CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ, como parte dos requisitos necessários à obtenção do grau de mestre.

Orientador:
DSc. João Roberto de Toledo Quadros

Rio de Janeiro,

Junho de 2024

PLATAFORMA PARA MAPEAMENTO DE ÁREAS DE RISCO PARA VEÍCULOS AUTOMOTORES BASEADA EM APRENDIZADO DE MÁQUINA ASSOCIADA À GEOLOCALIZAÇÃO

Dissertação de Mestrado em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ.

Jorge Eugenio Medeiros Anselmo

Aprovada por:

JOAO ROBERTO DE TOLEDO QUADROS:79616178768

Assinado de forma digital por JOAO ROBERTO DE TOLEDO QUADROS:79616178768
Dados: 2024.06.04 19:41:59 -03'00'

Presidente, Prof. DSc. João Roberto de Toledo Quadros, D.Sc. (orientador)

gov.br Documento assinado digitalmente
JOEL ANDRE FERREIRA DOS SANTOS
Data: 07/06/2024 10:22:06-0300
Verifique em https://validar.iti.gov.br

Membro 1, Prof. DSc. Joel André Ferreira dos Santos.

gov.br Documento assinado digitalmente
NADIA NEDJAH
Data: 07/06/2024 20:58:09-0300
Verifique em https://validar.iti.gov.br

Membro 2, Prof[a]. Phd. Nadia Nedjah

Rio de Janeiro,

Junho de 2024

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

A618 Anselmo, Jorge Eugenio Medeiros
Plataforma para mapeamento de áreas de risco para veículos automotores baseada em aprendizado de máquina associada a geolocalização / Jorge Eugenio Medeiros Anselmo. — 2024.
57f. + anexo : il. color. , enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2024.
Bibliografia : f. 54-57
Orientador: João Roberto de Toledo Quadros

1. Sistemas inteligentes de veículos rodoviários. 2. Internet das Coisas. 3. Aprendizado do computador. 4. Arduino (Controlador programável). I. Quadros, João Roberto de Toledo. (Orient.). II. Título.

CDD 623.893

Elaborada pela bibliotecária Tania Mello – CRB/7 nº 5507/04

DEDICATÓRIA

"Dedico primeiramente esse texto, a minha filha, Ana Luisa Duarte Anselmo, que é o motivo pelo qual luto todos os dias, e me faz tentar entregar uma versão melhor de mim a cada dia. Dedico também, aos meus pais, Jorge Anselmo e Sonia Medeiros Eufrásio, que cada um ao seu modo e tempo, vem me mostrando o poder da educação. Por fim e não menos importante, a minha esposa, Rubia Helena Barreto Duarte, que do seu jeitinho, me forneceu todo apoio nos momentos mais difíceis. Agradeço também aos meus irmãos e parentes que se foram, em especial a minha irmã: Pâmela Medeiros Eufrásio (em memória), que quando pequenos, era minha companheira de estudos. "

AGRADECIMENTOS

Ao meu orientador prof. João Roberto de Toledo Quadros, pelo grande apoio nessa caminhada, assim como, lá no início no processo seletivo, me mostrando que era possível um homem negro, morador da baixada fluminense, no Rio de Janeiro, conquistar mais esse grande feito, de estudar em uma das melhores universidades federais do país.

A professora de inglês Keyth Evelyn Menegaci Vasquez de Oliveira, que me ajudou no processo de preparação.

Aos meus familiares e demais amigos, que contribuíram deforma significativa nesta construção, me fornecendo todo o suporte necessário, formando com maestria a minha rede de apoio.

Aos demais discentes do programa, alguns em especial se tornaram amigos que levarei para a vida. Certamente, sem esses companheiros eu não teria chegado tão longe nessa jornada.

Por fim, e não menos importante, não poderia deixar de agradecer ao demais professores do programa, que me ajudaram na preparação para chegar até aqui.

A todos, meu mais sincero agradecimento, essa conquista também é de vocês!

RESUMO

Orientador: DSc. João Roberto de Toledo Quadros

Resumo da Dissertação submetida ao Programa de Pós-graduação em Ciência da Computação do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ como parte dos requisitos necessários à obtenção do grau de mestre.

Diante do ambiente de incertezas financeiras e redução nas oportunidades de trabalho, agravados ainda mais pela pandemia de covid 19, ocorrem, como uma das consequências o aumento no índice de criminalidades, em especial, roubos e furto de veículos. Dentro desse contexto, podem aparecer questões, tais como: Quais os modos para prevenir ou ajudar a recuperarem veículos furtados, ou roubados? Este estudo visa apresentar embasamento teórico e prático para uma proposta de uma solução IOT , que possibilitará aos usuários bloquearem seus veículos, sinalizando um roubo/furto , de modo que esses dados coletados sejam utilizados para um mapeamento, através de uma solução baseada em aprendizado de máquina, apresentando quais lugares são mais propensos a essas ocorrências de furtos ou roubos , permitindo criar alertas aos demais motoristas , podendo ainda, essa plataforma, ter sua base obtida compartilhada com demais setores da sociedade , como polícia, seguradoras entre outros.

Palavras-chave:
Rastreamento de veículos, Plataforma colaborativa IOT, Aprendizado de máquina e Arduíno

Rio de Janeiro,

Junho de 2024

ABSTRACT

Advisors: DSc. João Roberto de Toledo Quadros

Abstract of dissertation submitted to Programa de Pós-graduação em Ciência da Computação - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ as partial fulfillment of the requirements for the degree of master.

In faced of financial uncertainties and reduced job opportunities, further exacerbated by the COVID-19 pandemic, one of the consequences is the increase in crime rates, especially vehicle thefts and robberies. Within this context, specific problems may arise, such as: What are the ways to prevent or help in the recovery of stolen or robbed vehicles? This study aims to provide theoretical and practical foundation for a proposal for an IOT solution, which will enable users to block their vehicles, signalling a theft, so that the collected data can be used for mapping through a machine learning-based solution. It will identify areas more prone to these incidents of theft or robbery, allowing for the creation of alerts to other drivers. In addition, this platform can share its data with other sectors of society, such as police and insurance companies.

Key-words:
Vehicle tracking, Collaborative IOT platform, Machine learning and Arduino

Rio de Janeiro,

Junho de 2024

Lista de Figuras

II.1 Integração entre os componentes tecnológicos. Fonte: Elaboração própria 22

III.1 Representação da arquitetura proposta para a plataforma IoT de rastreamento, adequada a veículos automotores contendo os módulos físicos de conexão ao veículo (capaz de monitorar seus posicionamentos) a API e o Armazenamento (compondo o Back-End) e o aplicativo no dispositivo móvel compondo o front-end. Fonte: Elaboração própria 25

III.2 Tela mostrando as funcionalidades da API, que está em execução em servidor local, no qual o Swagger está instalado, contendo os acessos de login do usuário, veículos por usuário, cadastro de veículo, localização, listagem de viagem, tratamento do agrupamento e gravação das ocorrências. Fonte: Autoria e obtenção própria 27

III.3 Modelo Entidade Relacionamento(MER). (mostrando as tabelas e seus relacionamentos) que contém os dados usados para cálculos da Fórmula de Haversine, tais como, usuários, veículos, ocorrências, permissões, histórico e tipo de ocorrência. Fonte: Elaboração Própria 30

III.4 Esquema eletroeletrônico do hardware do Módulo Veículo, contendo a conexão com aa bateria de 12v, GPS, a interface do microcontrolador, relé e display . Fonte: Elaboração Própria 37

III.5 Representação das ligações eletrônicas entre os componentes, mostrando as pinagens utilizadas entre eles. Fonte: Elaboração Própria 37

III.6 Módulo Esp32 básico, com a pinagem usada e demais componentes acopladas ao chip-Fonte: Biblioteca do autor 38

III.7 Módulo GPS da família NEO-xM, do tipo GY-NEO6MV2.
Fonte: https://www.masterwalkershop.com.br/modulo-gps-gy-neo6mv2-com-antena 40

III.8 Dashboard do usuário com gráficos, sendo que o primeiro, mostra a comparação das distâncias percorridas nas viagens diárias e comparativo das distâncias percorridas mensalmente s. Fonte: Biblioteca de imagens do autor 41

III.9 Listagem de viagens do efetuados Fonte: Biblioteca de imagens do autor 41

III.10Exemplo de uma viagem efetuada Fonte: Biblioteca de imagens do autor 42
III.11Localização em tempo real do veículo 42
III.12Tela da interface com a visão da opção “Meus veículos” e o resultado com os veículos do usuário. Fonte: Biblioteca de imagens do autor 43
III.13Exemplo de locais que ocorreram bloqueios classificados como área de risco Fonte: Elaboração Própria 43

IV.1 Fluxo de execução para casos de bloqueio manual do veiculo, no qual a operação é armazenada no módulo BD, servindo de base para aplicação do Machine Learning posterior Fonte: Elaboração pelo autor 46
IV.2 Fluxo complementar para controle de bloqueio, executado de forma autônoma, com acesso ao mecanismo de Machine Learning, para identificar se o local do veiculo está de acordo com o padrão classificado à ele como locais permitidos. Caso o local identificado não esteja classificado como permitido e o usuário não o confirmou alteração, o mecanismo de treinamento vai classificar o local como não permitido e enviará a solicitação de bloqueio- Fonte: Elaboração pelo autor 46
IV.3 Visão geral do protótipo construído. Fonte: Elaboração própria 48
IV.4 Tela Inicial da A‘PI, com acesso aos Endpoints por ela controlados, “consumidos” pelo Módulo Veículo e o módulo Aplicativo WEB. Fonte: Biblioteca de Imagens do Autor 48
IV.5 Apresentação do teste de requisição e autenticação de um Endpoint, usando o método postman. Fonte: Biblioteca de imagens do autor 49
IV.6 Imagem da página do https://uolhost.uol.com.br/revenda-de-hospedagem, apresentando o custo de assinatura mensal de um ambiente de hospedagem para a Plataforma.. Fonte: Biblioteca de imagens do autor 50
IV.7 Cenário de cálculo de distância de forma retilínea, no qual os vértices representam os pontos de início e destino do deslocamento, enquanto as arestas representam as distâncias entre os deslocamentos. Fonte: Elaboração do autor. 52
IV.8 Cenário de cálculo de distância entre os pontos de deslocamento X e Y da IV.7, usando a solução de calcular sempre o ponto 10 segundos para trás. Fonte: Elaboração própria do autor. 52

V.1 Transformador/Inversor de corrente para adaptação a parte eletroeletrônica do veículo Fonte: Biblioteca de imagens do autor 53
V.2 Fonte 5V/1A com pino P4 para ser ligada ao inversor/transformador Fonte: Biblioteca de imagens do autor. 54

V.3 Tela do MS Visual Studio que apresenta o momento do processo de compilação para posterior publicação do código no site de hospedagem. Fonte: Biblioteca de imagens do autor. 54
V.4 Tela e para edição dos dados de segurança e armazenamento (através do FTP FileZilla) do código compilado no servidor. Fonte: Biblioteca de imagens do autor 55
V.5 Cena mostrando que a Plataforma identificou 39 Km/H em comparação com o que o radar identificou, no dia 31/01/2024, as 8h do amanhã, Barra da Tijuca-RJ Fonte: Biblioteca de imagens do Autor. 56
V.6 Relatórios de viagens em um mês específico, apresentando as linhas de histórico inseridas, a média de velocidade por viagens, distâncias totais percorridas e latitudes/longitudes do inicio e final d ecada viagem. Fonte: Biblioteca de imagens do Autor. 57
V.7 Gráfico gerado pelas Plataforma identificando as viagens, por dia, com as distâncias percorridas para cada viagem identificada, sendo que, assim, alimenta-se o processo de treinamento da Plataforma e permite-se que o algoritmo de Machine Leraning possa classificar como sendo uma viagem padrão ou não. Fonte: Biblioteca de Imagens do Autor. 58
V.8 Gráfico de média de distâncias percorridas em um dia (com vários períodos, tais como, pela manhã, tarde e noite), gerado pela Plataforma para fins de verificação do usuário (servindo para ver o quanto se gasta de quilometragem por um período determinado. Fonte: Biblioteca de imagens do Autor. 58
V.9 Tela de cadastro de usuários e configuração de bloqueios Fonte: Biblioteca de imagens do Autor. 59
V.10 Relatório gearado pela Plataforma apresentando as 1.228 viagens coletadas, que gerou 184.710 registros inseridos automaticamente no BD da Plataforma Fonte: Biblioteca de Imagens do Autor 59
V.11 Apresentação gráfica das distâncias percorridas pelo veículo controlado pela Plataforma, sendo indicado mês a mês. Dentro de um período especificado (no caso 11 meses) Fonte: Biblioteca de imagens do Autor. 60
V.12 Relatório de bloqueio disponibilizados nos 11 meses de testes da Plataforma, gerado pela procedures/métodos do BD, mostrando os bloqueios (e os tipos, se automático ou não) durante o período Fonte: Biblioteca de Imagens do Autor 60
V.13 Cena mostrando que a Plataforma identificou 39 Km/H em comparação com o que o radar identificou, no dia 30/03/2024, as 8h do amanhã, Barra da Tijuca-RJ Fonte: Biblioteca de imagens do Autor. 62

V.14 Rota do dia 17/02/2024 vista através da ferramenta Google Maps, no qual o Maps mediu 154 km e a Plataforma mediu 157 km de distância percorrida . Fonte: Elaboração Própria 63
V.15 Gráfico gerado pela Plataforma que apura distância média percorrida em dois meses consequentes, identificando as corridas por dia da semana. Fonte: Biblioteca de imagens do Autor. 65
V.16 Outro gráfico gerado pela Plataforma mostrando a distância média percorrida em outros dois meses consequentes, confirmando que os dias da semana de maior movimentação do veículo são terça e quarta, Fonte: Biblioteca de imagens do Autor. . . . 65
V.17 Grafico apresentando do mês de fevereiro de 2024 (por dias d euso do veículo), indicando quel dia houve mais viagens pela parte da manhã, no caso no 25/02/2024. Fonte Biblioteca de imagens do autor. 66
V.18 Gráfico apresentando aa distância média percorrida no mês de fevereiro de 2024, por dia da semana e período do dia. Fonte Biblioteca de imagens do autor. 66

Lista de Tabelas

II.1 Comparativo entre as abordagens Fonte: Elaboração própria 20

III.1 Descrição das entidades vistas na Figura III.3. 31
III.2 Descrição das entidades do componente Banco de Dados 32
III.3 Descrição das entidades do componente Banco de Dados 33
III.4 Apresentação de alguns dos métodos contidos no BD do Back-end 34
III.5 Apresentação de agrupamentos tratados no método/procedure ListViagensAgrupamento 36

IV.1 Tabela com os demais componentes utilizados na confecção do protótipo operacional utilizado nos testes de campo 44
IV.2 Especificação dos custos de cada componente de hardware ou físico relativco ao protótipo operacional. Fonte: Sites da Internet 45
IV.3 Sinais do LED e seus significados para a Plataforma 47

V.1 Comparação dos dados obtidos pela GPS da Plataforma com os extraídos do Google Maps da viagem de 17/02/2024 63

Lista de Abreviaturas e Siglas

Abreviatura	**Significado**
A	Amper
AC	Alternating Current (Corrente Alternada)
API	Application Programming Interface (Interface de programação de aplicações)
CFTV	Circuito Fechado de Televisão
CI	Circuito Integrado
DC	Direct Current (Corrente Contínua)
FTP	File Transfer Protocol (Protocolo de Transferência de Arquivos)
GIT	Sistema de Versionamento de Código
GSM	Global System for Mobile(Sistema global para Mobile)
GPS	Global Position System (Sistema Global de Posicionamento)
HTTP	Hyper Text Transfer Protocol (Protocolo de Transferência de Hipertexto)
IDE	Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
IA	Inteligência artificial
IOT	Internet Of Things (Internet das Coisas)
JSON	JavaScript Object Notation (Notação Objeto JavaScript)
LGPD	Lei Geral de Proteção de Dados
MER	Modelo Entidade Relacionamento
ML	Machine Learning
POST	Resquest Method by HTTP
REST	Representational State Transfer (Transferência de Estado Representacional)
RFID	Radio Frequency Identification (Identificação por Rádio Frequência)
SDK	Software development kit (Kit de Desenvolvimento de Software)
SOAP	Simple Object Access Protocol (Protocolo Simples de Acesso a Objetos)
VAC	Volts Alternating Current (Volts em Corrente Alternada)
VDC	Volts Direct Current (Volts em Corrente Contínua)
XML	Extensible Markup Language (Linguagem de Marcação Extensiva)
WAN	Wide Area Network (Rede de Área Ampla)
WEB	World Wide Web (WWW ou Web)
WI-FI	Wireless Fidelity(Fielmente sem fio)

Sumário

I Introdução **16**

II Trabalhos Relacionados **19**

II.1 Uso de GPS para auxiliar no monitoramento de veículos 19

II.2 Comparativo entre as principais abordagens e este trabalho 20

II.3 Microcontroladores 20

II.4 IoT Para Veículos 21

III Sobre Projeto **23**

III.1 Motivação 23

III.2 Metodologia do Desenvolvimento 23

III.3 Tecnologias escolhidas 24

III.4 Arquitetura 25

III.5 Detalhamento da Camada Back-end 25

III.5.1 Módulo API 26

III.5.2 Banco de dados 30

III.5.3 Modelagem do banco de dados 30

III.5.4 Módulo Veículo 36

III.6 Detalhamento da Camada Front-end 40

III.6.1 Aplicativo Web 40

IV Preparação do Protótipo **44**

IV.1 Descrição básica do protótipo 44

IV.2 Etapas da construção do protótipo da Plataforma 45

IV.2.1 Fluxos de integração dos módulos 45

IV.2.2 Funcionalidades do protótipo 47

IV.2.3 Custos de Hospedagem 50

IV.2.4 Problemas encontrados e soluções propostas 51

V Testes do protótipo: obtenção e análise de resultados **53**

V.1 Montagem de Ambiente de Teste 53
V.1.1 Descrição do domínio usado para os testes 55
V.1.2 Coleta de dados para armazenamento e operações de bloqueio 56
V.1.3 Comparações de aferição de medição de velocidade média do veículo 61
V.1.4 Comparações de aferição de distância 62
V.1.5 Classificações de padrões de viagens 64

VI Considerações Finais **67**

Referências Bibliográficas 69

Capítulo I Introdução

Desde o início do século XX, a introdução de novos fatores de desenvolvimento associados a veículos automotores fez nascer e crescer o uso profissional deste tipo de máquina. Ao longo do tempo, esses veículos vieram a ser usados por motoristas especializados em transporte pessoal, no início apenas os chamados taxi-drivers e, posteriormente, no início do século XXI, os chamados motorista de aplicativos, além do surgimento de novas tecnologias de impulsão por motores Farias et al. [2022].

