem criar gargalos de comunicação que afetam negativamente o desempenho do sistema.
- Flexibilidade: Por serem dois sistemas diferentes, tem-se a liberdade de uma constante evolução no desenvolvimento das aplicações, bem como emprego de uma nova tecnologia no *backend* ou *frontend* sem que um afete o outro.
- Independência: O tipo de linguagem de programação aplicado ao cliente não tem qualquer tipo de vínculo com a do servidor, o que torna as duas aplicações independentes uma da outra nesse quesito.

A solicitação do Cliente precisa conter o identificador de recurso exclusivo, no caso o URL, no qual ele especifica qual o caminho para o recurso e o método, ou seja, a especificação do que o cliente de fato deseja. Isto é realizado usando o *Hypetext Transfer Protocol* (HTTP), e são quatro métodos possíveis:

- *GET*: Quando o cliente quer alguma informação que está disponível no servidor.
- *POST*: Utilizado quando o cliente quer salvar alguma informação no servidor.
- *PUT*: Quando já há algum dado salvo no servidor porém o cliente quer fazer alguma alteração nele.
- *DELETE*: Método utilizado para excluir algum dado contido no servidor.

No quesito segurança, a API também tem seus pontos fortes. Ela possui alguns tipos de autenticação de HTTP: através da autenticação básica, ou seja, o cliente informa

o usuário e a senha no cabeçalho; a do portador, que é com o uso de um *token*; chaves de API, onde o servidor designa uma valor exclusivo para ser usado como chave a um cliente; e o OAuth, que é a combinação de senha com *tokens* e, por ser uma autenticação de dois fatores, é considerada a mais segura.

Assim como as requisições têm uma estrutura, as respostas também possuem. Elas são compostas por três itens:

- Linha de *status*: Contém a informação se foi bem sucedido ou não a requisição.
- Corpo da mensagem: Onde contém a representação da requisição. Normalmente são no formato JSON ou XML.
- Cabeçalhos: Fornecem mais dados sobre a resposta como codificação, data, tipo de conteúdo entre outros.

A ferramenta estuda acima é de grande valia, mas vale ressaltar que nem sempre é obrigatório seu uso. Um exemplo é a comunicação de uma aplicação com um banco de dados em que existe a possibilidade de fazer uso de *plugins* capazes de configurar o *frontend* para comunicar diretamente com o banco de dados. O fato é que, ao optar por esta opção, acaba deixando informações de login e senha "nas mãos do usuário".

2.5 Trabalhos relacionados

Durante a pesquisa de ferramentas para o trabalho, foram encontrados algumas empresas que oferecem produtos com finalidades um pouco similar ao que foi proposto. Na Tabela 2 pode-se ser visualizado um resumo da pesquisa de mercado no qual foram estudadas quatro empresas que oferecem o sistema de gestão de frotas na qual cada uma tem suas peculiaridades. Vale ressaltar que todas trabalham com contrato, ou seja, é estabelecido um período de vigência de utilização do *software*.

Tabela 2 – Comparativo de Soluções Tecnológicas em Gestão de Frotas.

Empresas	Cobli	Infleet	Sofit	GeoTab
Rastreamento em tempo real	√	√	√	√
Gestão de rotas	√	√	√	√
Monitoramento de consumo de combustível	√	√	√	√
Controle de manutenção	√	√	√	√
Solicitação e reserva de veículos			√	
Identificação do motorista	RFID			NFC
Acompanhamento por câmera	√	√		√

Fonte: Do Autor.

Após a análise dos dados obtidos na pesquisa, constatou-se que a empresa mais adequada para atender aos requisitos necessários aos desafios enfrentados pelas empresas na gestão de frotas seria a Sofit, especialmente no que se refere ao item “solicitação de reserva de veículos”. A observação dos demais aplicativos revelou que, em sua maioria, os gestores traçam uma rota para o motorista, e este, por meio do aplicativo, seleciona o veículo a ser utilizado. Esses aplicativos são predominantemente desenvolvidos para gestão de frotas, com foco mais direcionado a empresas que utilizam veículos para atividades como entrega.

O aplicativo Sofit, por sua vez, oferece precisamente dois tipos de solicitações que são fundamentais para a gestão atual: a solicitação de uso da viatura e a solicitação de reserva. No entanto, em relação à identificação, o aplicativo apresentou deficiências, pois não detalha o procedimento de certificação de aprovação. Por exemplo, considere o caso em que o motorista João faz uma solicitação para ir ao centro da cidade com um veículo Uno, placa X. O gestor recebe a solicitação e a aprova. No entanto, como o responsável pela portaria terá conhecimento de que a saída deste veículo está autorizada? Supondo que ele tenha acesso ao aplicativo do gestor e possa visualizar essas informações, aparentemente, apenas o nome do motorista e o veículo constarão. Em uma empresa de grande porte, como o vigilante terá certeza de que o motorista mencionado na planilha é de fato quem está solicitando a saída? Neste ponto específico, o aplicativo não atende a um dos requisitos essenciais do projeto, que é a identificação do motorista e do veículo na entrada e saída, para que seja possível certificar a autorização e registrar a data e horário desses acessos.

Diante dessa análise, conclui-se que é imperativo o desenvolvimento de um software para abordar essa lacuna no sistema, garantindo assim a efetividade no controle de acesso e na identificação do motorista e do veículo nas empresas.

3 DESENVOLVIMENTO

Neste capítulo, apresenta-se uma solução para enfrentar os desafios da gestão de frotas, motivado pela experiência do encarregado das viaturas no Destacamento de Controle do Espaço Aéreo de Florianópolis (DTCEA-FL). Diante das questões de credibilidade e do esquecimento das fichas de viatura, propõe-se a implementação de um sistema integrado às tecnologias da Internet das Coisas (IoT). A proposta envolve o uso do sistema de gerenciamento de frotas contando com o auxílio de um sistema para bloquear a saída de viaturas, através do circuito de alimentação da bomba de combustível e um outro sistema de identificação de veículos e pessoas com o intuito de verificar se ambos estão permitidos a saírem da unidade A ideia é minimizar os erros humanos de modo que, em caso de erro, o sistema irá sinalizar.

Para que seja possível o desenvolvimento do projeto se fez necessário criar uma metodologia, visando identificar os problemas a serem solucionamos, conhecer as características do cliente, buscar ferramentas, arquitetar a estrutura do sistema juntamente com seus componentes. Tais etapas encontram-se detalhadas nas seções abaixo.

3.1 Levantamento de requisitos

O levantamento de requisitos é executado após a definição dos objetivos e requisitos do projeto. Será nele que serão levantados dados sobre a frota de veículos do cliente, dados sobre os usuários, processos e procedimentos de gestão utilizados. É com base nessas informações que surgem ideias de soluções para o desenvolvimento do projeto.

3.1.1 Análise de requisitos e casos de uso

Conforme definido, o levantamento ocorreu no mês de março de 2023, onde foram obtidas informações essenciais que influenciaram na definição deste projeto. Para melhor entendimento das necessidades do cliente, foi realizado um estudo de caso no qual foi feito o levantamento de dez requisitos funcionais do sistema:

1. O sistema deve permitir que cada motorista realize a solicitação de saída da viatura, informando os dados pertinentes;
2. O sistema deve conter uma identificação de todos os usuários e veículos;
3. O sistema deve permitir que o aprovador autorize ou não as solicitações de saída de viatura dos motoristas;

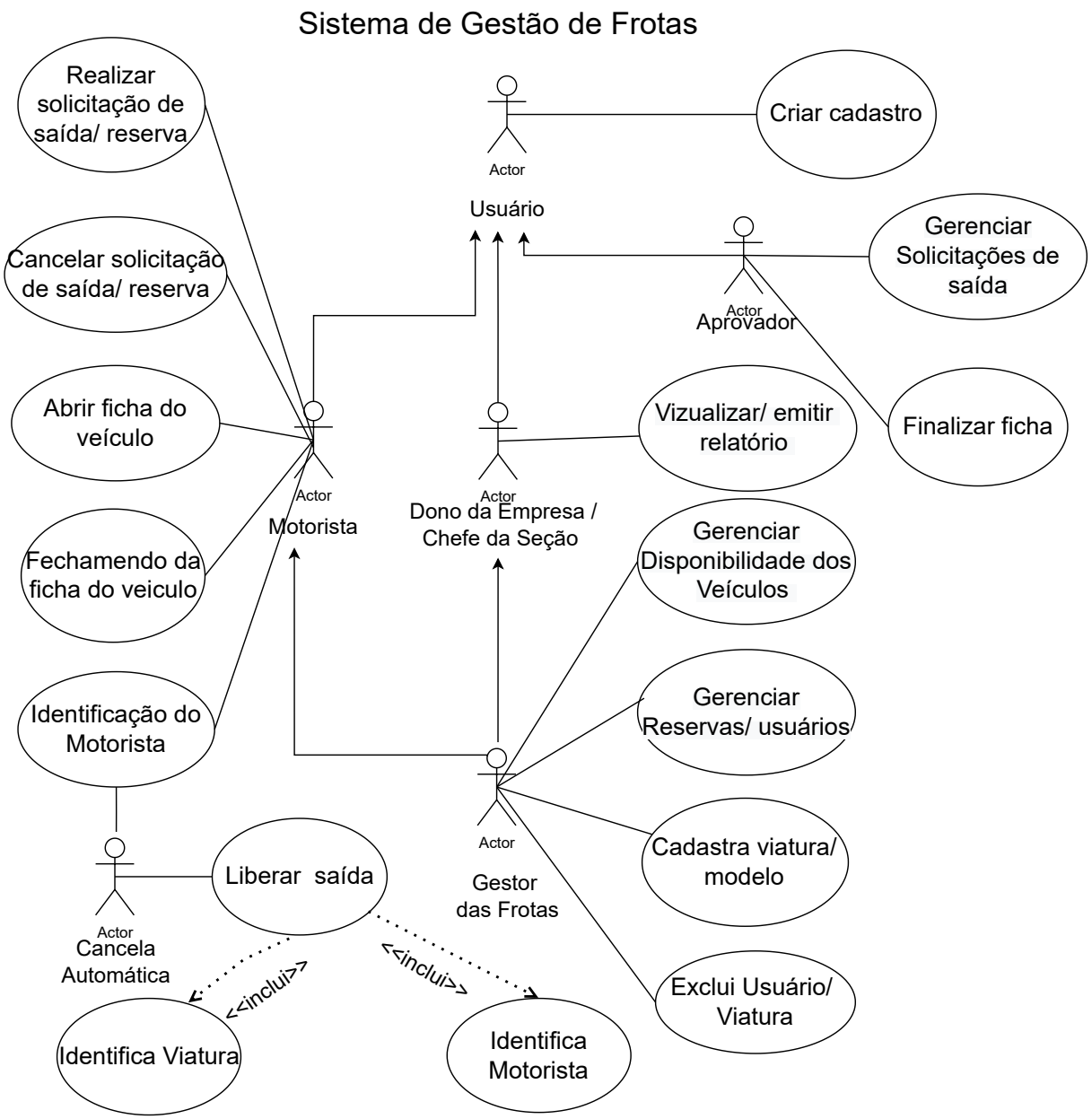
4. O sistema deve permitir que o encarregado das viaturas impossibilite o uso de viatura quando necessário;
5. O sistema deve permitir que cada motorista realize uma solicitação de reserva;
6. O sistema deve permitir que o encarregado autorize ou não a reserva de viatura feita pelo motorista;
7. O sistema deve permitir que o encarregado/comandante/chefe emita e visualize o relatório de entrada e saída das viaturas;
8. O sistema deve permitir que o guarita consiga verificar se a saída ou entrada da viatura está autorizada;
9. O sistema deve permitir que cada motorista possa cancelar a solicitação de saída da viatura e a reserva;
10. O sistema deve registrar a saída, após a identificação do guarita e validação do mesmo quanto à autorização.