Segundo um estudo do IPEA, existem cerca de 1,5 milhão de novos motoristas com seus veículos, atuando nessas novas modalidades de uso dos mesmos FGV [2020]. Um ônus, do crescimento de uso de veículos nas ruas e estradas, é a perda desses, devido ao aumento de roubos ou furtos FBSP [2022].

Esses eventos de roubos, que tem aumentado nas grandes cidades, ocasionam impactos sociais significativos, pois, como visto, uma parcela significativa da sociedade atual utiliza esses veículos automotores como ferramenta de trabalho. Segundo os dados do anuário brasileiro de segurança publicados em 2021, os roubos e furtos tem se apresentado com um número crescente a cada ano, isso diante de outras modalidades de roubos e furtos, segundo um comparativo de 2020-2021 FBSP [2022].

Associado ao crescimento do uso de veículos como instrumento de trabalho, acompanha-se também um maior uso de dispositivos móveis (tipo smartphones), com uma média de 1,5 dispositivos por pessoa no Brasil FGV [2020], pois essa forma de comunicação é utilizada justamente para se ter acesso aos aplicativos, que trabalham junto aos motoristas de transporte de pessoas.

Considerando o número de veículos furtados ou roubados, os impactos que isso causam na sociedade como um todo e percebendo na atualidade novas tecnologias que permitem controle e rastreamento, observou-se em pesquisas (cujos resultado estão nos próximos itens dessa dissertação), que assim como aumentaram os problemas, também vieram acompanhadas soluções para o mesmo.

Isso inclui o desenvolvimento de novas metodologias e tecnologias para tratar desse rastreamento, com identificação on-line dos veículos e sua geolocalização, utilizando ferramentas baseadas em IoT, GPS e algoritmos inteligentes, modelados por Deep Learning ou Machine Learning. Percebe-se que tais instrumentos também podem ser úteis para alimentar a base de informações sobre furtos e roubo de veículos utilizada pelos órgãos de segurança públicas dos Estados da União

FBSP [2022].

Desta feita, a proposta deste trabalho, que pretende apresentar um caminho de solução para esse problema, é o desenvolvimento de uma plataforma baseada em IoT, com uso de microcontroladores e interfaces de baixo custo (do tipo, Arduino Nano ou Uno), que, junto a um algoritmo de obtenção de dados, possa ser utilizado para mapeamento de áreas de risco de furtos e roubos associados aos veículos automotores. A ideia também é o desenvolvimento do componente de geolocalização com a tecnologia de IA baseada em Machine Learning.

A plataforma desenvolvida envia sinais para bloquear o veículo, de modo que se impeça a continuidade do furto ou roubo, sem afetar a segurança física do motorista, já que os dispositivos móveis poderão ser usados, a qualquer tempo, para atuar nesse controle, aumentando a segurança indicada na proposta desse trabalho.

Além disto, a plataforma pretende ser um instrumento de manipulação de dados de forma colaborativa, obtendo dados dos veículos (tais como o número de chassi, usuário atual ou de fontes públicas), associando-os aos dados de geolocalização e criando uma base de dados a ser usando no âmbito da segurança pública. O foco secundário é que possa ser possível usar os dados da plataforma para mapear ambientes e classifica-los de acordo com o risco de assalto, de modo a atuar de forma colaborativa junto a autoridades e órgãos de segurança.

Uma outra ideia é que, uma vez que os dispositivos móveis de comunicação já estão, de certo modo, integrados à vida dos motoristas, a plataforma também possa servir, via Wi-Fi, para enviar sinais de bloqueio, de forma que os veículos furtados possam ser paralisados nos momentos muito após o furto ou roubo, impedindo a continuidade do processo, sem riscos para o motorista.

O trabalho de Harbinarka [2022] é um exemplo do uso de dados de GPS para mapear zonas públicas. No caso, os dados foram utilizados para verificar as zonas onde ocorreriam mais acidentes em regiões específicas de Bangladesh. Também existe o trabalho de Letto et al. [2022], que utiliza dados de geolocalização para atuar junto a segurança de motoristas em estradas perigosas do USA.

Baseado nesses estudos, e em outros, foi traçado como objetivo principal deste trabalho o desenvolvimento de uma plataforma que permita a um motorista de veículo automotor localizar o seu veículo em qualquer situação, e, quando houver roubo ou furto, que ele possa também enviar para essa plataforma, via dispositivo móvel, sinais de bloqueio do veículo e sinalizar o roubo ou furto, enviando esses dados para algum ambiente de segurança pública ou privada, permitindo o acompanhamento da localização do veículo após o incidente.

Outra funcionalidade proposta é que, havendo a sinalização de roubo/furto do veículo, a plataforma vai armazenar essa informação, juntamente com a localização, em uma a base de dados em nuvem. Essas informações poderão ser acessadas para autoridades de segurança, ou outros motoristas, prevendo o uso nas tomadas de decisões sobre segurança de veículos e marcação de

áreas perigosas ou de risco, com identificação de movimentações atípicas, e com a criação de uma base de conhecimento para "aprender" sobre o perfil destes eventos na cidade do Rio de Janeiro.

Importante destacar, que comceber uma solução baseada principalmente na segurança, é um importante propósito para oferecer a comunidade uma solução confiável, mas além desse objetivo , visa-se também, perseguir os seguintes objetivos: apoiar a área de segurança pública no mapeamento de áreas de risco, utilizando-se tecnologias como IOT, como dito anteriormente.

Outro importante objetivo, foi sem dúvidas, construir um projeto economicamente viável, de modo a propiciar ainda mais a sua difusão e consequentemente o impacto social esperado. Para tanto, se fez necessário utilizar o máximo possível, soluções de código aberto, ou opensource, reduzindo significativamente o custo de construção desse projeto.

Para o desenvolvimento desta plataforma proposta, na parte de hardware, foi realizado uma pesquisa bibliográfica sobre uso de microcontroladores e tecnologias de implantação de dispositivos IoT, visando àqueles projetos que focaram em uso de geolocalização com utilização de dados GPS.

Na parte de software, foram estudados e aplicados algoritmos utilizados em geolocalização, e também, algoritmos baseados em aprendizado de máquina. O primeiro tipo de algoritmo corresponde aos critérios matemáticos para geolocalização; já a parte de aprendizado é para construção e classificação dos dados obtidos dentro dos critérios de segurança que se pretende trabalhar.

Esse trabalho possui uma introdução; trabalhos relacionados; o desenvolvimento da plataforma, com descrição da proposta de software e hardware; uma proposta inicial de cronograma para o protótipo e continuidade da dissertação e as considerações finais.

Capítulo II Trabalhos Relacionados

Conforme descrito, foi realizado uma pesquisa bibliográfica procurando trabalhos de uso das tecnologias de plataformas com GPS e microcontroladores, com foco nas aplicações com veículos automotores, além de busca de trabalhos com uso de algoritmos de geolocalização e aplicações de aprendizado de máquina em manipulação de dados de geolocalização (para eventual classificação).

A metodologia aplicada se utilizou da técnica de mapeamento sistemático, através de consulta as bases da Scopus e Google Scholar, utilizando as strings de busca: "mobile Anti-theft car system AND Machine Learning "; "Anti-theft car system AND microcontrolers" e "Haversine Algoritmics".

Essas consultas retornaram cerca de 72 artigos, dos quais foram selecionados, para pesquisa mais acurada, aproximadamente 20 artigos, voltados para a área de área de ciência da computação e robótica (devido aos projetos de IoT identificados na pesquisa).

II.1 Uso de GPS para auxiliar no monitoramento de veículos

No trabalho de [Lee et al., 2014] , foi realizado uma pesquisa sobre o monitoramento de veículos, utilizando a interface Arduino Uno, acoplada a um conjunto de módulos GPS, com um aplicativo do tipo mobile, de modo que tal plataforma pudesse coletar dados dos módulos GPS e notificar os usuários através de envio de mensagens a dispositivos móveis (utilizando SMS), possibilitando sinalização de bloqueio do veículo, com comunicação interna via bluetooth.

Nesse trabalho, vislumbra-se um problema na aplicação dessa abordagem, em relação ao do trabalho vigente. No caso de Lee et al. [2014], o controle é efetivado exclusivamente por um dispositivo móvel conectado a um arranjo simples de microntrolador (o Arduino UNO), de modo que, se o dispositivo móvel também for subtraído, não há como monitorar o veículo. No caso do trabalho vigente, o dispositivo pode se conectar visando bloqueio, mas a geolocalização não prescinde do uso do dispositivo móvel.

Em Mounika and Chepuru [2019], a abordagem visa inserir controle de rastreamento de veículos, tendo também como interface base o Arduino, mas com uso de cartão de proximidade RFID para bloquear/desbloquear as funcionalidades do mesmo, contudo essa abordagem se baseia em proximidade de comunicação, não fazendo uso de controle via wi-fi, ou seja, não há previsão de solução, caso haja problemas de comunicação.

A proposta apresentada por Tuyisenge et al. [2021], trabalha com a prevenção de furtos através de um software de controle, instalado em um smartphone, no qual, caso o usuário não realize a operação de liberação plena, independente de ocorrência de furto/roubo, a plataforma vai inabilitar o acesso ao veículo, cortando o fornecimento de energia para a bomba de combustível. Essa proposta é uma proteção de acesso, mas não utiliza dados de geolocalização de longa distância.

Já sobre a utilização de algoritmos específicos para geolocalização, em Garey and Johnson [1979] é apresentado uma solução matemática para o problema de identificar o caminho de menor deslocamento de um ponto "A" para um ponto "B", usando algoritmo do tipo NP- Completo, que se apresenta com execuções polinomiais de complexidade mais elevada, ele se baseou na utilização da fórmula Haversine.

Essa aplicação se tornou básica em qualquer sistema que faz uso de dados de geolocalização, como, por exemplo, o visto em Hizham and Ginardi [2021], no qual se apresenta uma variação do uso da fórmula algorítmica da fórmula Haversine, de modo a calcular à distância entre dois pontos, com base no comprimento de uma linha reta, utilizando dados de latitude e longitude prognosticada, para localização de veículos marítimos e similares.

II.2 Comparativo entre as principais abordagens e este trabalho

Agrupando-se os principais trabalhos que guardam similaridade temática com a presente dissertação, e que foram citados anteriormente, compõe-se uma tabela comparativa entre as abordagens, conforme pode ser visto na Tabela II.1 :

Tabela II.1: Comparativo entre as abordagens Fonte: Elaboração própria

Função	Lee et al. [2014]	Tuyisenge et al. [2021]	Mounika and Chepuru [2019]	Presente Trabalho
Baixo Custo	1	1	1	1
Bloqueio automático	0	0	0	1
ML	0	0	0	1
Georeferênciamento	0	0	0	1
App de controle	0	1	0	1
Compartilhamento de dados	0	0	0	1
Colaboratividade	0	0	0	1

II.3 Microcontroladores

Para procurar a interface mais adequada (tanto em termos de custo, quanto de tecnologia) utilizada para controle de veículos, foram observados diversos trabalhos, com o uso de microcontroladores com GPS ou para trabalhar com controle à distância.

No trabalho de Monawar et al. [2017] é visto o uso de microcontroladores, do modelo da família AVR, tais como o ATMega328P, para agir como interface de controle autônomo, de modo a controlar o ligamento ou desligamento de um motor de um veículo através de estímulos de tempo, no qual um motor de partida de carro é ligado dentro de intervalo de tempo e desligado após outro intervalo temporal.

Em Tuyisenge et al. [2021] é observado um trabalho sobre o uso da interface ESP32 (baseado no microcontrolador Tensilica Xtensa LX6), com seu uso como interface de rádio frequência para tecnologia WiFi e também para controle remoto de dispositivos IoT. Nesse trabalho observou-se o uso dessa interface para controles à distância, o que seria o modelo desse trabalho vigente (controle a uma distância maior).

Como o foco dessa dissertação também é a manipulação de dados de geolocalização, faz-se importante observar trabalhos que utilizaram dispositivos de tipo GPS, com dados geolocalizáveis, e como eles desenvolveram aplicativos e interface para trabalhar com esse tipo de módulo.

Em Monawar et al. [2017] é observado um estudo sobre as funcionalidades de um módulo GPS genérico, verificando como o mesmo pode fornecer informações de localização em qualquer hora ou lugar no planeta, ou perto dela, utilizando captação de sinais, denominados sinais de presença de linha direta de visão no qual se utiliza a conexão e intersecção com pelo menos quatro satélites de acesso aberto.

Ainda no trabalho de trabalho Monawar et al. [2017] é também observado o uso de módulos, denominados Sistema Global para Comunicações (GSM), sendo o GSM um padrão de comunicação de um dispositivo qualquer com a Internet. Observou-se que tal tecnologia existe desde a implantação do 2G para redes de dispositivos móveis. Monawar et al. [2017] observaram que o padrão GSM pode permitir que informações do tipo GPS possam ser transmitidas e recebidas em um ambiente de rede WAN (de longa distância), tais como, a própria Internet.

Já em Tsindeliani et al. [2022], é visto outro conjunto de bibliotecas para desenvolvimento de sistemas com GPS, o Microsoft ML (MML), que possui uma estrutura de aprendizado de máquina com código aberto. Esta tecnologia de software baseia-se em um ambiente de DataView Black et al. [2020], que serve para modelagem de ambientes geográficos, ajustando um conjunto de tabelas, em uma estrutura big data, com base em banco de dados relacionais. Com o MML, pode-se criar um modelo geográfico com dados de predição, utilizando uma linguagem de dados similar ao R.

II.4 IoT Para Veículos

Em relação a aplicativos para IoT de controle à distância, foi estudado o Mobile App Qadir and Cooper [2020], que é utilizado no ambiente Google Futter, desenvolvido para dispositivos móveis, cuja tecnologia, oferece acesso multiplataforma para controle IoT, tanto para Android, como para

IOS. Esta tecnologia permite que seja possível criar um ambiente de controle à distância de outros dispositivos com aplicações desse software.Embora estudado, optou-se por seguir com a tecnologia web, por levar em consideração a curva de aprendizado.

Para não haver prejuízos a usabilidade, optou-se por utilizar então, a tencnologia PWA , que consiste em criar um webapp instalável, como pode ser visto no trabalho de Bowman [2023] , que apresenta um comparativo e justifica quando usar cada abordagem.

Em Tsindeliani et al. [2022] pode-se observar um desenvolvimento de um sistema GPS para dispositivos com Android baseado na biblioteca Web API, utilizado para integração entre sistemas através de mecanismos e padrões específicos da Internet W3C.BR [2020].

É apresentado, nesse trabalho, uma arquitetura de integração com Web API, que acopla diferentes tipos de componentes, aplicando-se o conceito de internet das coisas, ou IOT. Isso pode ser visto na II.1.

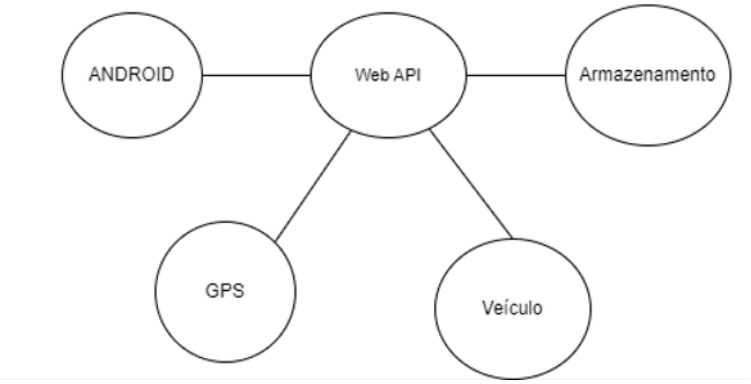


Figura II.1: Integração entre os componentes tecnológicos. Fonte: Elaboração própria

Em Mounika and Chepuru [2019] , é analisado uma interligação via internet de componentes tecnológicos, atuando como IoT para desenvolver dispositivos embarcados em veículos marítimos, interligando-os como uma rede e controlando-os via Internet, com dispositivos móveis.

Em Desai and Phadke [2017], aplica-se o conceito de IoT, no qual, utilizam-se de plataforma de eletrônica, para coletar e armazenar dados de velocidade, consumo de combustível, temperatura, entre outros parâmetros de um veículo escolhido, para então disponibilizá-los na nuvem.

Em Khan et al. [2022], viabilizou-se um dispositivo IoT sendo, um monitor de velocidade veicular, visando reduzir índices de acidentes ocasionado por excesso de velocidade. Neste trabalho, foi projetado uma espécie de radar que faz a leitura do código do veículo na entrada da via e o mesmo equipamento posto na saída, possibilita calcular a diferença de tempo entre a entrada e a saída na via.

Capítulo III Sobre Projeto

III.1 Motivação

Baseado no exposto anteriormente, a exemplo do observado na tabela II.1, observou-se que apesar de trabalhos voltados para essa área, a implementação de soluções de controle de veículos se faz necessário. Há uma clara lacuna de soluções que tragam mais autonomia, colaboratividade e compartilhamento de informações entre usuários e demais atores da sociedade. Para tanto, se faz necessário conceber solução acoplada a aplicativos multiplataformas, uso de georreferenciamento, aprendizado de maquina e IOT.

Como visto na introdução, os aspectos de segurança tendem a ser mais preciso com o uso de mapeamento de geolocalização e isso é um dos conceitos a ser utilizado no desenvolvimento da plataforma de mapeamento de áreas de risco para veículos automotores baseada em aprendizado de máquina associada a tecnologias de geolocalização.

Como existe uma crescente demanda na sociedade por soluções que contribuam para segurança pública (contra roubos e furtos de veículos), faz-se importante envolver o mundo acadêmico na busca de soluções cientificamente comprovadas que satisfaçam tanto a demanda de segurança pública, quanto a sensação particular de segurança.

A ideia de compartilhar parte desse controle entre o usuário comum e o usuário da segurança pública, seja pelo mapeamento das regiões de roubos ou furtos, seja pela capacidade de bloqueio (e assim reduzindo a área de risco) faz com que a solução proposta se encaixe nas necessidades tanto acadêmicas quanto necessidades leigas, tais como a segurança pública, e até a necessidade de mercado (para segurança privada de veículos automotores).

Os dados obtidos por esse tipo de plataforma podem ser usados também para construir uma base centralizada de informações que possibilite identificar regiões onde mais há essas ocorrências, regiões para onde mais são levados os veículos após a subtração e possibilitar o aumento de segurança para população em geral, de forma colaborativa.

III.2 Metodologia do Desenvolvimento

O desenvolvimento deste trabalho segue uma metodologia básica de desenvolvimento de sistemas, seguindo etapas, tais como:

• Planejamento da plataforma: estudo sobre os componentes necessários para o funcionamento da solução

• Busca de dados Impíricos: De acordo com a literatura, os dados podem ser obtidos a partir de buscas nos orgãos oficiais, por meio de fontes públicas, como registros de ocorrências policiais e dados de seguros de veículos Tursunbayev et al. [2021]. Os dados serão limpos e processados para garantir a qualidade dos dadosZheng et al. [2019].

• Tratamento de dados: Tratamento de dados obtios, como por exemplo, buscar as coordenadas geográficas das ocorrências a apartir do endereço.

• Treinamento do Modelo: Após tratamento dos dados, deve-se utiliza-los para treinar o modelo de predição, serão utilizados algoritmos de aprendizado de máquina para treinar modelos que possam prever a probabilidade de roubo de veículosBraga et al. [2019].

• Testes em campo: Propõe se validar se a aplicação de fato notificará após aproximar-se dos locais mapeados nas etapas anteriores.

• • Avaliação dos Resultados: Após a etapa de testes, se faz necessário validar a qualidade dos dados obtidos, assim como o funcionamento do modelo de predição, para apontar os resultados finais da proposta.Pode-se avaliar os modelos por meio de validação cruzada e outras técnicas de avaliação de desempenho [Zheng et al., 2019].

III.3 Tecnologias escolhidas

A escolha da linguagem miscrosoft Csharp(C#) , como linguagem principal, linguagem esta que guarda grande similaridade com a linguagem C que pode ser utilizada para programar os módulos API e Controladores, reduzindo a curva de aprendizagem. modular que pode criar novos módulos de API e controladores de microserviços.

Em relação a parte de microcontroladores e interfaces, a princípio, optou-se, para o modelo de teste final, pela plataforma ESP32, que já possui um ambiente de Wi-Fi integrado. Contudo, em paralelo, foi desenvolvido um protótipo com outro microcontrolador, com objetivo de auxiliar a configuração final da integração com os sistemas automotivos do veículo, que foi baseado na interface de desenvolvimento denominada Arduino NANO, com um módulo Esp8266 integrado.

O uso do protótipo com Arduino Nano foi para permitir pré-configurações junto aos veículos utilizados nos testes da plataforma, visando correção de erros na obtenção de sinal Wi-Fi, erros com o GPS e outros erros, inclusive de carga elétrica, pois, caso o Arduino Nano “queimasse” por um pico de energia do motor, ele se apresenta em um custo mais baixo do que os módulos ESP32, e qualquer dano a ele seria mais fácil de compensar.

Deste modo, foi facilitado a implementação final através do ESP32. Quanto ao módulo GPS, foi adquirido o GPS NEO-6M, que é capaz de informar a localização exata do objeto em que o

módulo esteja instalado, enviando dados referentes a latitude e longitude, data, hora e velocidade de deslocamento, e permite integração com redes Wi-Fi.

III.4 Arquitetura

Seguindo padrões de desenvolvimento de software já consolidados, soluções tecnológicas geralmente apresentam duas macro camadas de comunicação: a camada de front-end e a camada de back-end. A camada de front-end é a camada no qual o usuário interage, enquanto, a camada de back-end é a camada que contém geralmente as regras de negócios, controle de operações e armazenamento de dados, que por sua vez, dependem de estar instaladas em servidores em nuvem para ser consumida por usuários e/ou por recursos de IOT [Barbosa, 2023]. Na figura III.1. é apresentado o modelo proposto da arquitetura da plataforma.

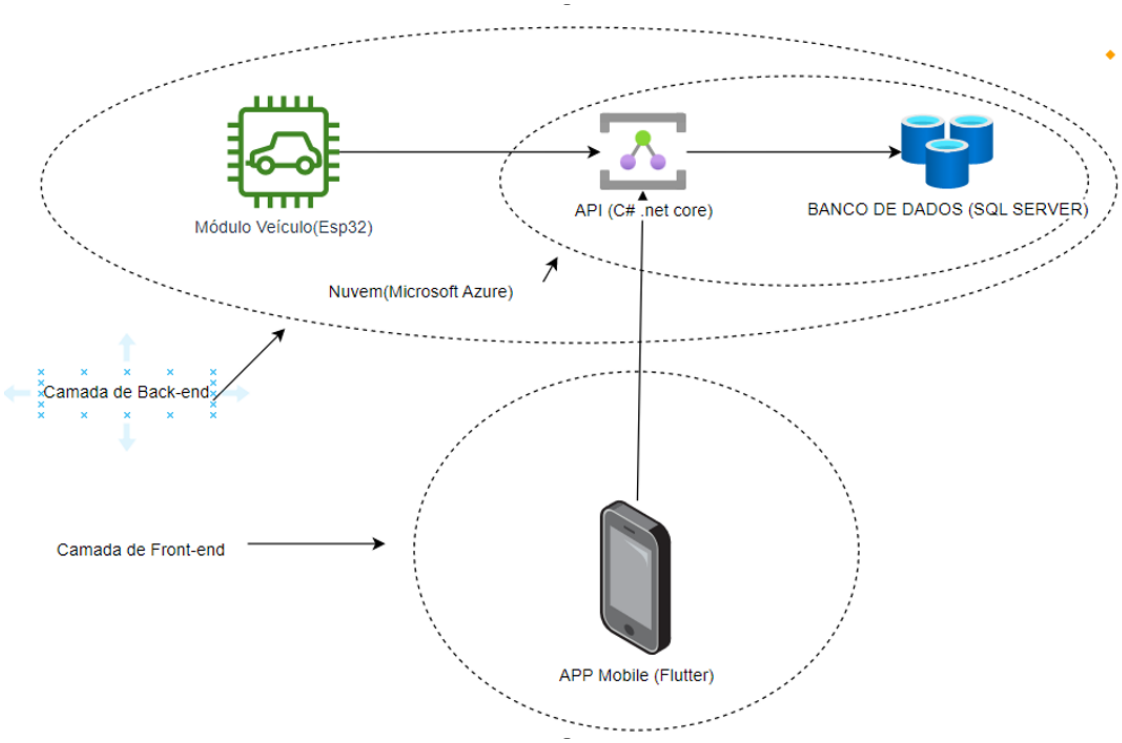


Figura III.1: Representação da arquitetura proposta para a plataforma IoT de rastreamento, adequada a veículos automotores contendo os módulos físicos de conexão ao veículo (capaz de monitorar seus posicionamentos) a API e o Armazenamento (compondo o Back-End) e o aplicativo no dispositivo móvel compondo o front-end. Fonte: Elaboração própria

III.5 Detalhamento da Camada Back-end

Esta camada contém os módulos API, banco de dados e de hardware do veículo. Este módulo é o responsável pelas regras de negócio da aplicação. Esse olhar de forma independente das camadas facilitam a manutenção, já que desacopla a visão da camada de apresentação da que contém as regras de negócio.

III.5.1 Módulo API

A API (Interface Programação de Aplicações) foi programada como uma aplicação web que vai depender de um servidor web para ser acessada pelos usuários , trabalhando com a interface principal entre os demais componentes do back-end, que são: Banco de Dados e o Módulo Veículos.

Essa API foi desenvolvida, como dito anteriormente, com uso da linguagem C# .net 6.0, e tecnologias associadas, que é um ambiente de desenvolvimento da Microsoft utilizado tanto no mercado quanto em projetos científicos. Nesse caso, utilizou-se a IDE (Ambiente de Desenvolvimento Integrado) denominada MS Visual Studio 2022, que também facilita no trabalho com o ambiente de nuvem, pois é integrada ao Microsoft Azure A Figura 3 apresenta uma tela da referida IDE, com parte do código da API desenvolvida.

Neste módulo estão implementadas a comunicação com o banco de dados, assim como a parte do código que contém o envio periódico das requisições para o hardware do veículo. Essas requisições são do tipo "HTTP POST", e servem para construir a consulta da situação de se manter, ou interromper, o funcionamento do veículo rastreado.

Este modelo de comunicação, tanto do módulo veículo quanto do aplicativo do cliente, é baseado no framework REST (Resource State Transfer), que é um padrão baseado em no formato de texto JSON, padrão este, que consome 73% menos banda móvel que o seu precursor SOAP, que é um padrão baseado no formato XML e que foi muito utilizado no mercado até meados de 2016 Garcia and Abilio [2017].

Para facilitar a usabilidade da API, assim como nos testes, uma opção foi utilizar a SDK de documentação Swagger, que é uma SDK de código aberto baseada na especificação da OpenAPI, que permite documentar, projetar, e testar APIs com padrão de comunicação baseado no framework REST Barbosa [2023].

A aplicação do Swagger traz uma comunicação mais formatada dos módulos, com um conjunto de documentações mais claras, objetivas e com uma eficiência de 20% a mais no auxílio de tempo de desenvolvimento da API aplicada, em comparação com outros modelos Barbosa [2023].

O Swagger permite construir uma documentação bem estruturada, o que, como foi dito, reduz o tempo de desenvolvimento dos tratamentos das requisições e facilita a identificação e correções de erros Barbosa [2023].

A Figura III.2 apresenta a API desenvolvida sendo executada no ambiente local (localhost) na IDE MS Visual Studio 2022 com o padrão Swagger instalado.

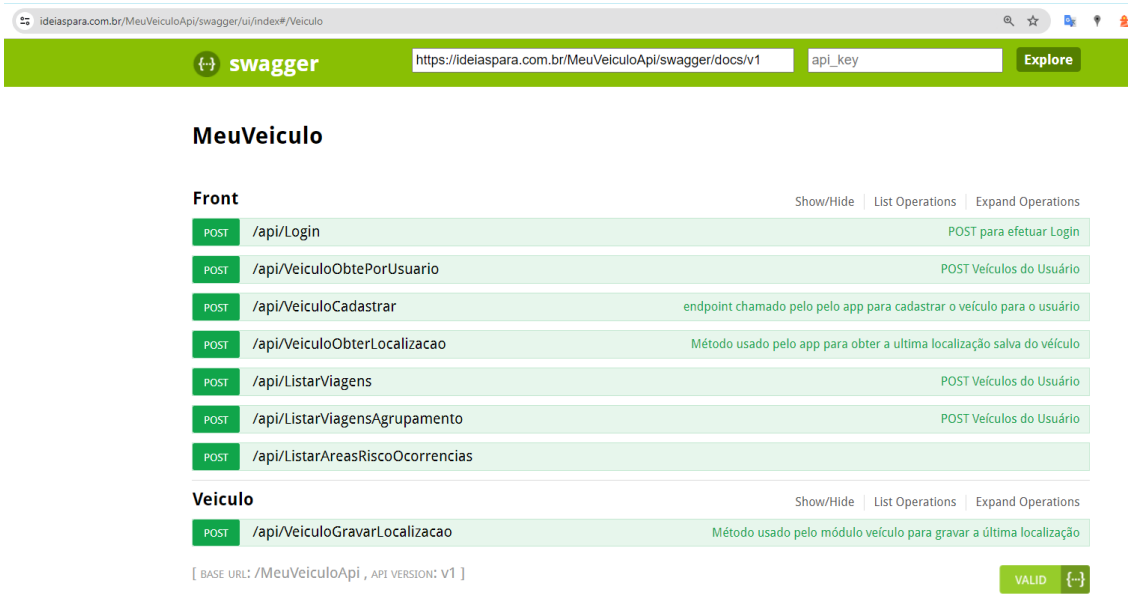


Figura III.2: Tela mostrando as funcionalidades da API, que está em execução em servidor local, no qual o Swagger está instalado, contendo os acessos de login do usuário, veículos por usuário, cadastro de veículo, localização, listagem de viagem, tratamento do agrupamento e gravação das ocorrências. Fonte: Autoria e obtenção própria

III.5.1.1 Algoritmo de Haversine

Em seu trabalhoGarey and Johnson [1979], classificam o problema de deslocar com o menor caminho, de um ponto "A"a um ponto "B", como problemas de NP- Completos, que são problemas de complexidade mais elevada em que suas soluções perpassam por tempo de execução polinomial. [Garey and Johnson, 1979].

A partir dessas conclusões de [Garey and Johnson, 1979], observou-se que essa aplicação poderia ser utilizada em problemas de localização geográfica, como é o caso do algoritmo da fórmula de Haversine. O Haversine é um método que calcula a distância entre dois pontos com base no comprimento de uma linha reta entre dois pontos na latitude e longitude Hizham and Ginardi [2021].

A equação de Haversine utilizada no trabalho vigente se baseia em uma das versões de implementação vista em Bala Dhanalakshmi et al. [2022] que indica que a solução algorítmica (do ponto de vista matemático) pode ser descrita em uma divisão em 3 equações (Equação III.1 , Equação III.2 , Equação III.3), conforme abaixo:

$$a = \sin^2(\frac{\Delta\phi}{2}) + \cos\phi 1 \cos\phi 2 \sin^2(\frac{\Delta\lambda}{2}) \qquad \text{(Equação III.1)}$$

$$c = 2atan2(\sqrt{a}, \sqrt{1-a}) \quad \text{(Equação III.2)}$$

$$d = Rc \quad \text{(Equação III.3)}$$

Onde:

$\phi 1$ = latitude da localização corrente

$\phi 2$ = latitude da localização de destino

$\Delta\phi$ = Diferença entre as Latitudes

$\Delta\lambda$ = Diferença entre as Longitudes

c = ângulo calulado aproximado

d = distância entre os dois pontos

Fórmula Haversine Fonte:[Bala Dhanalakshmi et al., 2022]

As equações acima, podem ser simplificadas , conforme equação abaixo:

$$d = 2r\sin^{-1}\left(\sqrt{\sin^2(\frac{\phi_2-\phi_1}{2}) + \cos\phi_{x_i}\cos\phi_{x_j}\sin^2(\frac{\lambda_2-\lambda_1}{2})}\right) \quad \text{(Equação III.4)}$$

Onde:

ϕ = latitudes

λ_1 = longitudes

r = raio da Terra

d = distância entre os dois pontos

Fórmula Haversine Fonte:[Nichat, 2013]

III.5.1.2 uma implementação da fórmula de Haversine no projeto:

No trabalho de Dauni et al. [2019], há um modelo de implementação da Fórmula de Haversine baseado em C#. Nesta implementação, usa-se o raio conhecido da linha do equador, que é 6371 km, calculando-se o radiano da diferencia entre longitude origem e destino. É um modelo para usar em cálculo de distâncias com GPS e que é adotado no trabalho vigente.