A reserva citada nos requisitos acima é utilizada para casos de manutenções programadas na pista e viagens programadas, ou seja, tem por objetivo auxiliar no planejamento de utilização da viatura bem como impedir que ocorra a utilização da mesma por um motorista não autorizado. Vale ressaltar que, independente da reserva estar autorizada, para se ausentar do destacamento se faz necessário abrir uma solicitação de saída.

Além dos requisitos funcionais, foram criados regras de negócio para a solicitação, cancelamento, aprovação, reserva, disponibilidade para reserva, identificação do Aprovador, identificação da viatura, fechamento de ficha, acesso ao relatório, inoperância de viatura, entre outras. Todas essas informações foram utilizadas para a confecção do diagrama de casos de uso conforme Figura 5. A ideia do diagrama é facilitar a visualização do projeto. Foram criados 5 atores:

- Aprovador: Pessoa responsável pelo controle diário das viaturas, o mesmo é responsável por gerenciar solicitações de saída e finalizar as fichas fechadas pelos motoristas;
- Motorista: Pessoa habilitada a conduzir as viaturas. Tem a função de abrir uma solicitação de saída para que, após aprovação do Aprovador, possa fazer uso da viatura. Também pode abrir uma solicitação de reserva que precisa ser aprovada pelo encarregado (Gestor). O mesmo pode fazer cancelamento, tanto do pedido de saída de viatura quanto o de reserva, porém no caso da reserva, a mesma depois que aprovada só poderá ser cancelada pelo encarregado. Ele precisa sempre que sair e

Figura 5 – Diagrama de Casos de Uso.



Fonte: Do Autor, utilizando Draw.io.

retornar se identificar na portaria e, ao término de sua missão, realizar o fechamento da ficha;

- Gestor: Pessoa responsável por gerenciar reservas, acessar/emitir relatórios de entrada e saída das viaturas e por bloquear ou disponibilizar um veículo no sistema de modo que possa estar disponível para os motoristas solicitarem seu uso;
- Cancela Automática: Equipamento responsável pela identificação do motorista e viatura para que, após a consulta no sistema, autorize ou não a entrada ou saída da viatura;

- Comandante ou Chefe da Sessão: Pessoa que pode visualizar ou imprimir o relatório de saída e entrada das viaturas.

3.2 Arquitetura da solução

Este tópico propõe apresentar uma análise da estrutura do sistema proposto, destacando a interconexão de componentes essenciais para o seu funcionamento. Em foco, temos a implementação de um banco de dados *MySQL* e a aplicação Java com *Spring Boot*, que constitui o núcleo do *backend*. Os componentes do sistema foram isolados em contêineres utilizando a ferramenta *Docker*.

O *backend* realiza uma comunicação com um *frontend* em *JavaScript* com *React* que, juntamente com o Sistema de Identificação desenvolvido em *Python* e o sistema de controle da bomba de combustível , são crucias para o entendimento da dinâmica do programa. Além disso, ressalta-se que tanto o *frontend* quanto o *backend* serão executados em contêineres, proporcionando uma padronização que contribui para a estabilidade e escalabilidade do sistema.

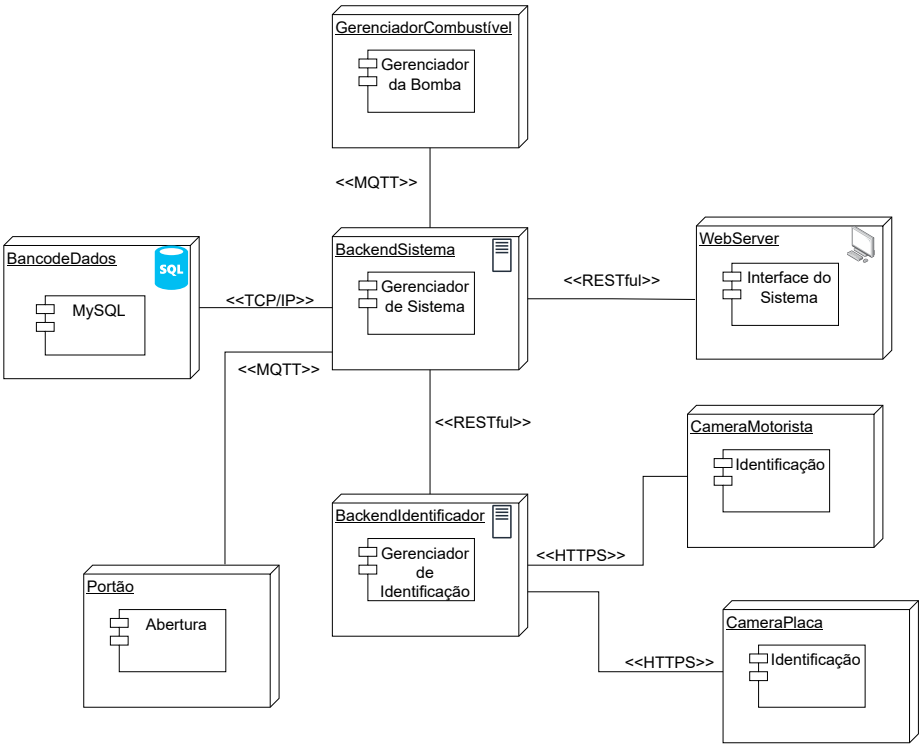
O Sistema de Identificação se comunicará diretamente a duas câmeras *Internet Protocol* (IP) para captura de imagens sob demanda, adicionando uma camada significativa de complexidade à estrutura do programa. A compreensão detalhada dessa interação é vital para avaliar a eficiência global do sistema.

Esta sessão visa oferecer uma visão aprofundada da arquitetura do sistema, identificando pontos de integração e destacando desafios potenciais. Ao analisar a comunicação entre os diferentes componentes, busca-se proporcionar uma compreensão de como cada elemento contribui para o funcionamento eficaz do programa. Essa análise não apenas esclarecerá a estrutura do programa, mas também fornecerá informações valiosas para otimizações e futuros desenvolvimentos. A Figura 6 demostra o diagrama de implantação do sistema.

A IoT-A utilizada será a de 4 camadas, conforme Satyabrata (2022) onde teremos:

- Camada de Percepção composta pelas tecnologias IoT;
- Camada de Rede: No qual teremos a *Wi-Fi* e rede móvel (4G/5G);
- Camada de *Middleware*: composta pelo banco de dados e API *RESTful*;
- Camada de Aplicação: composta pelo aplicativo desenvolvido na plataforma *web*.

Figura 6 – Diagrama de Implantação.



Fonte: Do Autor, utilizando Draw.io.

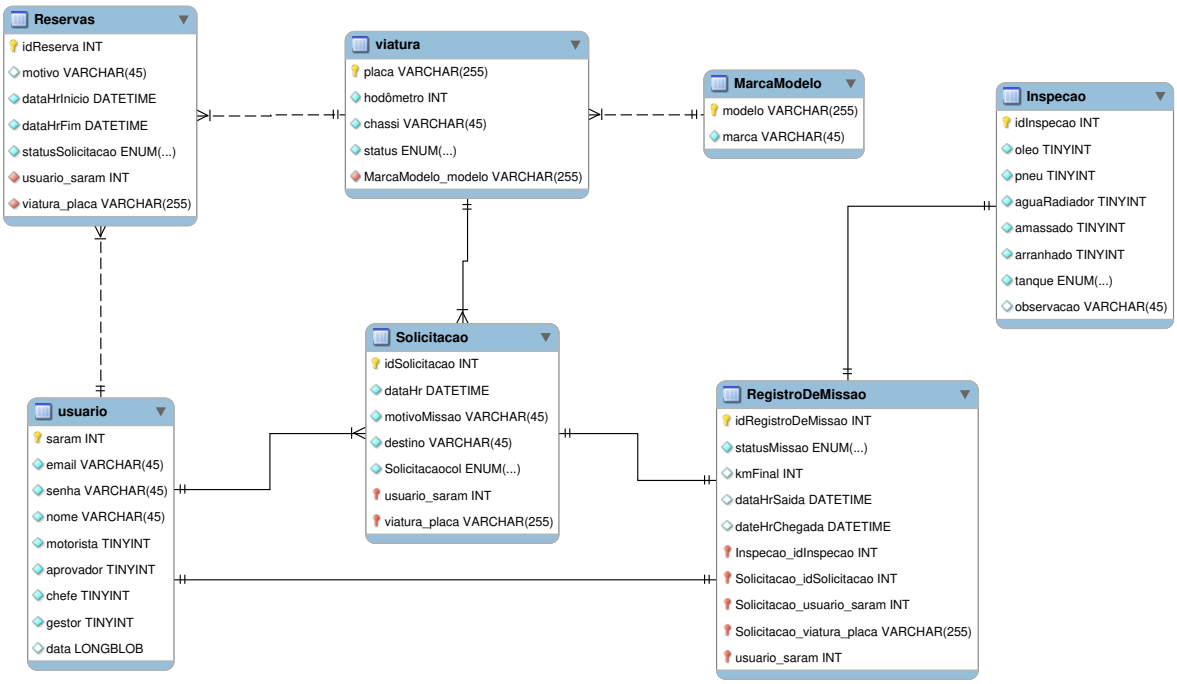
3.2.1 Backend do Gerenciador do Sistema

O núcleo de um sistema de controle de acesso reside na robustez de seu *backend*, uma infraestrutura central responsável por orquestrar operações críticas. No escopo desta pesquisa, dedicamos especial atenção ao desenvolvimento de um *backend*, caracterizado por sua eficiência e segurança. Este componente, meticulosamente concebido, desempenha um papel crucial na harmonização e gestão eficaz de dados relacionados ao controle de acesso. Ao longo deste trabalho, exploraremos em detalhes a arquitetura e as funcionalidades específicas desse *backend*, contextualizando sua relevância na criação de um sistema de controle de acesso robusto e eficiente.