O exemplo do código implementado pode ser visto na algorítimo 1. Dauni et al. [2019].

```
1  public static double calc(double lat1, double lon1, double lat2, double lon2)
2   {
3           double Raio = 6371; //Raio da terra
4           double RadianaLat = (Math.PI / 180) * ((lat2 - lat1)); // Calcula a
   Radiana da diferencia entre latitude de origem e destino
```

```
5           double RadianaLon = (Math.PI / 180) * (lon2 - lon1); // Calcula a
    Radiana da diferencia entre  longitude origem e destino
6           double parcial = Math.Sin(RadianaLat / 2) * Math.Sin(RadianaLat / 2) +
      Math.Cos((Math.PI / 180) * (lat1)) * Math.Cos((Math.PI / 180) * (lat2)) *
    Math.Sin(RadianaLon / 2) * Math.Sin(RadianaLon / 2);
7           double AnR = 2 * Math.Asin(Math.Min(1, Math.Sqrt(parcial)));
8           double distancia = Raio * AnR;
9           return distancia;
10    }
11
```

Algorítimo 1: Exemplo de código em C# usado para cálculos de distâncias, segundo as fórmulas de Haversine.Fonte: Dauni et al. [2019].

III.5.1.3 Uso de instrumentos de banco de dados para cálculo das fórmulas de Haversine:

A fórmula de Haversine também pode implementada com código SQL de Banco de Dados (BD). Nestes casos, o BD é responsável por armazenar os dados coletados pelos dispositivos e posteriormente aplica-se um código SQL para mapear os dados a serem utilizados na fórmula. No trabalho de Dauni et al. [2019], há um exemplo de código SQL, que utilizou dados de um BD MySQL para cálculo de distâncias de Haversine, como pode ser observado no algorítimo 2.

Neste implementação usa-se o raio conhecido da linha do equador "6371", como o raio é a metade da circunferência, multiplica-se este, por 2 para usar toda a circunferência da terra, conforme abordado no trabalho de Dauni et al. [2019] .

```
1       CREATE PROCEDURE `distance`(IN `orig_lat` DOUBLE(12,10),IN `orig_lon`
    DOUBLE(12,10),IN `dist` FLOAT(4))
2       BEGIN
3         SELECT stores.*,
4         6371*2*ASIN(SQRT(POWER(SIN((orig_lat -
5         stores.latitude)*pi()/180/2),2)+COS(or
6         ig_lat *pi()/180)*COS(stores.latitude
7         *pi()/180)*POWER(SIN((orig_lon - stores.
8         longitude)*pi()/180/2),2)))as distance,
9         FROM stores
10         HAVING distance <= dist
11        ORDERBY distance ASC;
12      END
13
```

Algoritimo 2: Fórmulas de Haversine implementadas com código SQL, utilizando os dados contidos em um BD MySQL. Fonte: [Verka Jovanović, 2015].

III.5.2 Banco de dados

O componente do Banco de Dados (BD) é aquele que armazena as principais informações tratadas pela plataforma, isso inclui os dados do veículo, ocorrências, dados do usuário, permissões e quaisquer outras identificadas para melhor aproveitamento da plataforma

Esses dados são os que se utilizam nos cálculos vistos no componente API. A representação, em forma de um tipo de diagrama de classes, com o mapeamento de todos os dados utilizados pela plataforma nos cálculos usando a Fórmula Base de Haversine (Equação 2.4), baseados no trabalho de [Verka Jovanović, 2015], podem ser observados na figura III.3

III.5.3 Modelagem do banco de dados

Abaixo, apresentação uma implementação do modelo entidade relacionamento

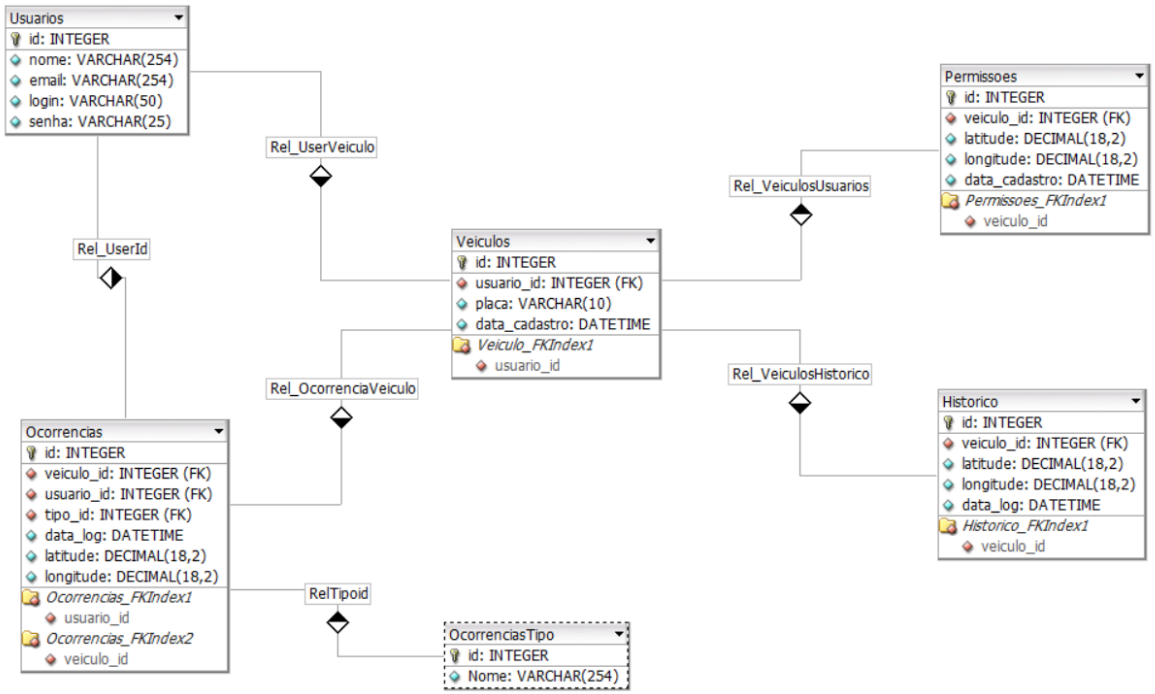


Figura III.3: Modelo Entidade Relacionamento(MER). (mostrando as tabelas e seus relacionamentos) que contém os dados usados para cálculos da Fórmula de Haversine, tais como, usuários, veículos, ocorrências, permissões, histórico e tipo de ocorrência. Fonte: Elaboração Própria

Neste modelo de entidade relacionamento, mer, a tabela Veículo é a principal, que congrega as informações das outras classes, de modo a estabelecer relacionamentos entre elas, visando conciliar as configurações das classes Usuários e Permissões, assim como os reportes de Ocorrências sinalizadas por eles. [Bazzi, 2018]. A tabela III.1 apresenta o significado de cada uma das classes, vistas como Tabelas do BD utilizado (no caso o MySQL).

Tabela III.1: Descrição das entidades vistas na Figura III.3.

Tabela	Função
Veiculos	Descrição
Usuarios	Cadastro de Usuários
Ocorrencias	Armazernar as ocorrências(Roubo, furto de veículos do usuários)
Historico	Histórico de ocorrências
Permissoes	Cadastro de veículos por usuário
OcorrenciasTipo	Possíveis tipos de ocorrências. Ex: Roubo, furto.

Na tabela/entidade **??** são vistos a descrição de todos os campos de cada classe/tabela utilizado no esquema conceitual visto na Figura III.3.

Tabela III.2: Descrição das entidades do componente Banco de Dados

Entidade veiculo		
Coluna	**Tipo de dado**	**Descrição**
id	Int	Identificador único de Veículo
usuario_id	Int	Identificador do usuário
placa	Varchar(20)	Placa do veículo
data_cadastro	Datetime	Data de cadastro do veículo
latitude	DECIMAL(18,2)	Última latitude conhecida do veículo
longitude	DECIMAL(18,2)	Última longitude conhecida do veículo
data_atualizacao	Datetime	Última data de atualização do sistema
veiculo_codigo	Varchar(254)	identificador único no sistema
veiculo_mensagem_io	Varchar(254)	Contém: latitude longitude, velocidade, satélites e altitude,. no seguinte formato: Latitude:-22.839722-Longitude:-42.133881-Velocidade:6.46-Satelites:8-Altitude:14.10
veiculo_chave_viagem	Varchar(254)	Identifica a última viagem iniciada
veiculo_distancia_ percorrida	DECIMAL(18,6)	distância percorrida na ultima viagem
veiculo_distancia_permitir_lugares_visitados	DECIMAL(18,6)	Local que o usuário configura a distância máxima permitida do veículo a lugares já visitados
veiculo_distancia_permitir_lugares_passagens	DECIMAL(18,6)	Local que o usuário configura a distância máxima permitida do veículo a lugares que já passou
veiculo_solicitar_ bloqueio	TINYINT(4)	Flag que para bloquear direto, caso posição atual seja fora do padrão, ou solicitar bloqueio a terceiros.
Entidade Historico		
Coluna	**Tipo de dado**	**Descrição**
id	Int	Identificar único de um histórico
veiculo_id	Int	Identificador do veículo
data_log	Datetime	data de gravação do histórico
latitude	DECIMAL(18,2)	latitude conhecida do veículo
longitude	DECIMAL(18,2)	longitude conhecida do veículo
veiculo_mensagem_io	Varchar(254)	Contem: latitude, longitude, velocidade, satélites e altitude; no seguinte formato: Latitude:-22.839722-Longitude:-42.133881-Velocidade:6.46-Satelites:8-Altitude:14.10
velocidade	DECIMAL(18,2)	Velocidade do veículo na linha específica
veiculo_chave_viagem	Varchar(254)	Identifica e agrupa as linhas de histórico e torno da viagem
veiculo_distancia_ percorrida	DECIMAL(18,6)	distância percorrida na última viagem

Tabela III.3: Descrição das entidades do componente Banco de Dados

Entidade Ocorrencia		
Coluna	**Tipo de dado**	**Descrição**
id	Int	Identificador único de uma ocorrência
veiculo_id	Int	Identificador do veículo
usuario_id	int	Identificador do usuário
tipo_id	Int	Tipo de ocorrência, que pode ser: Bloqueio por usuário , bloqueio por IA ou desbloqueio por Usuário
data_log	Datetime	data do evento
latitude	DECIMAL(18,2)	latitude conhecida do veículo
longitude	DECIMAL(18,2)	longitude conhecida do veículo
data_executado	Datetime	Data de execução, do bloqueio, isto é, a data e hora que o comando foi recebido e acatado pelo veículo
mensagem	Varchar(254)	Campo onde a solução grava o mensagem do motivo da ocorrência
Entidade Usuarios		
Coluna	**Tipo de dado**	**Descrição**
id	Int	Identificação única do Usuário
nome	Varchar(254)	nome do usuário
email	Varchar(254)	email do usuário
login	Varchar(50)	login do usuário
senha	Varchar(50)	Senha do usuário
Entidade Permissoes		
Coluna	**Tipo de dado**	**Descrição**
id	Int	Identificação única de uma permissão
veiculo_id	Int	Identificador do veículo
data_cadastro	Datetime	data de gravação da permissão
latitude	DECIMAL(18,2)	latitude permitida para o veículo
longitude	DECIMAL(18,2)	longitude permitida para o veículo
Entidade Ocorrencias_Tipo		
Coluna	**Tipo de dado**	**Descrição**
id	Int	Identificação única do tipo de ocorrências
Nome	Varchar(254)	Nome do tipo de ocorrência

III.5.3.1 Métodos das classes contidas no BD

Além das entidades descritas na tabela III.1, a solução ainda conta com os métodos a elas associados, que, no BD, são vistas como como procedures e views, que contém o código no BD que implementam esses métodos. Esse modelo de configuração, com os métodos e classe no mesmo BD, existe pelo fato da solução IoT necessitar de execuções das funções com respostas em tempo real.

Isso existe porque a solução necessita de alto desempenho, e, deste modo, esses métodos vão ser executados mais rapidamente, reaproveitando facilidades de cache lógico e físico do microcontrolador (a CPU do ESP32 permite caches similares a L1, L2 e l3 dos grandes processadores) [Larsson,2023]. A Tabela III.4 apresenta parte desses métodos, que também são controlados dentro

do BD utilizado para a solução.

Tabela III.4: Apresentação de alguns dos métodos contidos no BD do Back-end

Nome do Método	Parâmetros	Descrição
GetLogin	p_login varchar(254) , p_email varchar(254), p_senha varchar(254)	Procedimento que a API chama para efetuar o login
GetVeiculo	p_usuario_id int , p_veiculo_id int	Procedimento que api chama para obter os dados de um veículo específico
GetVeiculoLocalizacao	p_usuario_id int , p_veiculo_id int	Procedimento chamada via api para mostrar a última localização conhecida do veículo.
GetVeiculoOcorrencias	p_usuario_id int , p_veiculo_id int	Procedimento chamada via api pelo módulo veículo para listar as ocorrências de bloquei e desbloqueio dos veículos
GetVeiculosPorUsuario	p_usuario_id int	Procedimento que lista os veículos associados aos usuários
ListViagens	p_usuario_id int , p_data_inicio DATE, p_data_fim DATE	Controle da Lista as viagens agrupadas pelas chaves para determinado usuário, dados um período inicial e final
ListViagensAgrupamento	p_usuario_id int , p_data_inicio DATE, p_data_fim DATE, p_agrupamento VARCHAR(254)	Procedimento que lista as viagens agrupadas dado o parâmetro p_agrupamento pelo tipo de agrupamento: dia, mes, hora_dia_semana , hora_dia e dia_semana para determinado usuário, dados um período inicial e final
SetVeiculoLocalizacao	p_usuario_id int , p_veiculo_id int , p_latidude varchar(254) , p_longitude varchar(254) , p_veiculo_mensagem_io varchar(254) , p_chave_viagem varchar(254)	Procedimento que grava a localização do veículo na tabela de ocorrências
VeiculoCadastrar	p_usuario_id int , p_placa varchar(254)	Procedimento para cadastrar o veículo e associar ao usuário
vw_viagens	-	Apresenta a lista das viagens agrupadas pelas chaves de viagens

Um exemplo a ser visto é o do método vw_viagens (que vai apresentar uma view como resposta), que é utilizado como entrada para um outro método (que é um procedure), o ListViagensAgrupamento, esse usado para mostrar a consulta da situação da lista de viagens, ou de caminhos

percorridos.

Esses dois métodos juntos servem para agrupar todos os dados dos percursos, como, por exemplo, distância percorrida entre origem e destino, velocidade média, latitude e longitude da origem e destino, data/hora de início e término do percurso e o tempo total de caminho percorrido. Esse conjunto de métodos realiza o cálculo com base no parâmetro de z-score Leão et al. [2020] da linha de viagem. No Anexo F, pode-se verificar o código completo do método vw_viagens.

O método ListViagensAgrupamento, é o que vai gerar os gráficos que identificam os caminhos percorridos e que trabalham como interface final da camada de front-end (o aplicativo Web) da parte de software da plataforma No Anexo E, pode-se observar o código fonte especifico desse método. Entre alguns parâmetros examinados por esse método (que no BD é uma procedure), estão a data inicial da amostragem, a data final da amostragem e o tipo de agrupamento para ser apresentado em formato gráfico, isso para cada usuário

O uso agrupado desses dois métodos também vai mostrar a situação de cada percurso, de modo que seja possível rastrear os dias da semana /período do dia, para que se possa identificar os padrões típicos de uso associados ao veículo. Como essa técnica, pretende-se responder a perguntas, tais como: "Em determinado dia da semana, em determinado horário qual distância/quilometragem percorrida é a padrão para esse veículo do usuário X?". A resposta pode ajudar a observar desvios desse padrão e ativar, ou não, bloqueios do veículo.

O z-score, segundo Irúrtia and Iglesias [2009] , pode ser calculado segundo a Equação III.5 , que retorna com quantos desvios padrões o objeto se encontra dentro do domínio examinado.

$$Z = \frac{X_i - \mu}{\sigma} \qquad \text{(Equação III.5)}$$

No qual: Xi = valor da variável (objeto examinado); = média populacional (domínio do objeto) e = desvio padrão resultante. Neste trabalho, considerou-se Xi como a distância percorrida; u como a distância média percorrida e como o desvio padrão obtido através do método "STDDEV" (que é uma função do BD MySQL).

Outro componente importante na identificação de padrões são os agrupamentos, que, segundo Escalante et al. [2014] , são ferramentas para identificar outliers, que são valores discrepantes ou inconsistentes em relação ao conjunto de dados analisados Escalante et al. [2014].

Com o método da procedure ListViagensAgrupamento é possível formar agrupamentos, de modo a tornar possível identificar padrões inconsistentes e usar a Machine Learning da API para inferir sobre os erros e exibir gráficos mais próximos do percurso real, com uma variação negativa de até 5 graus, de modo que, no Front-end, apareça o percurso e seus dados mais próximo possível do real.

Lembrando que a precisão de GPSs público podem ter erros de até 20% de posicionamento e, por isso, a correção com o método ListViagensAgrupamento é necessária para reduzir qualquer

imprecisão no caso de marcação de posicionamento. A Tabela III.5 apresenta outros principais agrupamentos tratados pelo método/procedure ListViagensAgrupamento.

Os agrupamentos da Tabela III.5 visam possibilitar as apurações de distâncias percorridas, por dia, por mês, por dia da semana, por período do dia, ou por período do dia da semana , e gerar o quantitativo de viagens foram feitas para cada período examinado (dia, período, hora, semana ou mês).

Tabela III.5: Apresentação de agrupamentos tratados no método/procedure ListViagensAgrupamento

Nome do Agrupamento	Coluna Agrupada	Função
hora_dia_semana	dia_semana, periodo	Surge de forma virtual a partir da extração do dia da semana e do período do dia (manhã , tarde e noite). Visa agrupar por dia da semana/período do dia que foram realizadas as viagens e o quantitativo de viagens nesses períodos.
hora_dia	periodo	Surge de forma virtual a partir da extração do período do dia (manhã, tarde e noite). Visa agrupar por período do dia em que foram realizadas as viagens, com o quantitativo de viagens nesses períodos.
dia_semana	dia_semana	Surge de forma virtual a partir da extração do dia da semana da viagem. Visa agrupar por dia da semana que foram realizadas as viagens. além de fornecer o quantitativo de viagens nesses períodos.
dia	data_viagem	Agrupa-se pelos dias que foram realizadas as viagens.
mes	data_viagem	Agrupa-se pelos dias/meses que foram realizadas a viagens.

III.5.4 Módulo Veículo

Este módulo possui os componentes de hardware e realiza a coleta das informações lidas pelos sensores acoplados ao veículo (GPS etc) enviando esses dados para a API controladora, que, por sua vez, vai preencher os dados do BD com essas informações e ativar os métodos para exibição de resultados na camada Front-end. Entre os componentes de hardware, existe o microcontrolador, o módulo GSM, o módulo GPS o módulo OBDII, os módulos reles e os módulos conversor de 12 VDC para 5 VDC.

III.5.4.1 Eletrônica e hardware do Módulo Veículo

A arquitetura utilizada pela plataforma pode ser vista no esquema eletrônico da Figura III.4

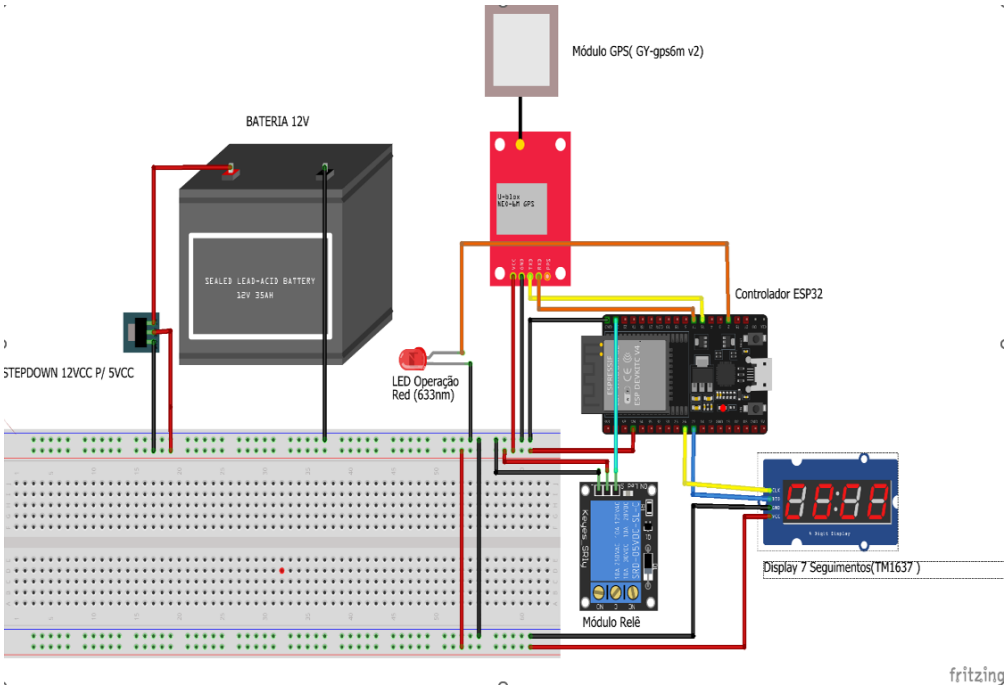


Figura III.4: Esquema eletroeletrônico do hardware do Módulo Veículo, contendo a conexão com aa bateria de 12v, GPS, a interface do microcontrolador, relé e display . Fonte: Elaboração Própria

A Figura III.5 apresenta as ligações eletroeletrônicas do mecanismo de hardware da plataforma, apresentando na Figura III.4.

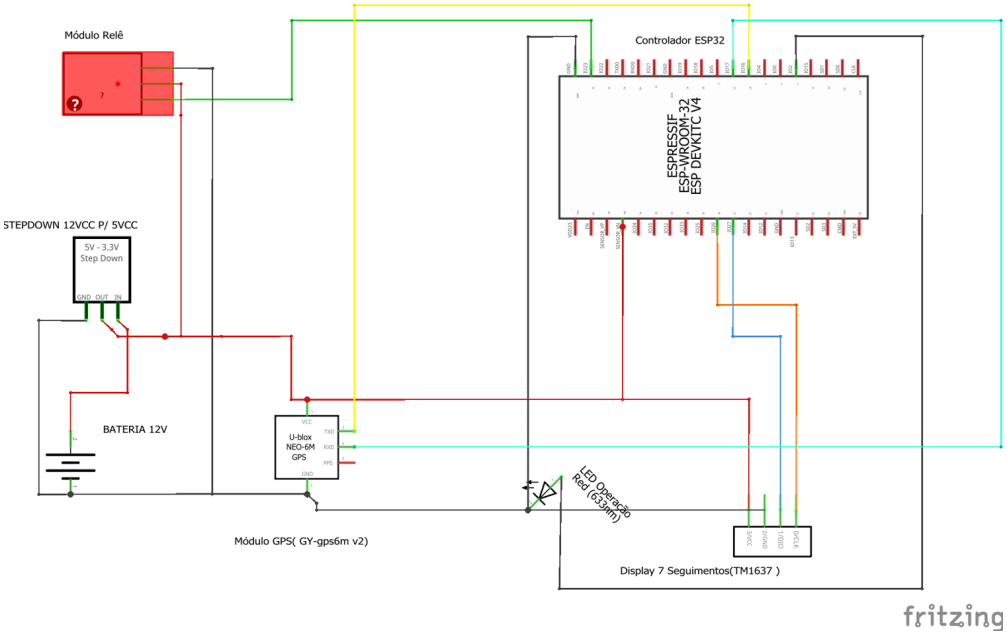


Figura III.5: Representação das ligações eletrônicas entre os componentes, mostrando as pinagens utilizadas entre eles. Fonte: Elaboração Própria

Cada componente do hardware é descrito a seguir, com seu objetivo, motivação de escolha, características associadas ao projeto, com seus propósitos e forma de manipulação de dados.

III.5.4.2 Microcontrolador

Como foi escrito antes, um pre-prototipo com Arduino Nano foi concebido, para testar aspectos como, suportar tensões de 3 V aplicada à plataforma, capacidade de sincronismo com frequências de 16 MHz e outras características.

Uma vez observado que um microcontrolador com potência de 1,4W pode ser usado de como controladora, percebeu-se também a necessidade de um dispositivo para responsável pela coleta de dados e transição via WI-Fi, para posterior envio ao API,

Como isso implicaria em obter um dispositivo extra acoplado ao Arduino Nano, elevando o custo e introduzindo um dispositivo com cerca de 1,5 W de consumo a mais, optou-se por uma interface mais robusta, com Wi-Fi já integrado, com um consumo abaixo dos 2 W, com frequência de clock entre 1 e 2 GHz (bem mais rápido do que o Arduino Nano), que vai também atuar como chave de ignição, inabilitando o veículo.

Devido a isso optou-se, como controlador principal, pelo dispositivo conhecido como ESP32 (Figura 10). Esse dispositivo possui como características básicas ser um módulo com Wifi e Bluetooth ESP-WROOM-32 embutido, baseado no chip ESP32-D0WDQ6. Ele conta com um microprocessador Xtensa 32-Bits LX6, que apresenta baixo consumo de energia (menos de 2 W), possui dois núcleos de CPU (processador dual core), permitindo aplicação em conjunto com sensores, módulos e outros tipos de interfaces. Ele pode ter o firmware atualizado, mantendo dentro dos padrões tecnológicos exigidos.

Esse módulo também possui memória SRAM de 520Kb; memória Flash externa de 32-Bit de acesso e 4Mb, trabalha com tensão de alimentação entre 2,7 à 3,6 V; está de acordo com o padrão WiFi 802.11 b/g/n; com segurança WiFi padrão WPA, WPA2, WPA2-Enterprise ou WPS; e Bluetooth 4.2 BR / EDR e BLE. Tem controles para interface tipo UART, SPI, SDIO, I2C, I2S, IR, PWM; suportando temperaturas entre -40° à +85° C.

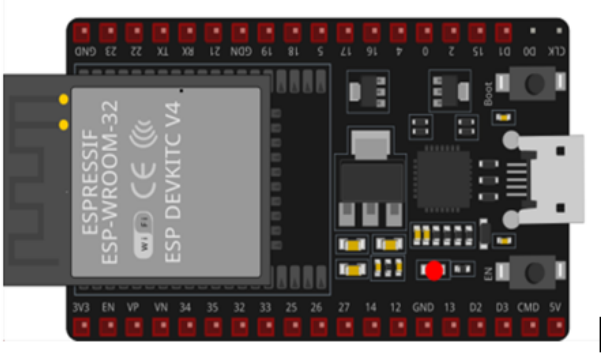


Figura III.6: Módulo Esp32 básico, com a pinagem usada e demais componentes acopladas ao chip-Fonte: Biblioteca do autor

O ESP32 já possui um módulo GSM embarcado também (o Arduino Ano não tem esse módulo).

Esse GSM é usado para fazer o CPU do ESP32 se comunicar com o aplicativo em nuvem, via API, e que permite o acesso do ESP32 à internet via uma operadora de telefonia. Seu funcionamento é o de um modem 4G. Para isso, é importante observar as frequências das redes 4G brasileiras, de modo a adquirir o chip de celular apropriado através de uma das operadoras credenciadas pela Anatel.

Este aspecto basicamente reflete no custo fixo para os futuros usuários da plataforma desenvolvida, sendo dependente dos custos junto a operadora escolhida e seus planos de telefonia Cruz et al. [2018].

III.5.4.3 Módulo Regulador de Tensão Step/Down 12V DC para 5V DC

Uma vez que, nos veículos automotores encontra-se uma tensão de 12 V em corrente contínua (DC), e que o ESP32 e seus módulos são alimentados por uma tensão entre 2,6 e 5 V DC, usa-se esse módulo como transformador de tensão, para converter uma tensão entre 12 e 20 V para 3 a 5 V, segundo visto em Monk [2014].

III.5.4.4 Módulo Rele de 2 Canais

Este módulo atua como chaveamento, recebendo sincronismos do ESP32 através de dos pinos GPIO (0 ou 1), para atuar como chave de liga e desliga do motor, caso o ESP32 envie esse sinal para o motor do veículo. A sugestão do módulo de 2 canais é para possibilitar chavear dois componentes simultaneamente.

III.5.4.5 Módulo GPS

Este módulo basicamente recebe um sinal de satélite e com isso, obtêm-se a posição do componente no globo (Latitude e Longitude). Cruz et al. [2018]. . Sua função no circuito é de transmitir, em certo intervalo de tempo, a latitude e a longitude, cabendo ao controlador, comparar esses dados com os dados fornecidos pela nuvem, para então atuar interrompendo o funcionamento do veículo ou não. Optou-se, no projeto, pelo uso dos módulos da família NEO-xM. Os módulos dessa família são acessíveis, de baixo custo, operam com tensões de 2,7 a 5V DC, trabalham com comunicação serial, com conector UFL (antena), trabalham com temperaturas entre -40 a 85° Celsius e tem uma precisão de trabalho de 5m. Um exemplo dele pode ser visto na Figura III.7.

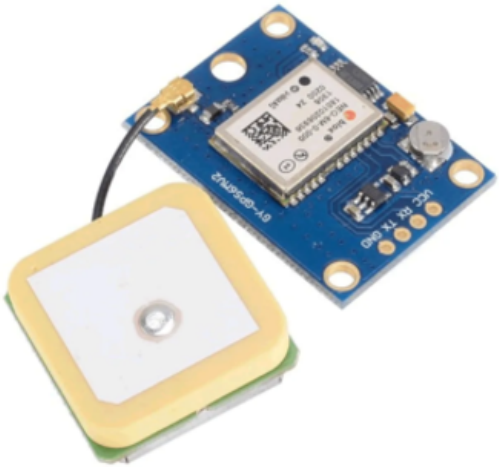


Figura III.7: Módulo GPS da família NEO-xM, do tipo GY-NEO6MV2.
Fonte: https://www.masterwalkershop.com.br/modulo-gps-gy-neo6mv2-com-antena

III.6 Detalhamento da Camada Front-end

Esta camada é a camada na qual o usuário interage, insere os dados e suas preferência das rotas, veículos etc; e acessa as funcionalidades da solução, esta camada é composta pelo Aplicativo Web, que será detalhada abaixo.

III.6.1 Aplicativo Web

O Aplicativo Web, como já posto, é a camada de interface do usuário, construído em HTML5, CSS3, jquery e Boostrap4, que é um framework de código aberto.

Como o foco deste trabalho é oferecer o maior impacto social possível, destaca-se que usar componentes de código aberto é um fator importante nesse contexto, uma vez que viabiliza a característica "baixo custo"desta plataforma.

Diante disto, todos os mapas constantes na solução, foram construídos com a solução leafletjs , que é uma solução de código aberto. Conforme visto no trabalho de Donohue et al. [2020], neste, também foi implementado com o framework jquery.

Entre telas da sua interface há a da Figura III.8, que mostra o Dashboard (painel de informações) do cliente, com os dados de acompanhamento de suas viagens, que são vistos em dois tipos de gráficos, sendo que o primeiro, mostra a comparação das distâncias percorridas nas viagens diárias, em um período de até 60 dias.

Neste gráfico exibe-se o dia da semana correspondente ao dia do mês, para facilitar a identificação do padrão de distância percorrida por dia da semana. O segundo gráfico, mostra o comparativo das distâncias percorridas mensalmente pelo usuário tendo como filtro o período de até 300 dias atrás.