Para o desenvolvimento do sistema, contaremos com a utilização das seguintes dependências e bibliotecas com suas respectivas funcionalidades:

- *Spring Boot Dev Tools*: Conjunto de funcionalidades para agilizar o desenvolvimento dos projetos, oferecendo recursos como reinicialização automática da aplicação durante o desenvolvimento, recarregamento de alterações no código-fonte e melhorias na experiência de desenvolvimento, tornando o processo mais eficiente e produtivo;
- *Lombok*: Biblioteca que visa a geração de códigos, já criando de forma oculta alguns métodos comuns e necessários como: *getters, setters, equals*;

Figura 7 – Diagrama de Entidade Relacionamento.



Fonte: Do Autor.

- *Spring Data* JPA: Cuida das implementações de nossos repositórios e da persistência de dados;
- *O Spring Boot Data Rest*: É uma extensão do *Spring Boot* que facilita a criação de *APIs RESTful* diretamente a partir de repositórios *Java Persistence API* (JPA). Ele elimina a necessidade de escrever manualmente os controladores para manipular operações *Create, Read, Update, Delete* (CRUD) em entidades persistentes;
- *MySQL Driver*: Permite fazer a conexão com o MySQL.

Para a utilização das respectivas dependências, foi utilizado o site do *spring boot* no qual foram feitas todas as configurações necessárias e a seleção das dependências desejadas. Após esta etapa, iniciou a montagem do diagrama de entidade relacionamento para auxiliar na posterior confecção do código. Conforme a Figura 7, podemos ver como será composto o diagrama.

Com base no diagrama de entidade relacionamento, o *backend* inicial foi desenvolvido em Java, composto por sete classes principais e quatro classes *enum*. É relevante destacar que, mediante a utilização de dependências, a criação manual de um banco de dados MySQL torna-se dispensável. Isso se deve ao fato de que, a partir das classes desenvolvidas e dos relacionamentos definidos no código Java, o banco de dados é gerado automaticamente. Nesse contexto, é necessário apenas criar um arquivo de configuração

denominado “application.properties”. Este arquivo requer a inclusão dos detalhes do servidor de banco de dados previamente criado no ambiente Docker.

Após a implementação do banco de dados, tornou-se imperativo desenvolver uma API para atender às requisições do *frontend*, bem como para encapsular toda a lógica subjacente dos sistemas. Este processo assegurou que todas as situações delineadas nos casos de uso fossem adequadamente verificadas pelo sistema, minimizando a ocorrência de erros. A extensão utilizada para alcançar esse objetivo foi o *Spring Boot Data Rest*. Para facilitar a criação de rotas para a API, foram organizadas diversas classes, cada uma cumprindo funções distintas e segregadas em pastas específicas:

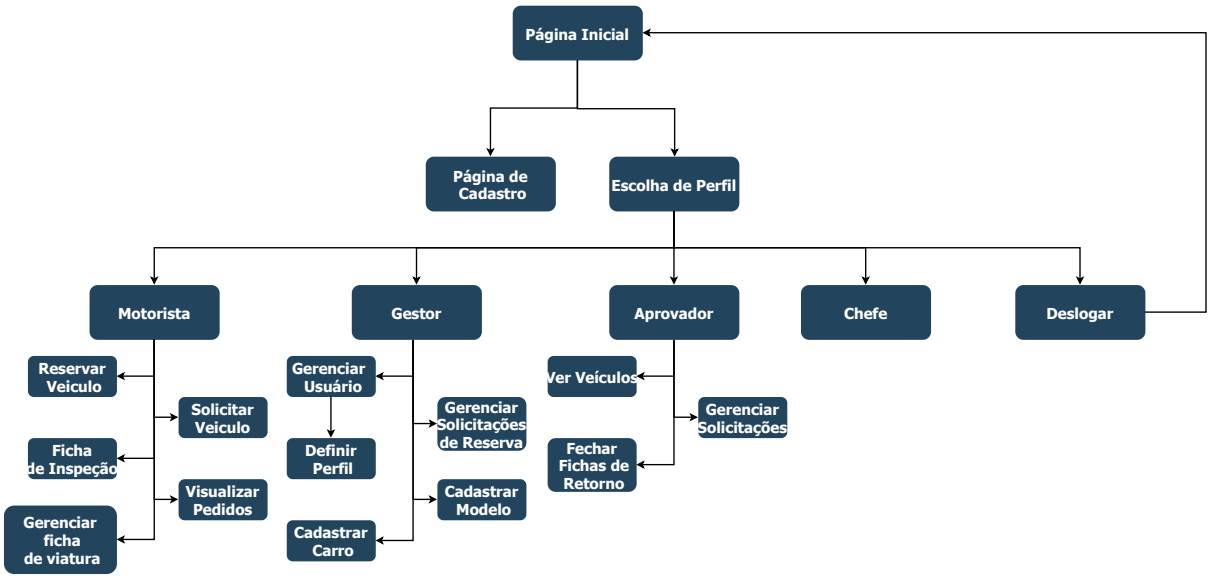
- *Repository*: Esta pasta abriga diversas interfaces responsáveis por inicializar a extensão *REST*, estabelecendo a ligação entre as entidades do sistema e os *endpoints* da API.
- *Controller*: Nesta pasta, encontram-se classes nas quais são definidas as rotas e implementada toda a lógica do sistema, garantindo a correta manipulação das requisições recebidas.
- *Requests*: Classes nesta pasta são responsáveis por definir a estrutura do *body* do JSON utilizado nas requisições, contribuindo para uma comunicação eficiente entre o *frontend* e o *backend*.
- DTO (Data Transfer Object): Classes nesta categoria criam objetos contendo informações específicas para serem retornadas ao *frontend*, promovendo uma separação clara entre dados manipulados internamente e dados enviados para a interface do usuário.

3.2.2 Frontend

Após desenvolver integralmente a estrutura lógica do sistema, é imperativo estabelecer uma interface de interação do usuário com o sistema. Para este propósito, empregamos o *frontend*, desenvolvido em *JavaScript* com a utilização da biblioteca *React*. Para estruturar o código, estilizar o *layout* e fazer a lógica de programação, foi utilizada a linguagem *TypeScript* com auxílio da biblioteca *mui*. Conhecido por ser uma aplicação *Single Page Application* (SPA), o *frontend* consiste em criar uma interface de usuário que possibilite a execução de ações de maneira interativa, sem a necessidade de acesso ao código-fonte ou compreensão profunda de operações em um banco de dados. O *frontend* foi concebido com o intuito de abranger todas as ações de requisição e gerenciamento identificadas no diagrama de Casos de Uso anteriormente apresentado.

A estrutura do código segue a abordagem em que cada página do programa possui um arquivo index.tsx e uma pasta components contendo outro arquivo .tsx. O arquivo

Figura 8 – *Site Map* da interface.



Fonte: Do Autor.

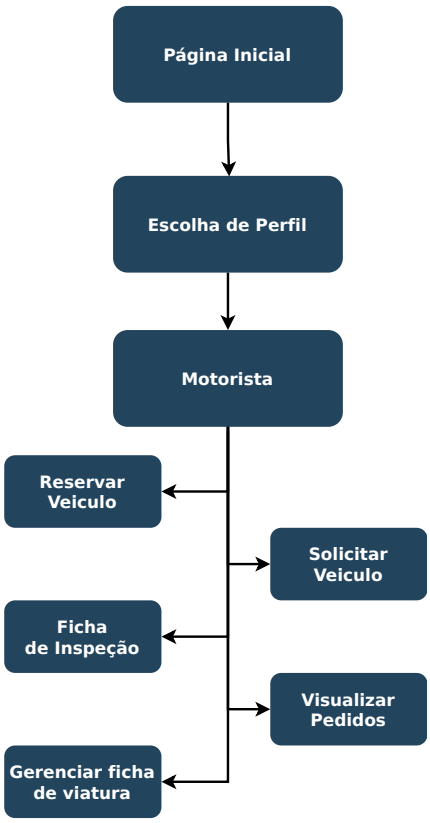
index.tsx tem como propósito modelar a interface gráfica e gerenciar a execução do código com base nas interações do usuário. Por exemplo, na tela inicial de login, o arquivo index.tsx redireciona para a pasta “components”, que contém o arquivo LoginForm.tsx. Este arquivo tem a função de coletar as informações inseridas pelo usuário na página, realizar uma requisição via API para o *backend* a fim de validar as informações fornecidas e receber um objeto de usuário com detalhes sobre o mesmo, caso o login e a senha estejam corretos. Em seguida, o código é redirecionado para a pasta “dashboard”, que representa a página de escolha de perfil. Esse ciclo se repete a cada interação do usuário. Para melhor entendimento, foi criado um *Site Map* na Figura 8.

A concepção da implementação de perfis de usuários tem como objetivo primordial a segmentação de responsabilidades, visando conceder a cada usuário autorizações específicas para a execução de tarefas determinadas. Essa abordagem proporciona uma interface de programa mais acessível e organizada. Para cada perfil de usuário, foi desenvolvido um fluxograma meticulosamente elaborado. Este instrumento destina-se a proporcionar uma compreensão clara e concisa da interação do usuário com a interface gráfica, delineando as ações necessárias e as disponíveis para cada perfil. Essa estratégia visa não apenas melhorar a usabilidade do programa, mas também contribuir para a sua organização eficiente.

A implementação do perfil do motorista inclui cinco opções distintas, conforme Figura 9, para facilitar e organizar suas atividades relacionadas a missões com viaturas. Estas ações são:

- Solicitar reserva: Esta opção direciona o motorista para uma nova tela na qual são

Figura 9 – *Site Map* Motorista.



Fonte: Do Autor.

solicitados dados como veículo desejado, motivo da reserva, data de início e data de término. Estas informações são posteriormente revisadas pelo gestor, que decide se aceita ou recusa a reserva.

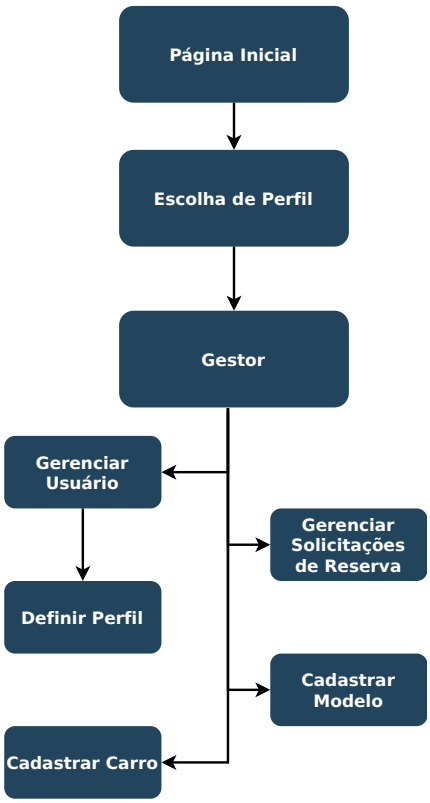
- Solicitar Veículos: Essa funcionalidade permite que o motorista visualize as viaturas disponíveis para uso, escolha uma delas e solicite aprovação ao aprovador. Informações como motivo e destino são fornecidas durante o processo de solicitação.

- Ficha de Inspeção: Para utilizar esta opção, o motorista deve ter previamente submetido uma solicitação de viatura que foi aprovada. Essa opção oferece campos para o motorista realizar a inspeção da viatura, abrangendo aspectos como nível de combustível, condição dos pneus, limpeza, estado da lataria e outras informações mecânicas relevantes.

- Ver solicitações: Criada com o intuito de proporcionar ao motorista a capacidade de acompanhar as solicitações que foram abertas por ele, seja para reserva de viatura ou solicitação de veículos.

- Gerenciar Ficha de Viatura: Para comprovar a autorização para circular com um veículo específico, o motorista deve portar a ficha de viatura. Esta ficha é gerada automaticamente pelo sistema assim que o motorista é identificado na saída.

Figura 10 – *Site Map* Gestor.



Fonte: Do Autor.

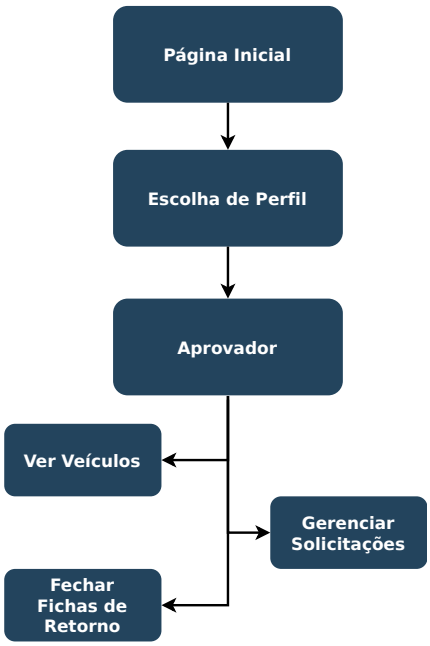
Contendo informações detalhadas sobre a missão, ela também oferece a opção de fechamento, sendo destacado que o fechamento só é possível após identificação no retorno.

O perfil do gestor desempenha um papel crucial na administração de reservas e novos usuários, incluindo a atribuição adequada de perfis e a inserção de novos veículos conforme necessário. O *Site Map* correspondente está apresentado na Figura 10. Quatro ações foram desenvolvidas para atender a esses propósitos, e elas serão elucidados abaixo:

- Gerenciar Usuário: Esta página lista todos os usuários que realizaram cadastro. O gestor pode visualizar todos os dados relativos a essas solicitações e determinar qual perfil atribuir a cada usuário. Por exemplo, se um novo motorista se cadastrar, o gestor selecionará o perfil dele e clicaŕa em abrir formulário, será aberto uma nova página na qual o gestor irá determinar quais perfis aquele usuário terá acesso.

- Gerenciar Solicitações de Reserva: Com o objetivo de visualizar todas as solicitações abertas por motoristas, esta página apresenta todas as informações relevantes para que o gestor possa aprovar ou recusar cada uma delas. À medida que são aprovadas ou recusadas, essas solicitações não ficam mais disponíveis na página. É importante ressaltar que essas reservas são fundamentais, pois durante sua vigência, alteram

Figura 11 – *Site Map* Aprovador.



Fonte: Do Autor.

o status de visualização para os motoristas, permanecendo visíveis apenas para o motorista que detém aquela reserva.

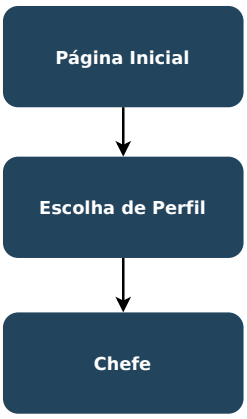
- Cadastrar Carro: Em casos em que há necessidade de inclusão de um veículo novo, se ele for de um modelo já existente no banco de dados, basta inserir as informações pertinentes a ele. Caso contrário, será necessário realizar o cadastro do modelo primeiro, o que também estará disponível.
- Cadastrar Modelo: Esta opção foi criada com o objetivo de cadastrar um modelo novo de carro. Antes de cadastrar um veículo novo, é necessário incluir as informações básicas sobre o modelo, garantindo um registro completo e preciso.

Essas funcionalidades foram desenvolvidas para permitir que o gestor exerça controle e supervisão eficazes sobre as reservas de viaturas, atribuição de perfis de usuários e gerenciamento do banco de dados de veículos.

O perfil do Aprovador desempenha um papel fundamental no sistema, sendo responsável pelo controle de saída de viaturas. Suas funcionalidades, disponíveis na Figura 11, são essenciais para garantir que nenhuma viatura saia do estabelecimento sem a devida aprovação. Algumas das funcionalidades implementadas para esse perfil incluem:

- Ver veículos: Nesta funcionalidade, o aprovador pode verificar o status de todas as viaturas cadastradas no estabelecimento. Se um veículo estiver na sede e sem uso, seu status será apresentado como disponível. Caso contrário, será marcado como em missão.

Figura 12 – *Site Map* Chefe.



Fonte: Do Autor.

- Gerenciar solicitações: Neste campo, todas as solicitações em aberto são apresentadas ao aprovador. Ele tem a capacidade de aprovar ou recusar cada solicitação. Vale ressaltar que esta etapa é um pré-requisito para a ficha de inspeção feita pelo motorista. A ficha de inspeção só fica disponível caso o aprovador aprove a solicitação.
- Fechar Fichas de Retorno: Essa funcionalidade visa confirmar a entrega das chaves da viatura e a verificação de que todas as informações lançadas na ficha de viatura são verídicas. É uma etapa crucial para assegurar a precisão das informações registradas durante a missão.

Esse conjunto de funcionalidades foi projetado para garantir o controle efetivo e a verificação rigorosa de todas as saídas de viaturas, proporcionando segurança e confiabilidade ao sistema.

O perfil de Chefe foi concebido com o propósito específico de gerar relatórios das saídas de veículos aprovadas. Este perfil contém todas as informações relevantes para cada viagem, incluindo data e horário de início e fim, autorização, motorista designado, destino, motivo da viagem, condição da viatura, entre outras. A utilização deste perfil é bastante simplificada, exigindo apenas a informação do mês e ano desejados para visualização das informações, conforme apresentado na Figura 12

Esse processo simplificado torna a geração de relatórios acessível e eficiente, proporcionando ao chefe uma visão abrangente e detalhada de todas as saídas de veículos autorizadas durante o período desejado. O *site map* correspondente, conforme apresentado na Figura 12, visa auxiliar na compreensão dessa funcionalidade.

Agora que obtivemos uma visão geral do desenvolvimento do *frontend*, no próximo tópico será apresentada a API. Esta foi criada com o objetivo de estabelecer a comunicação entre o *frontend* e o *backend*.

3.2.3 API RESTful

A API RESTful desenvolvida desempenha um papel essencial na comunicação entre o *frontend* e o *backend* de nossa aplicação. Seguindo os princípios da arquitetura REST, essa interface proporciona um meio flexível e escalável para a transferência de dados. Projetada para ser intuitiva e de fácil integração, a API RESTful permite que o *frontend* envie solicitações para o *backend* de maneira padronizada, utilizando os métodos HTTP convencionais. Essa abordagem simplifica a interação entre as duas partes, facilitando o desenvolvimento, manutenção e expansão da aplicação. Ao utilizar *endpoints* claramente definidos, a API possibilita que o *frontend* acesse e manipule recursos no *backend* de forma segura e eficiente. A padronização RESTful também promove uma arquitetura uniforme, permitindo uma escalabilidade mais simples à medida que a aplicação cresce.

A execução de determinadas funcionalidades no *frontend* requer a realização de requisições, como, por exemplo, a verificação de solicitações criadas. A concepção de um *backend* tem como principal propósito a eliminação da necessidade de armazenar dados no *frontend*, resultando em uma redução da carga de processamento neste último. Retornando ao contexto das requisições, para obter informações sobre as solicitações existentes, o *frontend* simplesmente envia uma requisição específica ao *backend*.

A Tabela 3, no Apêndice A, abaixo apresenta um compêndio das rotas GET implementadas no *backend*, cada uma delas associada a suas respectivas funções.

Além das requisições do tipo GET, foram implementadas o POST e o PUT, ambas realizando alguma alteração no banco de dados, sendo o POST para criação e o PUT para modificação. Na Tabela 4, no Apêndice A, é possível verificar todas as rotas criadas para este tipo de requisição.

Essa abordagem facilita a compreensão e a integração eficiente entre as camadas *frontend* e *backend*, seguindo os princípios de comunicação estabelecidos.

3.2.4 Backend do Subsistema de Identificação

O *backend* de Identificação foi concebido com o propósito de armazenar dados dos motoristas e empregar técnicas de *machine learning* para o reconhecimento facial e de placas de veículos. Este *backend* incorpora funções cruciais, incluindo treinamento, atualização de dados, e, por fim, a capacidade de identificar, através de reconhecimento facial, tanto o motorista quanto a placa do veículo por meio do reconhecimento de formas e caracteres. Composto por seis classes distintas, destacam-se as seguintes:

- Menu: Uma classe desenvolvida para a inicialização do programa, contendo um switch com opções para realizar treinamento, atualizar dados e efetuar identificação.