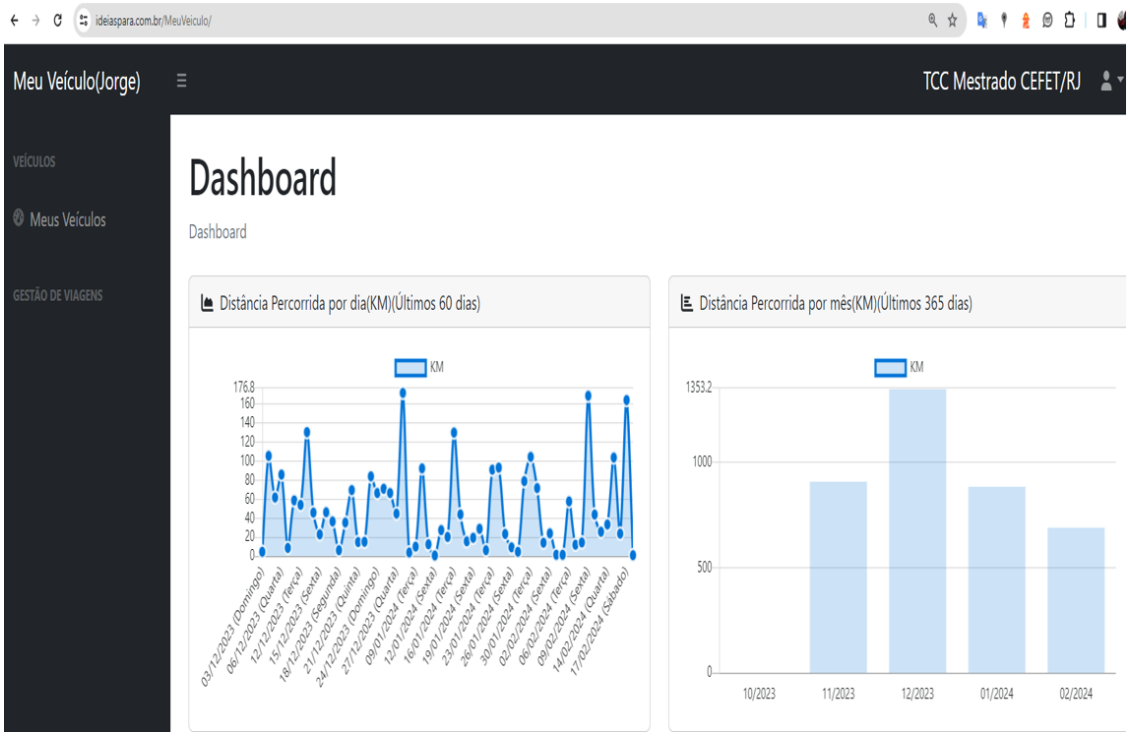


Figura III.8: Dashboard do usuário com gráficos, sendo que o primeiro, mostra a comparação das distâncias percorridas nas viagens diárias e comparativo das distâncias percorridas mensalmente s. Fonte: Biblioteca de imagens do autor

Ainda na tela inicial, há possibilidade de o usuário poder acompanhar a listagem de deslocamentos efetuados, ou viagens, como são chamadas neste trabalho, conforme visto na figura III.9.

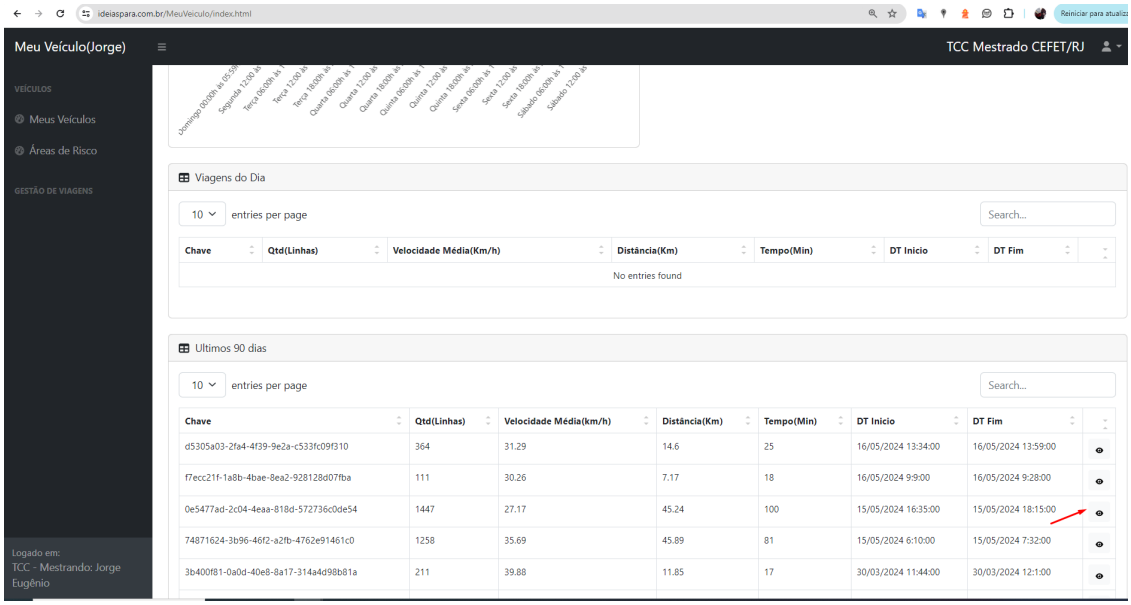


Figura III.9: Listagem de viagens do efetuados Fonte: Biblioteca de imagens do autor

A figura III.9 mostra essa listagens, onde há o botão de ver o detalhe de uma viagem específica num mapa, conforme exibido na figura III.10.

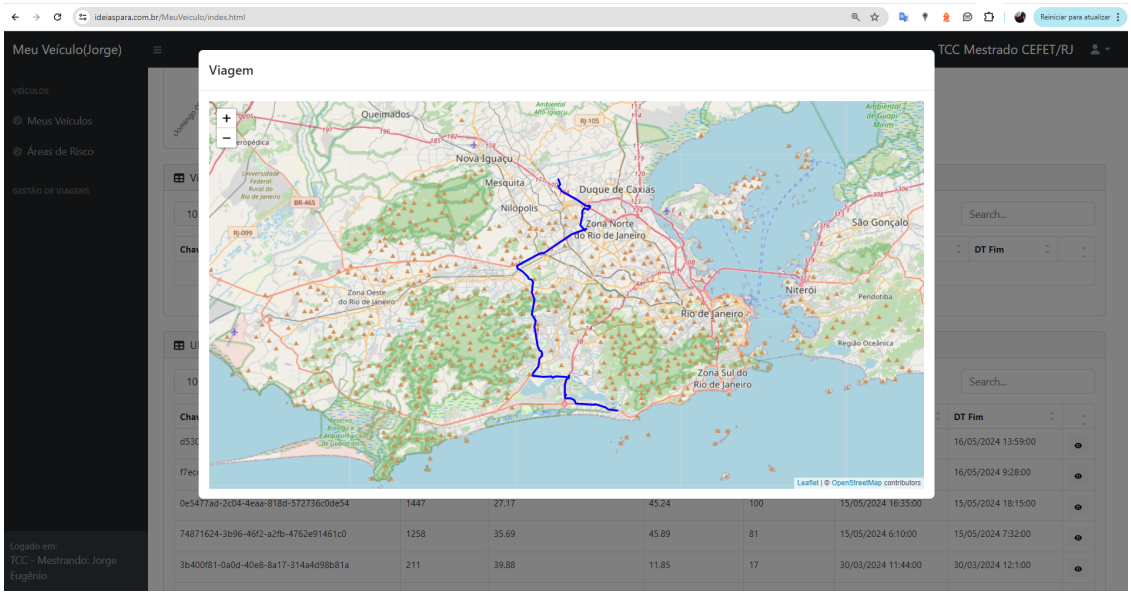


Figura III.10: Exemplo de uma viagem efetuada Fonte: Biblioteca de imagens do autor

A Figura III.11 mostra a opção “meus Veículos” da interface do Aplicativo Web. Utilizando a opção “Meus Veículos”, vai ser mostrado a listagem de todos os veículos relacionados ao usuário, com suas configurações e nos quais ele pode escolher para ver os dados de viagens de cada um.

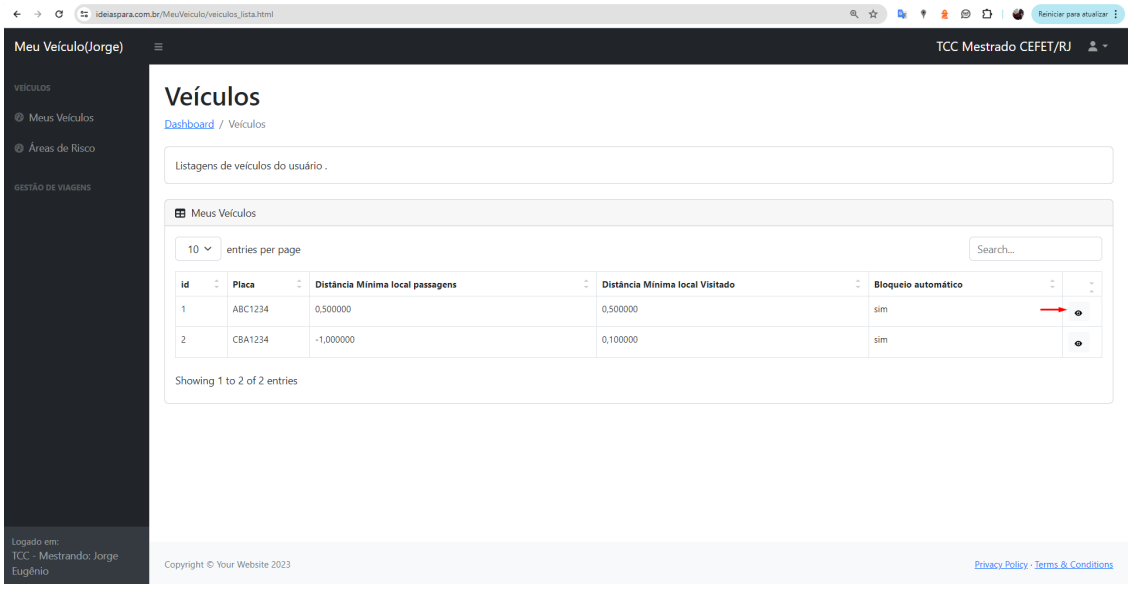


Figura III.11: Localização em tempo real do veículo

Vale ressaltar, que o ponto marcado na figura III.11, sinaliza o botão de navegação que mostra para o usuário a localização atual do veículo, como pode ser visto na figura III.12

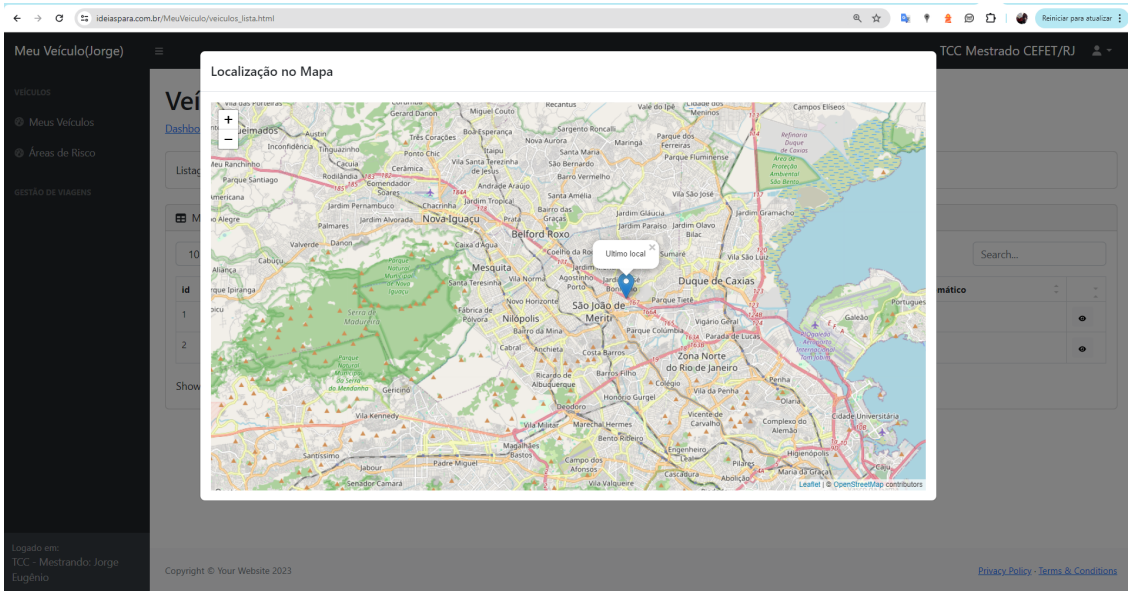


Figura III.12: Tela da interface com a visão da opção “Meus veículos” e o resultado com os veículos do usuário. Fonte: Biblioteca de imagens do autor

Outro requisito importante deste trabalho, é o compartilhamento de informações entre os usuários da plataforma, condição que trás o carácter colaborativo desta plataforma. Destaca-se que as informações compartilhadas são anônimas, conforme previsto na LGPD da República [2018] , sendo apenas os pontos onde ocorrem os bloqueios de veículos, de modo a gerar o mapeamento das áreas de risco.

A figura III.13 monstra um mapa com locais onde ocorreram bloqueios de veículos, e portanto, foram classificados como áreas de risco.Este mapa encontra-se acessível no menu ”Área de Risco”da aplicação.

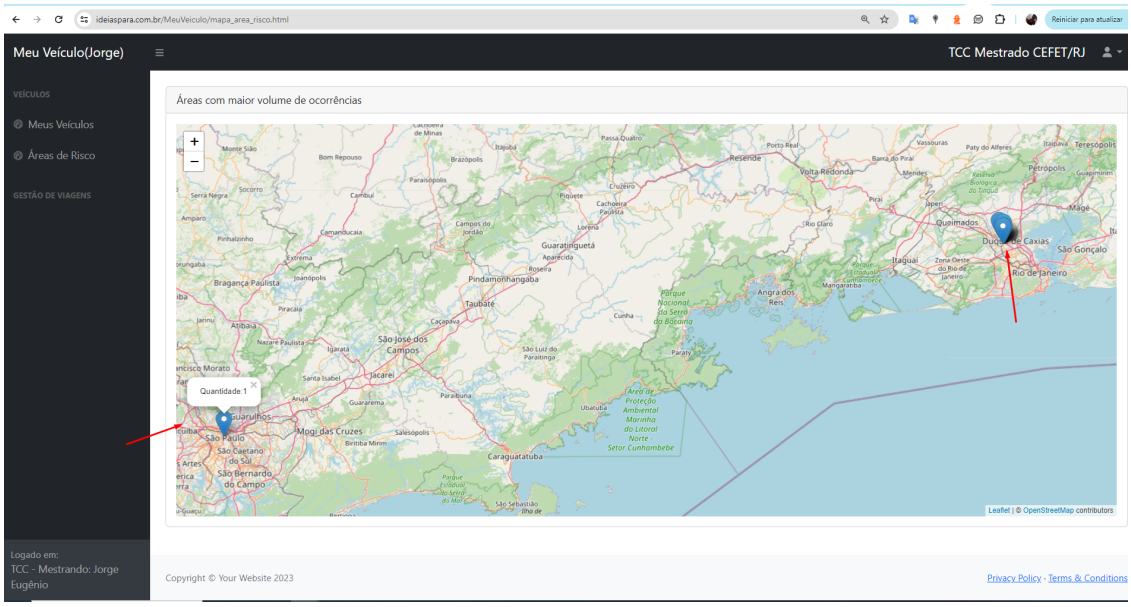


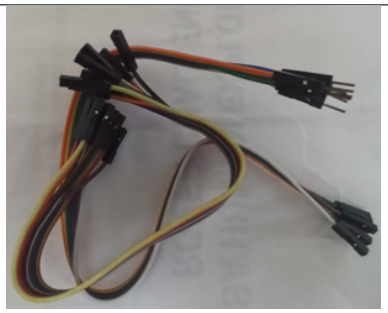
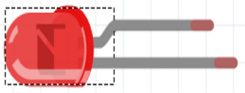
Figura III.13: Exemplo de locais que ocorreram bloqueios classificados como área de risco Fonte: Elaboração Própria

Capítulo IV Preparação do Protótipo

IV.1 Descrição básica do protótipo

O protótipo foi construído com base nos componentes vistos no item 2.4.1.c Além das componentes básicos vistos no Item indicado, foram acrescentados outros componentes, de modo a tornar o protótipo operacional de pleno e, com isso, poder utiliza-lo em campo. A Tabela IV.1 apresenta os demais componentes (além dos já vistos) utilizados na execução para o protótipo operacional.

Tabela IV.1: Tabela com os demais componentes utilizados na confecção do protótipo operacional utilizado nos testes de campo

Produto	Função no Circuito Protótipo	Imagens (da biblioteca de imagens do autor)
Módulo Display 7 Seguimentos (4 Dígitos TM1637)	Apresenta a velocidade real do veículo.	
Caixa para CFTV de tamanhos 8,3 x 8,3 x 5,5 cm	Caixa na qual está armazenada os componentes, representando o componente físico da plataforma.	
Kit de cabos Jumpers	Usado para as conexões eletroeletrônicas entre dispositivos.	
LED Alto Brilho 5mm vermelho	Usado para exibir o status de conectividade e inicialização do protótipo.	

De modo a se ter uma ideia dos custos alocados ao hardware e demais componentes físicos, a Tabela IV.2 apresenta os custos dos componentes necessários para a construção do protótipo. Os valores estão de acordo com o que foi encontrado no mercado entre janeiro de 2023 e dezembro de 2023.

Tabela IV.2: Especificação dos custos de cada componente de hardware ou físico relativco ao protótipo operacional. Fonte: Sites da Internet

Produto	Preço (em reais)
MiicrocontroladorEsp32	R$ 50,00
Módulo Display 7 Seguimentos (4 Dígitos TM1637)	R$ 15,00
Módulo Relé 5V 10A 2 Canais	R$ 15,00
Módulo GPS Gy-gps6m V2	R$ 40,00
Caixa para CFTV de tamanhos 8,3 x 8,3 x 5,5 cm	R$ 4,00
Kit de cabos Jumpers	R$ 12,00
LED Alto Brilho 5mm Branco	R$ 0,22
TOTAL	R$ 136,22

IV.2 Etapas da construção do protótipo da Plataforma

Nesse tópico são vistos as fases de implementação da Plataforma e suas integrações de módulos e componentes.