- train: Classe que realiza o treinamento da *machinelearnig* no reconhecimento facial, ela obtém as imagens realiza o *encoding* facial e adiciona a uma lista

- dados: Esta classe assume a responsabilidade de comunicar-se com o backend principal, solicitando, por meio da rota /usuarioMotorista, os IDs dos motoristas. Posteriormente, realiza uma requisição pela rota /download/saram para manter atualizado o *backend* de identificação, armazenando os dados de forma que o nome de cada imagem do motorista seja seu ID.

- rccognize: Classe projetada para o reconhecimento facial, contendo métodos para carregar a lista criada pela classe Train, detectar faces conhecidas (ou seja, aquelas salvas na lista) e comparar se é a mesma face.

- rccognizeCar: Implementada com o propósito de detectar a placa do veículo por meio de contornos e detecção de caracteres.

- identify: Classe criada para realizar a validação da saída. Nela, métodos foram implementados para estabelecer comunicação com ambas as câmeras (para o rosto e placa), capturar a imagem, utilizar a função de detecção de face implementada na classe Rccognize e, em seguida, chamar a função capturaplaca() da classe RccognizeCar para coletar a placa do veículo. Posteriormente, realiza uma requisição PUT na rota /validaSaida, com o intuito de informar ao *backend* principal o ID do motorista e a placa do veículo para consulta de autorização. Em caso positivo, recebe um status 200 e, em seguida, emite o comando para a abertura da cancela.

  Em síntese, o *backend* de Identificação apresenta-se como uma solução abrangente para o reconhecimento facial e validação de saída de motoristas, integrando de forma eficaz as diferentes funcionalidades propostas.

3.2.5 Dispositivos Conectados ao Sistema

Na conformação do sistema de controle em consideração, estão incorporados quatro dispositivos fundamentais, cujas funcionalidades coadjuvantes se revelam cruciais. Dois dispositivos de câmera *Internet Protocol* (IP), responsáveis pela captura e transmissão de dados visuais, desempenham papel essencial na supervisão do ambiente em questão. Adicionalmente, um componente de *hardware* é dedicado ao controle da bomba de combustível, enquanto um segundo *hardware* é alocado para a gestão da cancela. Este arranjo estratégico propicia uma integração sinérgica dos dispositivos, alinhando-se ao propósito de otimizar as operações inerentes ao sistema de controle em foco.

3.2.5.1 Câmera IP

As câmeras IP, ou câmeras de rede, representam uma categoria avançada de dispositivos de vigilância que incorporam tecnologias de rede para transmissão e gerenciamento de imagens. Estas câmeras operam mediante a captura de imagens e vídeos, que são posteriormente digitalizados e transmitidos através de uma rede IP (Internet Protocol). O processo de funcionamento de uma câmera IP inicia-se com a captação da cena por meio de um sensor de imagem, geralmente um sensor CMOS ou CCD. Essas tecnologias convertem a luz incidente em sinais elétricos, representativos da imagem. Os sinais elétricos são então processados internamente, passando por etapas como amplificação e conversão analógico-digital para obtenção de dados digitais. A transformação em dados digitais facilita a transmissão eficiente dessas informações por meio de uma rede IP. A câmera IP, portanto, possui capacidades de conectividade, permitindo a transmissão dos dados através de redes locais (LAN), redes corporativas ou até mesmo pela internet.

No contexto do sistema implementado, a câmera desempenha um papel crucial, incumbida da tarefa de captura em tempo real de imagens. Sua funcionalidade primordial reside na assistência ao processo de identificação facial e leitura de placas veiculares. A captura em tempo real, realizada por meio desta câmera, desempenha um papel estratégico na obtenção de informações visuais instantâneas, as quais são posteriormente submetidas a procedimentos analíticos voltados à identificação facial e ao reconhecimento de placas.

3.2.5.2 Hardware da bomba de combustível

Para a instauração deste dispositivo, fez-se imprescindível a incorporação de uma placa ESP32, um módulo SIM808, um módulo de relé e dois reguladores de tensão, um operando a 5V e o outro a 3,3V. A disposição do circuito se delineia da seguinte maneira: a bateria do veículo, ostentando uma tensão de 12V, alimentará ambos os reguladores de tensão. O regulador de 5V será destinado à alimentação da placa ESP32, enquanto o de 3,3V servirá ao módulo de relé (VCC). O pino GPIO 4 da ESP32 será empregado para conectar ao pino IN (entrada de controle) de ativação do relé, o pino NO (normalmente aberto), por sua vez, estará vinculado à bomba de combustível e o pino COM que ficará conectado a bateria sendo alimentado com 12V. Importante ressaltar que o relé receberá energia diretamente da bateria de 12V, visto que é esta a tensão destinada ao funcionamento da bomba. O relé desempenhará unicamente a função de chaveamento no circuito. E um SIM808 onde os pinos, TX e RX se conectam aos pinos RX e TX do ESP32 respectivamente.

A escolha destes componentes para compor o sistema de gerenciamento da bomba de combustível se dá pelo fato da necessidade de comunicação do sistema com a bomba de combustível, para isto, foi utilizado a ESP32 que ficará responsável por realizar o comando do chaveamento que será feito pelo módulo do relé, e o SIM808 tem como função de

integrar funcionalidades de comunicação por dados móveis e um módulo GPS, que para este projeto inicial não será utilizado mas visando um upgrade posterior, o mesmo poderia ser utilizado para conversar com o sistema de modo a se ter um rastreamento em tempo real do veículo.

O sistema, em sua essência, opera da seguinte maneira: ao condutor concluir a inspeção do veículo, uma mensagem *Message Queuing Telemetry Transport* (MQTT) é enviada para a placa ESP32, solicitando a ativação do pino 4, levando-o a um estado HIGH. Subsequentemente, quando o condutor retorna e finaliza a ficha de inspeção, o sistema envia outra mensagem MQTT para que o pino 4 retorne a LOW, efetivamente interrompendo a alimentação da bomba de combustível. A utilização do protocolo MQTT é crucial para garantir a segurança do processo, pois através do mesmo foi utilizado os recursos de autenticação garantindo apenas que dispositivos autorizados possam estar comandando o EPS32, além da criptografia e da confiabilidade na entrega, tendo em vista que o mesmo possui mecanismo de confirmação e garantia de entrega da mensagem.

3.2.5.3 Hardware da cancela

Comandado pelo *backend* por meio da comunicação via protocolo *Message Queuing Telemetry Transport* (MQTT) sobre TCP/IP, o *hardware* da cancela foi desenvolvido com exclusividade para emular o processo de abertura de uma cancela. O dispositivo foi concebido utilizando um ESP32, no qual foi implementado um sistema composto por LEDs destinado a simular a operação de abertura da cancela. O propósito primordial desse arranjo residia na verificação da permissão ou restrição da saída. Através da visualização do comportamento do LED, era possível determinar se a saída estava autorizada.

Para a comunicação efetiva com o *backend*, optamos por utilizar o protocolo MQTT, garantindo uma transmissão eficiente e segura de comandos entre o dispositivo e o servidor. O componente de conectividade celular escolhido foi o módulo SIM800L, proporcionando uma comunicação estável e adaptada ao ambiente fixo da cancela. A escolha do SIM800L, em detrimento do SIM808, foi motivada pela natureza fixa da instalação, eliminando a necessidade de funcionalidades de localização GPS e tornando o dispositivo mais eficiente para a aplicação específica.

Adicionalmente, considerando uma futura integração com uma cancela automática, o *hardware* projetado permite uma transição simplificada. Bastaria substituir a comunicação com o ESP32 pela interação direta com o hardware específico da cancela a ser instalada, mantendo a lógica de controle previamente estabelecida. Esse design modular facilita a evolução do sistema para uma cancela automática mais robusta, alinhada aos requisitos futuros, sem a necessidade de alterações significativas na infraestrutura já existente.

4 RESULTADOS

O objetivo geral deste trabalho é desenvolver um sistema computacional que permita reservas e controle da entrada e saída de veículos, visando a resolução de problemas relacionados a gestão de frotas. Com o intuito de demonstrar a eficácia na consecução desse objetivo, esta seção explora a experiência do usuário durante a utilização do sistema, enfatizando os processos automatizados que simplificam a operação. Ao longo deste capítulo, também são fornecidas visualizações das interfaces acessadas pelo usuário ao interagir com a ferramenta desenvolvida.

4.1 Interações do Usuário com o Sistema

A estratégia adotada para o desenvolvimento do sistema baseou-se no conceito de Produto Mínimo Viável (*Minimum Viable Product* (MVP)), o qual propõe o lançamento no mercado da versão mais simplificada e funcional do produto. Essa abordagem prioriza a inclusão apenas dos recursos essenciais para atender às necessidades básicas dos usuários, com o propósito de obter *feedback* rápido, validar hipóteses e reduzir o tempo e os recursos alocados no desenvolvimento de funcionalidades que possam não ser consideradas indispensáveis.

Para a concepção do *frontend*, inicialmente, foram desenvolvidas as telas de login e cadastro apresentadas nas Figura 13 e Figura 14. Essas telas têm como propósito principal auxiliar o sistema na identificação da pessoa que está interagindo com ele, ao mesmo tempo em que solicitam informações cruciais do *backend*. Essa abordagem visa restringir o acesso do usuário e permitir que novos usuários solicitem acesso ao sistema, inserindo seus dados. Após essa etapa, aguarda-se a aprovação do gestor, que definirá o perfil a ser atribuído ao usuário.

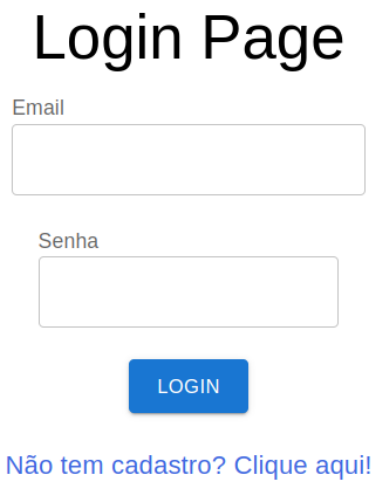
Com o propósito de elucidar de maneira mais aprofundada o funcionamento do sistema, procederemos à sua descrição apresentando as etapas necessárias para que um veículo possa sair e qual procedimento deve ser realizado ao retornar.

4.1.1 Etapas para saída do veículo

Para poder sair com a viatura, é necessário cumprir as seguintes etapas:

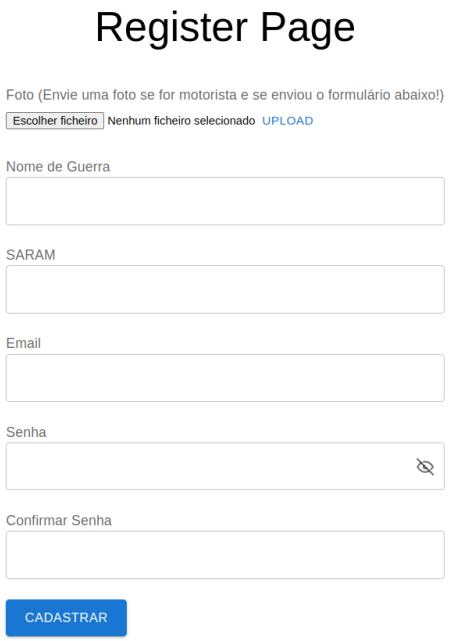
- Solicitação de veículo pelo motorista;
- Aprovação da solicitação pelo aprovador;

Figura 13 – Tela de Login.



Fonte: Do Autor.

Figura 14 – Tela de Registro.



Fonte: Do Autor.

Figura 15 – Telas para fazer uma solicitação de veículo.

(a) Tela Inicial do Motorista.

(b) Tela de Solicitação.

- Preenchimento da ficha de inspeção pelo motorista;
- Identificação na cancela.

Para acessar o perfil de motorista, após fazer o *login*, o usuário precisa selecionar o perfil motorista. A partir desta ação, será apresentado uma tela inicial a ele, conforme Figura 15a. A seguir será demostrado o passo a passo para abrir uma solicitação, realizar uma inspeção e finalizar uma ficha e posteriormente, a solicitação de uma reserva.

Para a criação de uma solicitação de viatura, o motorista inicia realizando o login,

Figura 16 – Telas para gerenciar uma solicitação de veículo.

(a) Tela Inicial do aprovador. (b) Tela de Solicitação.

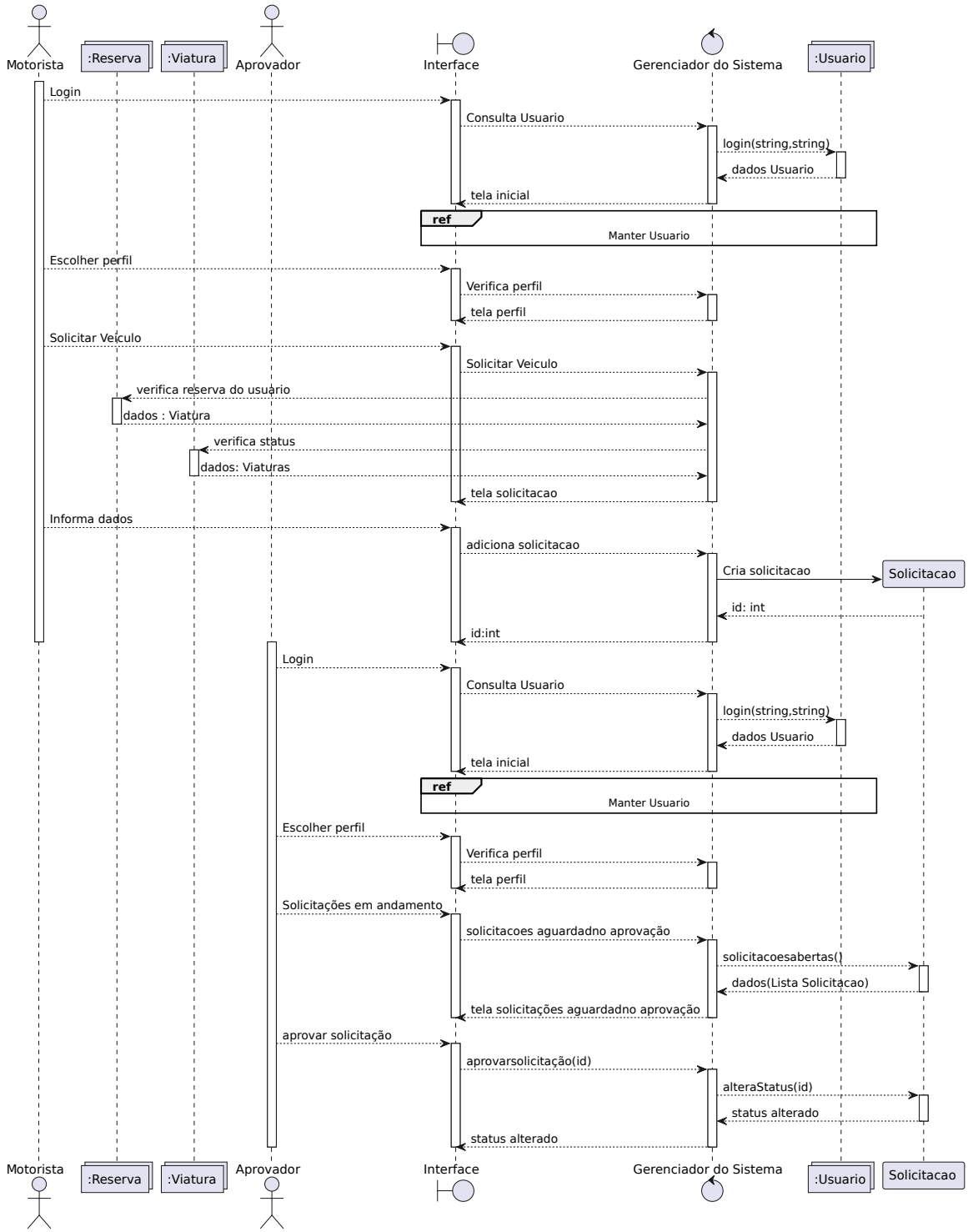
acessa seu perfil específico de motorista e, em seguida, entra no campo de solicitação de viatura, onde fornece as informações necessárias e envia, bastando agora aguardar a aprovação, conforme a Figura 15b. Todo esse processo de criar a solicitação, envolve diversas atividades que são desempenhadas pelo *frontend* e o *backend*. Com o intuito de proporcionar um entendimento mais claro sobre o funcionamento interno do programa, foi elaborado um diagrama sequencial desta etapa, apresentado na Figura 17.

Ao analisar o diagrama, torna-se evidente a complexidade dessa requisição. A interface interage com o sistema solicitando informações sobre as reservas desse usuário e a situação atual das viaturas. Em caso de o usuário possuir uma reserva aprovada em vigor exatamente naquele momento, a viatura que, para os demais, estaria indisponível, apareceria como disponível para esse usuário. Além disso, o usuário pode visualizar a disponibilidade de outras viaturas. Essas informações compõem o botão *select*, que estará disponível na página, permitindo ao usuário escolher a viatura desejada e fornecer dados relevantes, como motivo e destino. E assim que ele submete os dados, o *backend* cria uma nova linha na tabela de solicitação e retorno do id dessa solicitação para o *front*.

Na etapa subsequente, ocorreu a implementação do gerenciamento das solicitações. Nesse contexto, o aprovador realiza o login e acessa o campo de solicitações, onde são exibidas todas as solicitações pendentes aguardando uma avaliação (Figura 16). Nessa interface, o aprovador tem acesso a todas as informações necessárias para tomar uma decisão sobre a solicitação, podendo aceitá-la ou rejeitá-la ao selecionar a opção desejada.

Todo esse procedimento encontra-se representado na Figura 17, proporcionando uma compreensão detalhada do funcionamento interno do programa. Conforme observado, o processo de login é uniforme para todos os perfis. Em seguida, para que o aprovador possa visualizar as solicitações em aberto, é imprescindível que o *front-end* realize uma

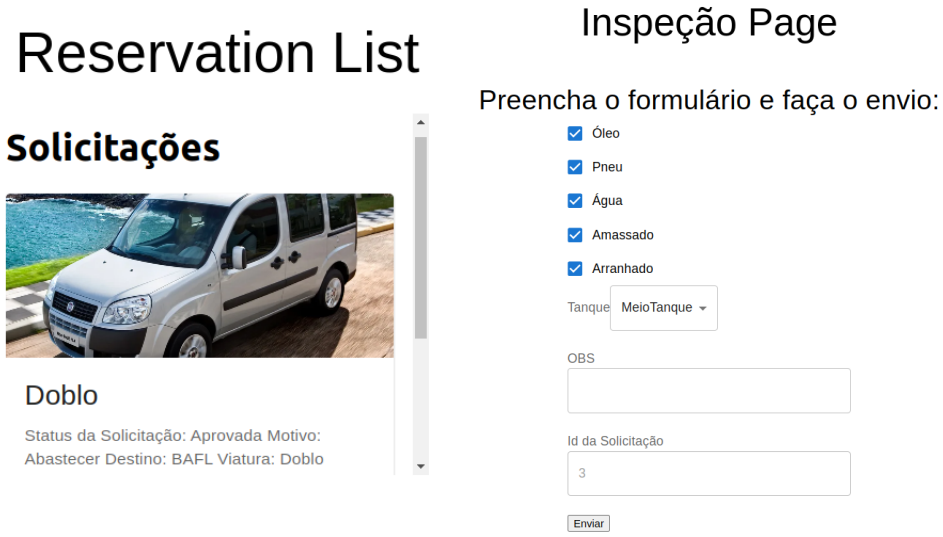
Figura 17 – Diagrama de Sequencia - Visualização e aprovação de solicitações.



Fonte: Do Autor.

requisição para o *back-end*, de modo a obter, de maneira pré-filtrada, todas as solicitações pendentes, ou seja, aquelas que ainda não foram aprovadas ou recusadas, possibilitando sua exibição na tela. E em seguida o aprovador realiza a aprovação da solicitação, fazendo com que novamente haja uma nova requisição de forma que está sendo solicitado que seja alterado o status da solicitação no banco de dados.

Figura 18 – Telas para gerenciar uma solicitação de veículo.



(a) Tela acompanhamento de pedidos. (b) Tela de ficha de inspeção.