IV.2.1 Fluxos de integração dos módulos

Para os quatro principais módulos da solução, que são o Aplicativo Web, Hardware Veículo, API e BD, existem fluxos que descrevem suas integrações. Esses fluxos são apresentados nas Figura IV.1 e IV.2 .

Eles identificam as trocas de mensagens entre os módulos, de modo que a plataforma possa tratar a localização, a comunicação de bloqueios do veículo roubado/furtado e alterações de cadastro. A parte de bloqueio vai trabalhar com funcionalidades de Machine Learning, para que se possa classificar a não conformidade com os padrões comuns de uso do veículo por parte do usuário, na qual se poderá realizar os controles automatizados de bloqueio, ou informar ao usuário para que ele possa enviar comando à plataforma.

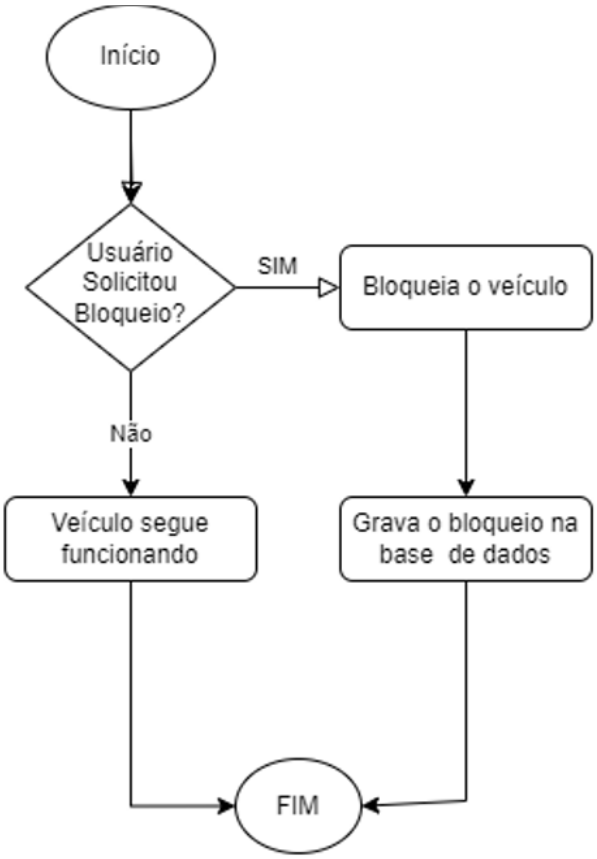


Figura IV.1: Fluxo de execução para casos de bloqueio manual do veiculo, no qual a operação é armazenada no módulo BD, servindo de base para aplicação do Machine Learning posterior Fonte: Elaboração pelo autor

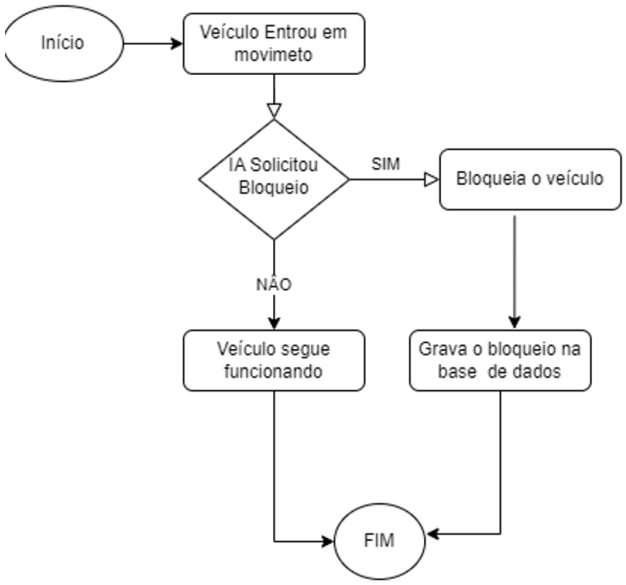


Figura IV.2: Fluxo complementar para controle de bloqueio, executado de forma autônoma, com acesso ao mecanismo de Machine Learning, para identificar se o local do veiculo está de acordo com o padrão classificado à ele como locais permitidos. Caso o local identificado não esteja classificado como permitido e o usuário não o confirmou alteração, o mecanismo de treinamento vai classificar o local como não permitido e enviará a solicitação de bloqueio- Fonte: Elaboração pelo autor

No fluxo das Figuras IV.2, são os quais é aplicado a fórmula de Haversine, que calcula a posição geográfica e as possíveis rota válidas, usando dados de latitude e longitude obtidas pelo módulo de hardware de GPS. Essa funcionalidade também é usada para o cálculo probabilístico da velocidade, de modo a acurar mais precisamente o sinal de bloqueio emitido pelo usuário.

Para esses cálculos, admitiu-se valores de z-score entre -2 e 2, ou seja, para cálculo das quilometragens de percursos nas viagens, usando desvio padrão. Deste modo é possível identificar os padrões de uso, sob a ótica de distância percorrida. Esses dados são utilizados pelo Machine Learning na funcionalidade de tomada de decisão de bloqueio.

IV.2.2 Funcionalidades do protótipo

As funcionalidades oferecidas pela Plataforma consistem em: cadastrar usuário, cadastrar veículos do usuário, identificar quando se iniciou um deslocamento de um veículo; iniciar a gravação das viagens na tabela histórico, para posterior agrupamento, tratamento e classificar os padrões de viagens.

Para tanto, o Módulo Veículo emite requisições ao Módulo API a cada 10 segundos, enviando dados da latitude, longitude, velocidade e distância percorrida desde a última requisição.

No mesmo momento em que envia a localização, o sistema questiona se tem ocorrências para bloquear, de modo que aproveite a mesma requisição para duas ações específicas.

Havendo notificação de bloqueio, essa é guardada na memória do módulo veículo, que por sua vez aguarda o sinal do veículo estar parado (num sinal, ou estacionamento), para, nesse instante, efetuar o bloqueio. Este bloqueio permanece até que o usuário envie a notificação de desbloqueio.

A Figura IV.3 apresenta a foto representativa do protótipo utilizado nos testes. Esse foi construído seguindo o esquema elétrico disposto na figura III.4. A caixa organizadora de CFTV possui as dimensões de 8,3 x 8,3 x 5,5 cm. A antena do módulo GPS fica ao lado, posicionada extremamente, visando uma melhor captação de sinal.

O display de sete seguimentos serve para exibir a velocidade apurada pelo módulo GPS, permitindo acompanhamento visual pelo usuário. O status do sinal de GPS pode também ser acompanhado pelo usuário através de um led simples, que possui a função de mostrar para o usuário o status da Plataforma, conforme o disposto na Tabela V.1, que segue abaixo.

Tabela IV.3: Sinais do LED e seus significados para a Plataforma

Funcionalidade	Função no circuito
LED Apagado	Indica: Plataforma não ativa
LED Piscando	Indica: Plataforma ligada, porém sem conexão ao servidor
LED Continuamente acesso (sem piscar)	Indica: Plataforma ativa, e conectada ao servidor.



Figura IV.3: Visão geral do protótipo construído. Fonte: Elaboração própria

Para acesso a interface mobile do Aplicativo Web, há uma a página inicial, o Dashboard, com o menu da opção “Meus Veículos”. Como já referenciado, o Dashboard mostra as viagens do usuário em dois gráficos, já descritos no item 2.4.2.b, na Figura III.8.

Com relação a API, a Figura IV.4 mostra a camada de apresentação dessa. A API, por essa tela, divide-se os Endpoints para o Armazenamento de Dados e Endpoints de Tomada de Decisão, respectivo ao módulo FrontEnd (o Aplicativo Web).



Figura IV.4: Tela Inicial da A‘PI, com acesso aos Endpoints por ela controlados, “consumidos” pelo Módulo Veículo e o módulo Aplicativo WEB. Fonte: Biblioteca de Imagens do Autor

Para garantir que somente pessoas ou dispositivos autorizado tenham acesso a consumir os

serviços da API, foi acrescentado, para os testes de campo, um token de acesso do tipo Bearer Neves and Domingues [2007] , disponível no Aplicativo Web e nas mensagens trocadas via dispositivo móvel. Um teste referente a verificação de autenticação pode ser visto na Figura IV.5 .

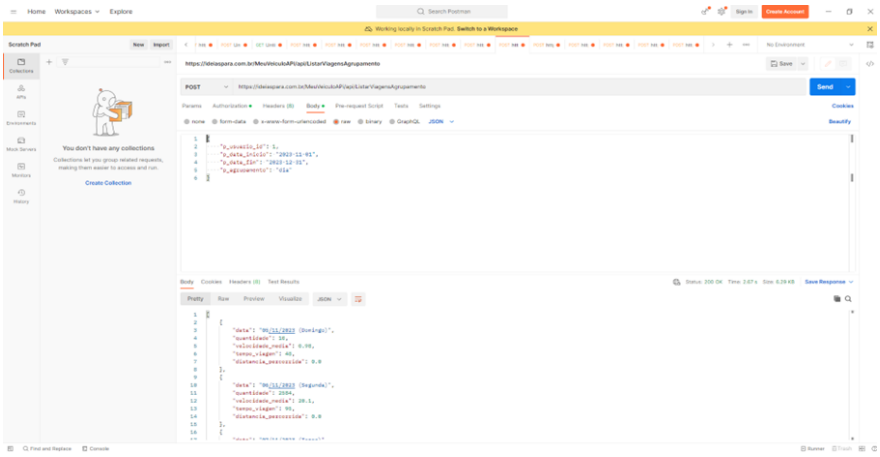


Figura IV.5: Apresentação do teste de requisição e autenticação de um Endpoint, usando o método postman. Fonte: Biblioteca de imagens do autor

Conforme especificado anteriormente, o módulo Veículo é o que obtém a localização do componente GPS e faz uma requisição post para a API mais precisamente o endpoint implementado pela procedure/método “VeiculoGravarLocalizacao”.

Este endpoint possui três responsabilidades: Gravar informações da viagem (Latitude, Longitude, velocidade e altitude), através da execução da procedure/método “SetVeiculoLocalizacao”, gravando na tabela Histórico do BD; Listar as ocorrências de bloqueios, através da execução da procedure/método “GetVeiculoOcorrencias”, que retorna as ocorrências que ainda não foram executadas pelo módulo API, de modo que seja possível ao usuário ou a Plataforma autonomamente, bloquear o veículo; e Prever se o veículo deve ou não ser bloqueado, executando a gravação dos dados da viagem, através da procedure/método “VeiculoGravarLocalizacao”, que verifica as variações do padrão de uso do veículo, classificando-as.

A plataforma utiliza, como solução para bloqueio automático, alguns indicadores, entre eles:

• Lugares já visitados (origem ou destino de viagens): Cujo bloqueio ocorre excluindo-se a viagem atual, como pode-se observar na implementação da procedure/método “VerificaBloqueio”.

• Lugares que já se passou durante uma viagem: cujo bloqueio ocorre excluindo-se a viagem atual, como pode-se observar na implementação da procedure/método “VerificaBloqueio” .

• Distância Percorrida por período do dia da semana (manhã, tarde ou Noite): que utiliza o “VerificaBloqueio”, com aspectos estatísticos, visando a previsibilidade de um veículo estar naquele local.

• Local de passagem do veículo: que verifica o destino, classificando-o como sendo local de risco ou não. Se utiliza das ocorrências de bloqueios que estão no BD, e valida o quão próximo o usuário

atual está das ocorrências sinalizadas com este tipo para classificar o local como sendo de risco ou não.

Todos os principais código inseridos nos diversos módulos da Plataforma se encontram nos anexos indicados, sendo que no Anexo A está o código embarcado no ESP32, em linguagem C; no Anexo B, o código fonte do endpoint "VeiculoGravarLocalizacao", desenvolvido com o C#; no Anexo C apresenta-se o código fonte da procedure/método "SetVeiculoLocalizacao"; no Anexo D apresenta-se o código fonte da procedure/método "VerificaBloqueio"; o Anexo E contém código fonte da procedure/método "ListViagensAgrupamento" em TSQL e o Anexo F contém o código fonte da view/método "vw_viagens" em TSQL.

IV.2.3 Custos de Hospedagem

Para que haja capacidade da Plataforma se comunicar via Wiofi e, com isso, otimizar o controle sobre os veículo do usuário, houve necessidade de se introduzir um custo relativo a hospedagem da página que contém a interface

API, em comunicação com o Aplicativo Web, disposto em um mobile qualquer do usuário A solução do protótipo da Plataforma precisa também estar hospedada em empresa de hospedagem de páginas web, localizada no Brasil. Foram levantados custos em empresas de hospedagem, para que se possa levantar o preço de hospedar, não somente os módulos que necessitem da

Internet (Aplicativo Web e API), assim como guardar em nuvem os dados do BD. A Figura IV.6 mostra um exemplo de custo mensal desses serviços, expresso nos planos de hospedagem da UOL Host.

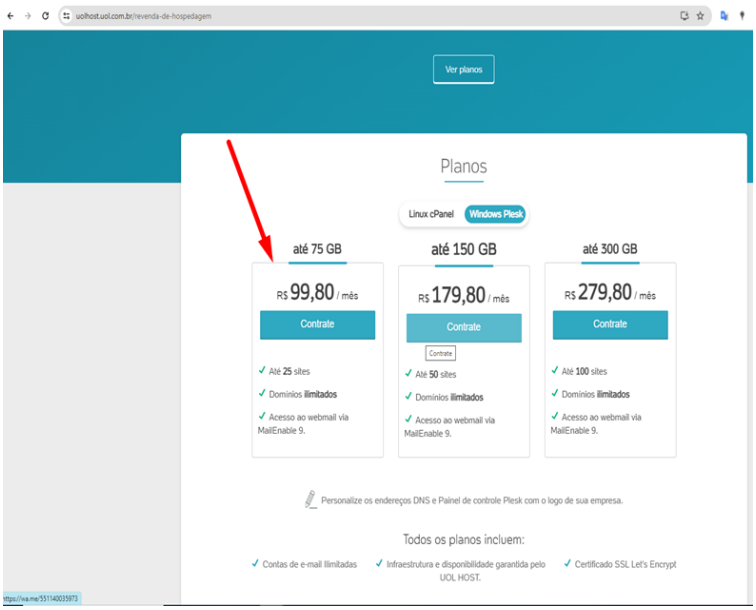


Figura IV.6: Imagem da página do https://uolhost.uol.com.br/revenda-de-hospedagem, apresentando o custo de assinatura mensal de um ambiente de hospedagem para a Plataforma.. Fonte: Biblioteca de imagens do autor

IV.2.4 Problemas encontrados e soluções propostas

Na confecção do protótipo completo da plataforma, encontraram-se alguns problemas, que necessitaram de um exame com maior acurácia a fim de que, ao colocar os testes em campo, pudessem ser solucionados e, com isso, fosse mais seguro a obtenção dos dados.

IV.2.4.1 Identificação da viagem

De acordo com a arquitetura, um dos problemas foi a identificação precisa de onde se iniciava e onde se concluía um trajeto. Diante deste problema, partiu-se do pressuposto que quando o usuário deseja iniciar um deslocamento, ele precisa ligar o veículo.

Neste momento o módulo Veículo solicita ao módulo API, a chave inicial de viagem, e então armazena na memória SRAM do ESP32. A partir daí, em todas as requisições se envia essa chave e todas as linhas de histórico são agrupadas usando-a como chave primária dos dados armazenados no BD, uma vez que a API vai também armazenar no BD cada item de viagem com a chave guardada na SRAM do ESP32. Esta chave é o campo "chave_viagem" da tabela Histórico, conforme descrito na Tabela III.3.

IV.2.4.2 Imprecisões do uso da fórmula Haversine para longas distâncias

Após usar a fórmula Haversine, dando como entrada a latitude e a longitude entre o início e o fim da viagem, observou-se que a mesma apresentava grande divergência da distância aferida do odômetro do veículo, ou ainda da distância medida por aplicativos de GPS.

A fórmula calcula a distância em percurso "retilíneo", porém na cidade, para chegar ao destino, percorre-se distâncias laterais ou ainda em direção oposta, até que finalmente se chegue ao destino.

Como solução para esse problema, utilizou-se o recurso de usar a referida fórmula, para calcular a distância entre o ponto atual e o de 10 segundos atrás, e ao final do percurso, soma-se todas essas pequenas distâncias percorridas e se obtém a distância percorrida no total com mais precisão. Após aplicação desse recurso, os valores de distância percorrida ficaram muito próximas do odômetro do veículo ou da medição de aplicativos de GPS.