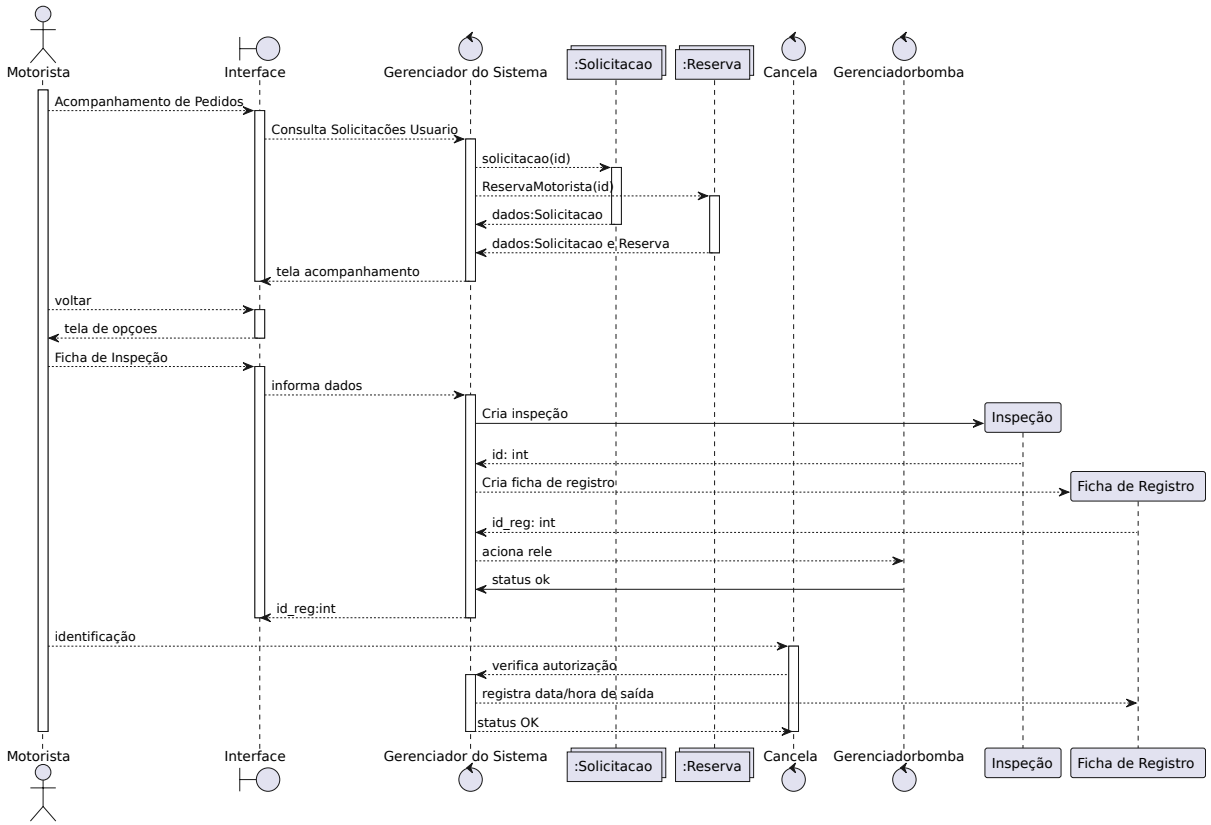
Fonte: Do Autor.

Para acompanhar o pedido, o motorista deve acessar o campo “Visualizar Solicitações”, onde terá acesso às informações conforme ilustrado na Figura 18a Após a aprovação da solicitação, o motorista deverá realizar a inspeção. Para isso, basta acessar o campo “Ficha de Inspeção” disponível em sua tela, conforme demonstrado na Figura 15a, e preencher os campos com as informações solicitadas, conforme apresentado na Figura 18b. Ao sair da unidade, o motorista deve ser identificado na cancela para que o sistema verifique se há autorização para a saída. Se todas as condições estiverem autorizadas, a cancela se abrirá.

Agora que já foi visto as etapas a serem realizadas, será analisado o que ocorre internamente no sistema para melhor entendimento foi desenvolvido o diagrama de sequência destas etapas na Figura 19:

- Visualização das solicitações:
  - O motorista acessa a página de acompanhamento para visualizar seus pedidos.
  - Para tornar essas informações visíveis, o *frontend* envia duas requisições para o *backend* sendo uma solicitando as reservas vigentes do motorista e outra solicitando as solicitações abertas do motorista.
  - O *backend* verifica essas informações junto ao banco de dados e as retorna.
  - Assim que o *frontend* recebe as informações, ele as exibe na tela.
- Criação de Inspeção:

Figura 19 – Diagrama de Sequência para acompanhamento, inspeção e identificação na saída do veículo.



Fonte: Do Autor.

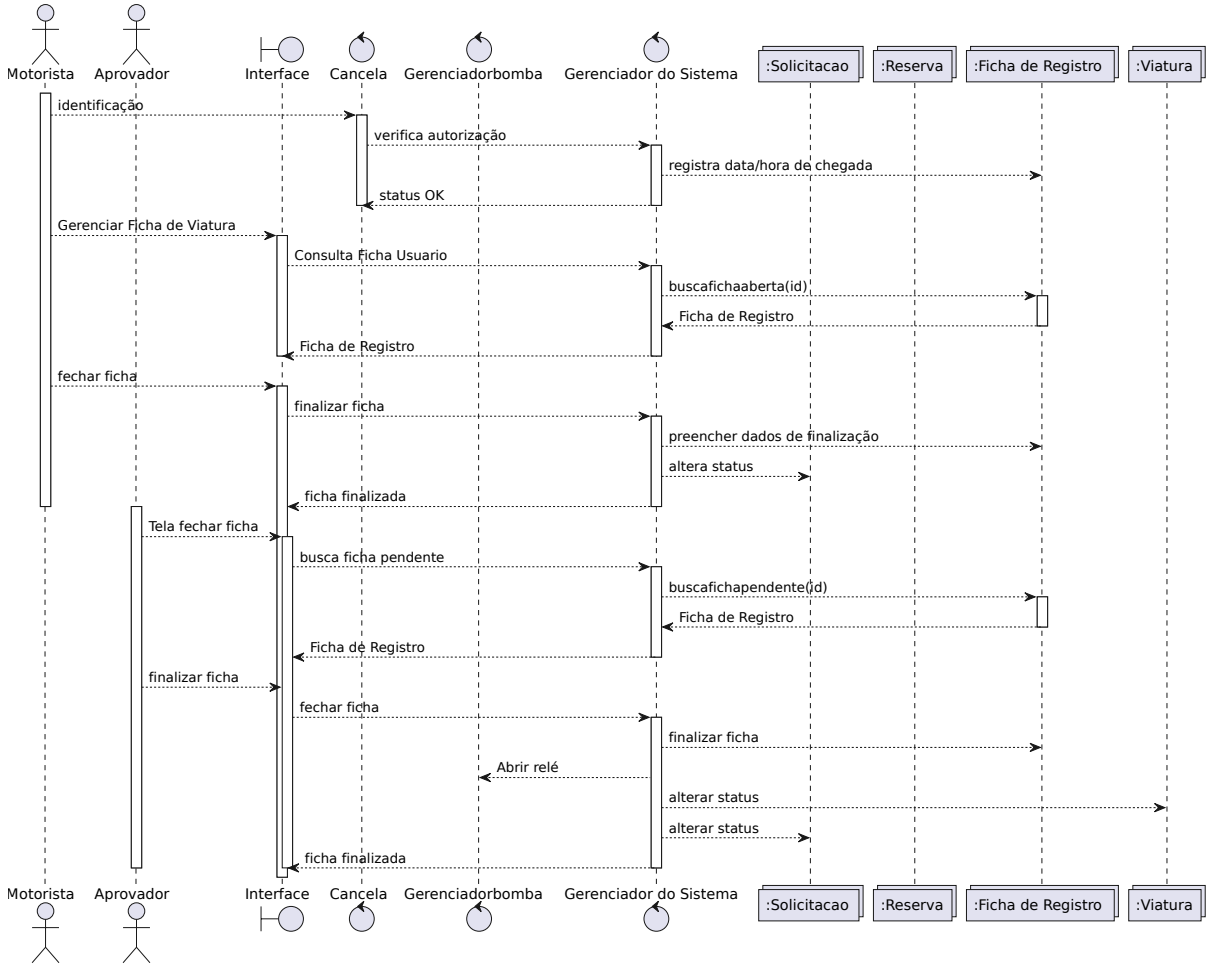
  - Após verificar que sua solicitação foi aprovada, o motorista acessa o campo para criação de inspeção.
  - Ele fornece os dados necessários, que são enviados para o *backend*.
  - O gerenciador do sistema adiciona uma linha à tabela de inspeção e outra à tabela de registro, contendo todas as informações referentes a essa solicitação. Em seguida, envia um comando para o gerenciador da bomba, solicitando o fechamento do relé, permitindo que o veículo ligue.

- Identificação

  - A cancela identifica e envia uma requisição de validação ao gerenciador.
  - O gerenciador registra linha de ficha de registro criada anteriormente a hora e data de saída.
  - E valida a saída do motorista.

O condutor terá acesso à ficha da viatura em sua conta de usuário. Para visualizá-la, basta acessar o campo destinado à gestão da ficha de viatura. Neste local, ele não apenas poderá consultar os dados, mas também realizar a solicitação de encerramento. Contudo,

Figura 20 – Diagrama de Sequência para acompanhamento, inspeção e identificação no retorno do veículo.



Fonte: Do Autor.

esta operação somente será efetuada após a confirmação de que o condutor retornou. Quando o condutor retorna e se identifica, a cancela envia uma requisição ao gerenciador do sistema para verificar a autorização de retorno. Se autorizado, o gerenciador registra imediatamente a hora e a data de retorno, permitindo, a partir desse momento, que o condutor finalize a ficha. Ao solicitar o encerramento, o *frontend* envia ao *backend* as informações pertinentes para que este verifique a possibilidade de conclusão. Em caso afirmativo, o *backend* preenche automaticamente os campos na ficha de registro e altera o status da solicitação.

Para concluir o processo, é necessário a intervenção do aprovador, incumbido de encerrar a ficha. Basta acessar o campo correspondente, verificar a conformidade dos dados fornecidos e, em seguida, concluir a operação. Essa etapa de finalização consiste na requisição da *interface* para enviar ao *backend*o ID do aprovador, a fim de registrar na ficha de registro quem recebeu o veículo no retorno. Além disso, atualiza os status da solicitação e da ficha de registro para "finalizada"e o da viatura para "disponível". Uma

requisição é também enviada ao gerenciador da bomba para abrir o circuito da bomba de combustível, impedindo assim o uso não autorizado, além de liberar no sistema o condutor para que ele possa fazer novas solicitações de veículo, se necessário. Todo esse processo está representado na Figura 20.

5 CONCLUSÃO

O propósito central deste projeto consistiu no desenvolvimento de um sistema, fundamentado em um sistema computacional, para solucionar desafios associados à gestão de frotas. A proposta foi inspirada na experiência da autora na gestão das viaturas do Destacamento de Controle do Espaço Aéreo de Florianópolis (DTCEA-FL), com uma abordagem de implementação economicamente viável. O escopo abrangeu a criação de um banco de dados, a elaboração de uma interface gráfica para o sistema de gerenciamento e a construção de dois componentes de *backend*: um destinado a cumprir regras predefinidas, como a proibição de múltiplos pedidos em aberto para um mesmo veículo, e outro voltado para a identificação do condutor e da viatura.

Para efetuar a identificação, utilizou-se um sistema de câmeras IP para capturar imagens, as quais foram processadas por meio de reconhecimento facial baseado em aprendizado de máquina. Além disso, foram integrados dois dispositivos de hardware: um para abrir o portão após a autorização do sistema principal e outro para interagir com esse sistema, controlando o relé no circuito da bomba de combustível.

A comunicação eficaz entre esses elementos envolveu a implementação de interfaces, como a API *RESTful*, e a adoção dos protocolos de comunicação. A API serviu como uma interface de programação, enquanto os protocolos MQTT, TCP/IP e HTTPS foram os meios de comunicação utilizados.

Este projeto atendeu aos requisitos estabelecidos, proporcionando a criação de um sistema que automatiza o controle de entrada e saída de veículos, contribuindo não apenas para a segurança do estabelecimento, mas também para o aprimoramento do gerenciamento e controle das viaturas das empresas. O uso do reconhecimento facial, juntamente ao reconhecimento de placas foram indispensáveis pois reduzem o erro humano na identificação correta bem como o sistema de controle da bomba de combustível que impede o uso não autorizado do veículo, além do sistema de gerenciamento que centraliza todas as informações de forma automática facilitando a geração de dados para possíveis estudos de demandas recorrente por exemplo.

5.1 Trabalhos futuros

Dada a restrição temporal inerente ao projeto final de curso, algumas funcionalidades cruciais podem não ter sido implementadas. No entanto, para futuros trabalhos, sugerem-se diversas áreas de expansão e aprimoramento do sistema, abrindo oportunidades para um desenvolvimento mais robusto e abrangente. Algumas sugestões incluem:

- Utilização de *sing-on* (SSO) na criação de cadastro. Visando a praticidade e o aproveitamento dos dados do usuário na empresa.
- Estabelecer um processo de avaliação contínua do sistema, incorporando *feedbacks* regulares dos usuários para direcionar melhorias incrementais principalmente visando a interface gráfica.
- Ampliação do sistema de modo que permita que o gestor possa rastrear o veículo por GPS e também a implementação de uma função de bloqueio em caso de roubo/furto.
- Implementação de um sistema semelhante ao OBD II para coletar em tempo real o hodômetro do veículo. Essa abordagem permitirá o acompanhamento do rendimento do veículo, identificação de possíveis falhas nos sensores e aprimoramento geral do monitoramento do estado do veículo. Juntamente com a integração de um controle de manutenção no sistema, de modo que seja possível ver o histórico de manutenção do veículo.
- Criar aplicativos móveis que permitam aos usuários realizar tarefas relacionadas à gestão de veículos a partir de dispositivos móveis, proporcionando maior flexibilidade e mobilidade. Tendo em vista que atualmente só é possível utilizar no celular pelo *browser*.

Essas sugestões representam apenas algumas áreas que podem ser exploradas em trabalhos futuros, visando a evolução constante do sistema e sua capacidade de atender às demandas em constante mudança dos potenciais clientes.

REFERÊNCIAS

Amazon Web Services. *O que é uma API RESTful?* 2023. Internet. Acessado em: [14-Jun-2023]. Disponível em: <https://aws.amazon.com/pt/what-is/restful-api/>. 26

BERNARDI, E. et al. Modelo sistêmico e classificação de falhas associadas ao sistema de reconhecimento de placas para fiscalização automática de veículos. In: *Anais do XXIX Congresso de Pesquisa e Ensino em Transportes*. [S.l.: s.n.], 2015. p. 1510–1521. 21

BERTAGLIA, P. R. *Logística e gerenciamento da cadeia de abastecimento*. [S.l.]: Saraiva Educação SA, 2017. 17

CAMPOS, F. C. d.; BELHOT, R. V. Gestão de manutenção de frotas de veículos: uma revisão. *Gestão & Produção*, SciELO Brasil, v. 1, p. 171–188, 1994. 16

CLEMENTE, Q. K. *Gestão de frota de veículos*. 2013. 16

COTTI, L. Formulários para administração racional da frota. *Síntese, São Paulo*, 1989. 16

DIGITAL, C. O. *Reconhecimento facial policial da Champion's League errou 92% das vezes — olhardigital.com.br*. 2018. Internet. [Accessed 12-Jun-2023]. Disponível em: <https://olhardigital.com.br/2018/05/08/seguranca/reconhecimento-facial-policial-confundiu-2300-pessoas-com-potenciais-criminosos/>. 23

ELMASRI, R.; NAVATHE, S. Sistemas de banco de dados. 6a edição. *São Paulo: Person. 837p*, p. 24, 2011. 26

GLUHAK, A. et al. A survey on facilities for experimental internet of things research. *IEEE Communications Magazine*, IEEE, v. 49, n. 11, p. 58–67, 2011. 19

HEUSER, C. A. *Projeto de banco de dados*. [S.l.]: Grupa A, 2011. 25

JARA, A. J. et al. Extending the internet of things to the future internet through ipv6 support. *Mobile Information Systems*, IOS Press, v. 10, n. 1, p. 3–17, 2014. 13

MASCHIETTO, L. G. et al. *Arquitetura e Infraestrtutura de IoT*. Grupo A, 2021. Internet. [Accessed 12-Jun-2023]. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9786556901947/>. 19, 20

MATTOS, V. H. E.; BEGOSSO, L. R. Tecnologias ágeis para entrada de dados. 2018. 23

SALLES, S. R. d. Gestão estratégica de frota leve: o caso de são josé dos campos-sp. Universidade Tecnológica Federal do Paraná, 2011. 16

SANTOS, B. P. et al. Internet das coisas: da teoria à prática. *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuıdos*, v. 31, p. 16, 2016. 18

SATYABRATA. *Arquitetura de 5 camadas da Internet das coisas*. 2022. Internet. [Accessed 18-Jun-2023]. Disponível em: <https://acervolima.com/arquitetura-de-5-camadas-da-internet-das-coisas/>. 18, 33

TELECO. *RFID: Faixas de Frequências e Protocolos*. 2023. Internet. [Accessed 28-Jun-2023]. Disponível em: <https://www.teleco.com.br/tutoriais/tutorialrfid/pagina_3.asp>. 21

UVIASE, O.; KOTONYA, G. Iot architectural framework: connection and integration framework for iot systems. *arXiv preprint arXiv:1803.04780*, 2018. 18

VALADÃO, R. B. *Biometria - Reconhecimento de íris — gta.ufrj.br*. 2008. Internet. [Accessed 12-Jun-2023]. Disponível em: <https://www.gta.ufrj.br/grad/08_1/iris/index.html>. 23

VALENTE, A. M. et al. Gerenciamento de transporte e frotas. 2º edição. *Ed. Cengage Learning, São Paulo*, 2016. 13

VALENTE, A. M.; PASSAGLIA, E.; NOVAES, A. G. *Gerenciamento de transporte e frotas*. [S.l.]: Pioneira, 1997. 16, 17

VERMESAN, O. et al. Internet of things strategic research roadmap. *Internet of things-global technological and societal trends*, v. 1, n. 2011, p. 9–52, 2011. 13

Apêndices

APÊNDICE A – API RESTFUL

Tabela 3 – Tabela de utilização de requisição GET para o *backend*.

Caminho	**Ação**
/fichaStatus	Lista o status de todas as fichas de viatura
/relatorio/data	Retorna lista das ficha de viatura do mês e ano informado
/fichaMissao/id	Lista a ficha de viatura correspondente ao ID fornecido
/email/email	Retorna o Usuario correspondente ao email
/usuariosPendentes	Retorna a lista de todos os usuário que criaram cadastros e ainda não foram definidos seus respectivos perfil
/download/id	Retorna a foto do Usuario correspondente ao ID
/download	Retorna todas as fotos dos Usuarios com perfil motorista
/usuarioMotorista	Retorna a lista de todos os Usuarios que são motoristas
/listaReserva/id	Retorna objetivo Reserva correspondente ao ID
/listaReservaGestor	Retorna lista do tipo Reserva com todas as reservas pendentes vigentes
/listaSolicitacao/id	Retorna objeto Solicitação correspondente ao ID
/listaSolicitacaoInspecao/id	Retorna objeto Inspeção correspondente ao ID
/listaSolicitacaoAprovada	Retorna a lista de objetos Solicitação que estão com status aprovada
/listaSolicitacaoAguardandoAprovacao	Retorna a lista de objetos Solicitação que estão com status AguardandoAprovacao
/listaSituacaoViaturas	Retorna lista de objeto Viaturas com seus respectivos status
/listaTanque	Retorna lista contem os enum Tanque
/listaViaturasDisponiveis/saram	Retorna uma lista do tipo Viatura cujo status seja disponível para aquele motorista através do saram
/listaViaturasFora	Retorna lista de objeto Viaturas com seus respectivos status seja EmMissao
/listaViaturasReservada	Retorna lista de objeto Viaturas na qual estejam reservadas no dia
/listaModelos	Retorna lista de Modelos

Fonte: Do Autor.

Tabela 4 – Tabela de utilização de requisições POST e PUT para o *backend*.

Caminho	**Método**	**Ação**
/validaSaida	PUT	Consulta de o ID e a placa fornecidos estão respectivamente autorizados a sair e registra na ficha de viatura
/fechaFichaMotora	PUT	Insere dados necessários ao fechamento da ficha por parte do motorista
/fechaFichaAprovador	PUT	Insere o nome do aprovador que finalizou a ficha bem como altera o status da viatura para disponível
/gestorDePerfil	PUT	Define o perfil do usuário informado
/modificaStatusReserva	PUT	Altera o status da reserva informada
/modificaStatusSolicitacao	PUT	Altera o status da solicitacao informada
/adicionaInspecao	POST	Cria uma nova linha na tabela Inspeção com os dados fornecidos
/registraUsuario	POST	Cria um novo usuário com todos os perfils vazio na tabela Usuario
/loginUsuario	POST	Realiza a autenticação do login
/adicionaReserva	POST	Cria uma nova linha na tabela Reserva com os dados fornecidos
/adicionaSolicitacao	POST	Cria uma nova linha na tabela Solicitação com os dados fornecidos
/adicionaViatura	POST	Cria uma nova linha na tabela Viatura com os dados fornecidos
/adicionaModelo	POST	Cria uma nova linha na tabela Modelo com os dados fornecidos
/upload	POST	Insere uma foto no atributo data da tabela Usuario

Fonte: Do Autor.