A estratégia de calcular a distância entre pontos distintos separados pelo intervalo de tempo de 10 segundos é possível, visto que este, é intervalo de tempo em que o módulo veículo requisita o endpoint da API para gravar a localização. Os grafos das Figura IV.7 e Figura IV.8 , exemplificam melhor a solução abordada.

Nestes grafos, os vértices representam os pontos de início e destino do deslocamento, enquanto as arestas representam as distâncias entre os deslocamentos. Para tal, aborda-se dois cenários distintos para exemplificar o deslocamento entre as cidades hipotéticas "X e "Y"

No cenária A, como pode-se observar na Figura IV.7, calculando diretamente a distância percorrida data as latitudes e longitudes das cidades “X” e “Y”, obtém-se que a distância percorrida seria de 10 km. O que pode não refletir a realidade, como explicado anteriormente.

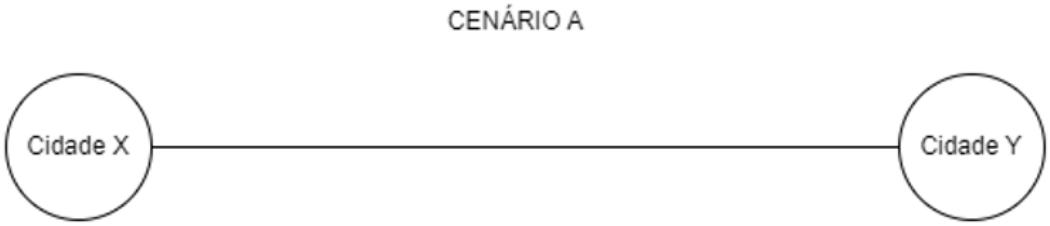


Figura IV.7: Cenário de cálculo de distância de forma retilínea, no qual os vértices representam os pontos de início e destino do deslocamento, enquanto as arestas representam as distâncias entre os deslocamentos. Fonte: Elaboração do autor.

Aplicando-se então a solução abordada, como pode ser visto na Figura IV.8, tem-se o deslocamento entre as cidades hipotéticas “X” e “Y”, constantes também do cenário A, nota-se que há diferentes distâncias percorridas entre os pontos de deslocamento, até ser atingido o destino final. Isto pode ser explicado, tendo em vista que o tempo é fixo (10 segundos), porém dentro desse tempo, pode ser percorrer mais ou menos distâncias. Atendo-se a isto, neste cenário, nota-se que a distância percorrida real entre as cidades “X” e “y”, são de na verdade 15 km.

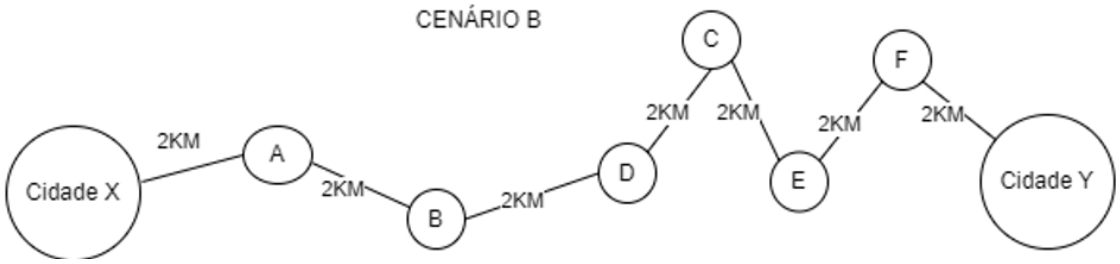


Figura IV.8: Cenário de cálculo de distância entre os pontos de deslocamento X e Y da IV.7, usando a solução de calcular sempre o ponto 10 segundos para trás. Fonte: Elaboração própria do autor.

Capítulo V Testes do protótipo: obtenção e análise de resultados

Para os testes a metodologia escolhida foi a de base qualitativa que, segundo Mattar and Ramos [2021] , caracteriza-se pela proximidade do pesquisador e o evento a ser observado ou ser pesquisado, no qual é facilitado a preparação do ambiente de testes. Também foi observado as etapas relativas aos testes, desde a montagem do ambiente, até na forma de captação dos dados.

V.1 Montagem de Ambiente de Teste

Foi adquirido, para os testes, um inversor de corrente, que é um dispositivo que converte a tensão do veículo 12V DC (continua) em tensão 127V AC (alternada), que é o tipo de corrente disponível na maioria das residências. Este dispositivo visa a ligação do módulo Veículo do protótipo com tomada de alimentação do veículo A Figura V.1 apresenta um exemplo deste tipo de dispositivo.



Figura V.1: Transformador/Inversor de corrente para adaptação a parte eletroeletrônica do veículo
Fonte: Biblioteca de imagens do autor .

Ainda para essa adaptação eletroeletrônica, foi adquirido também, uma fonte 5V/1A, com Pino P4, para ser ligada no inversor de corrente citado anteriormente, como pode ser visto na Figura V.2.



Figura V.2: Fonte 5V/1A com pino P4 para ser ligada ao inversor/transformador Fonte: Biblioteca de imagens do autor.

Posteriormente foi realizado o procedimento de instalação do módulo API no servidor escolhido para hospedagem. Para esse procedimento foi preciso compilar o código na IDE MS Visual Studio, e posteriormente enviar para o servidor através da ferramenta de FTP FileZilla. Esse processo pode ser visto através das Figura V.3 e Figura V.4 .

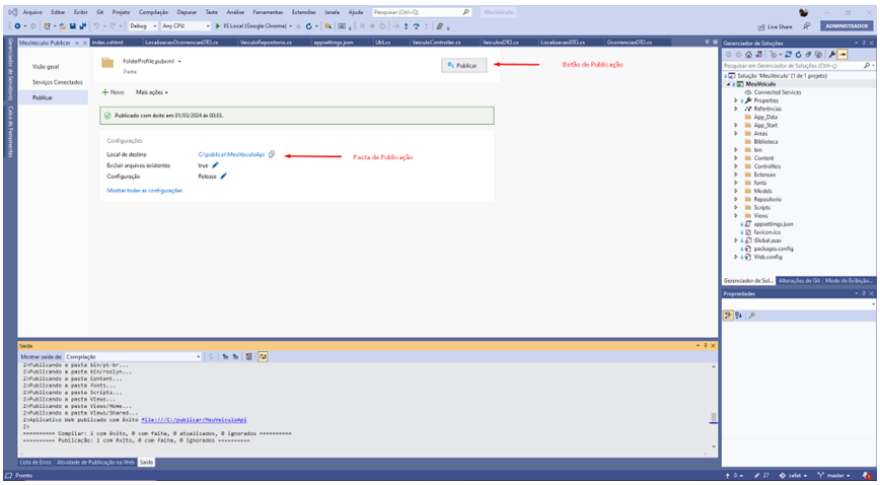


Figura V.3: Tela do MS Visual Studio que apresenta o momento do processo de compilação para posterior publicação do código no site de hospedagem. Fonte: Biblioteca de imagens do autor. .

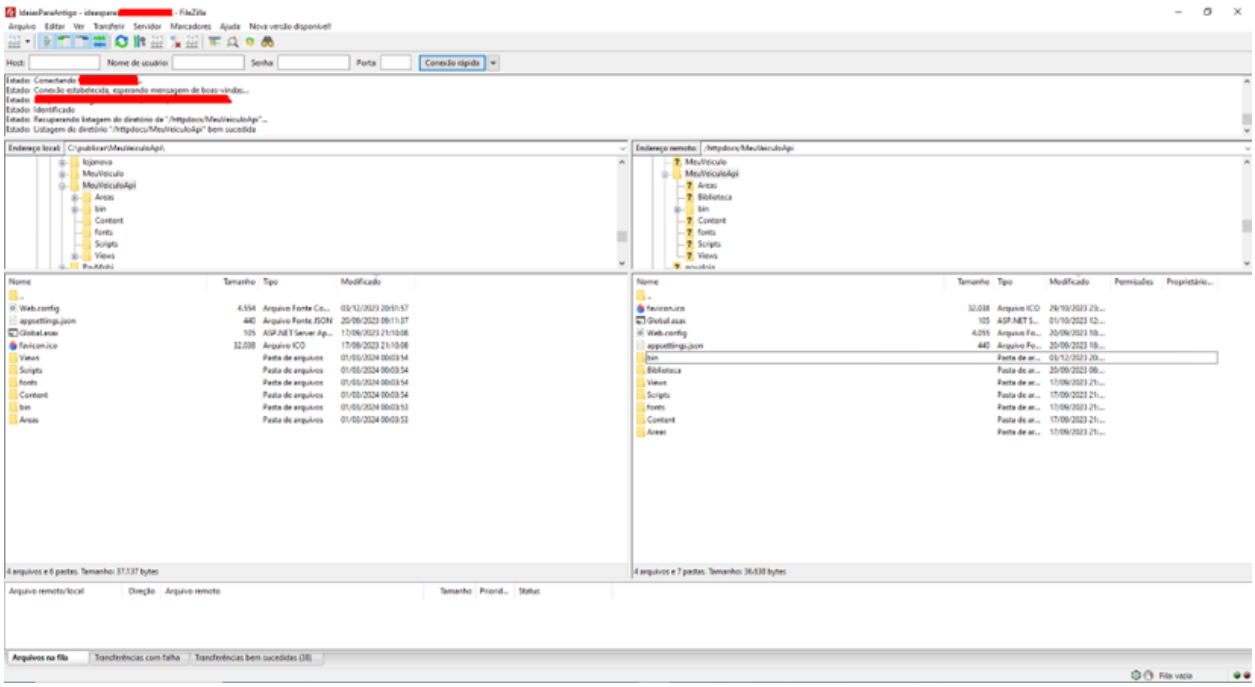


Figura V.4: Tela e para edição dos dados de segurança e armazenamento (através do FTP FileZilla) do código compilado no servidor. Fonte: Biblioteca de imagens do autor

Após o processo final para instalação do módulo API, foi feito o procedimento de compilação do módulo Aplicativo Web e posterior armazenamento no dispositivo móvel que foi utilizado para os testes (um celular A55 Samsung).

Para efeito de se ter a local base para posteriores instalações deste Aplicativo Web, o código compilado também foi publicado no mesmo servidor do módulo API. A Figura **??** mostra parte desse procedimento. Uma vez instalado os módulos, foi realizado a operacionalização, com acesso as opções de configuração, conforme visto no Item 3 dessa dissertação.

V.1.1 Descrição do domínio usado para os testes

Após a etapa de montagem de ambiente, iniciou-se os testes de campo contando da data de 01/01/2023 até 22/02/2024, com duração de 11 meses, no qual foram realizadas as tarefas de obtenção e classificação dos dados, com exibição dos gráficos gerados pela Plataforma, Entre alguns objetivos dos testes, foi possível verificar se o funcionamento da Plataforma estava aderente ao objetivo desta pesquisa.

Os testes iniciaram-se, primeiro no Laboratório de Robótica Educacional, localizado na sede, primeiro andar do Bloco E, sala E110, da Unidade Maracanã, visando ajustes locais do ESP32 e compilação dos códigos.

Posteriormente a solução foi testada a partir da residência do autor, verificando os deslocamentos diários, entre a residência, trabalhos, e locais de estudos, utilizando o veículo do próprio autor como objeto usado para obtenção de dados e bloqueio.

V.1.2 Coleta de dados para armazenamento e operações de bloqueio

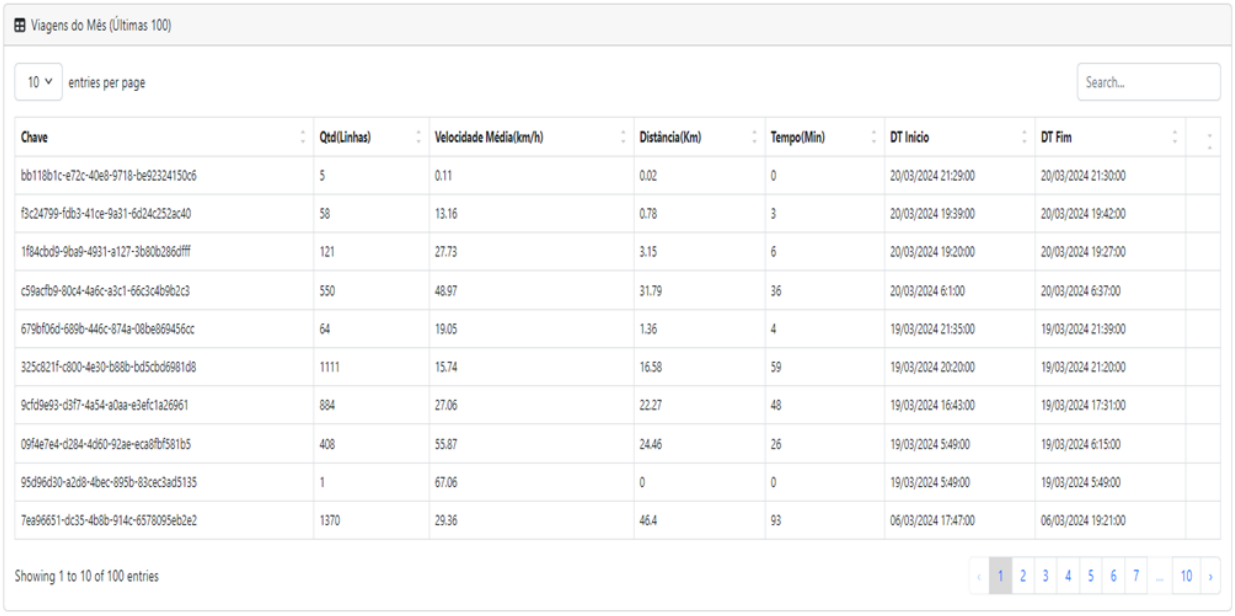
Durantes os 11 meses de testes, com o veículo do autor, foram coletadas informação de deslocamento, distância, velocidade, locais padrões, classificação e comparações com os dados obtidos dos radares móveis da cidade do Rio de Janeiro, no bairro Barra da Tijuca.

Foram comparados os parâmetros de velocidades lidas pela Plataforma (que é exibida no painel Led da caixa da Plataforma) com o que foi observado nos radares de velocidade, situados no final da Avenida Airton Sena (Barra da Tijuca-RJ), que exibem o valor da velocidade em painel visível, a partir do condomínio Alfa Barra. A Figura V.5 apresenta uma cena de um evento destes testes, datado do dia 31/01/2024. A gravação desta viagem pode ser acessada em: https://ideiaspara.com.br/MeuVeiculo/testes/31_01_2024.mp4.



Figura V.5: Cena mostrando que a Plataforma identificou 39 Km/H em comparação com o que o radar identificou, no dia 31/01/2024, as 8h do amanhã, Barra da Tijuca-RJ Fonte: Biblioteca de imagens do Autor.

Alguns exemplos de coleta podem ser vistos nas Figuras V.7 , V.8, V.6. Estas mostram várias coletas de dados em intervalos diferentes, com filtros de consulta diferentes, de modo que seja possível identificar viagens, velocidade média, latitude e longitude de origem e destino e os outros parâmetros que vão compor o BD. Esses dados são os que foram utilizados para o treinamento e consequente classificação pelo processo de Machine Learning da API e do Aplicativo Web.



Viagens do Mês (Últimas 100)

10 entries per page

Search...

Chave	Qtd(Linhas)	Velocidade Média(km/h)	Distância(Km)	Tempo(Min)	DT Inicio	DT Fim
bb118b1c-e72c-40e8-9718-be92324150c6	5	0.11	0.02	0	20/03/2024 21:29:00	20/03/2024 21:30:00
f3c24799-fdb3-41ce-9a31-6d24c252ac40	58	13.16	0.78	3	20/03/2024 19:39:00	20/03/2024 19:42:00
1f84cbd9-9ba9-4931-a127-3b80b286dfff	121	27.73	3.15	6	20/03/2024 19:20:00	20/03/2024 19:27:00
c59acfb9-80c4-4a6c-a3c1-66c3c4b9b2c3	550	48.97	31.79	36	20/03/2024 6:1:00	20/03/2024 6:37:00
679bf06d-689b-446c-874a-08be869456cc	64	19.05	1.36	4	19/03/2024 21:35:00	19/03/2024 21:39:00
325c821f-c800-4e30-b88b-bd5cbd6981d8	1111	15.74	16.58	59	19/03/2024 20:20:00	19/03/2024 21:20:00
9cfd9e93-d3f7-4a54-a0aa-e3efc1a26961	884	27.06	22.27	48	19/03/2024 16:43:00	19/03/2024 17:31:00
09f4e7e4-d284-4d60-92ae-eca8fbf581b5	408	55.87	24.46	26	19/03/2024 5:49:00	19/03/2024 6:15:00
95d96d30-a2d8-4bec-695b-83cec3ad5135	1	67.06	0	0	19/03/2024 5:49:00	19/03/2024 5:49:00
7ea96651-dc35-4b8b-914c-6578095eb2e2	1370	29.36	46.4	93	06/03/2024 17:47:00	06/03/2024 19:21:00

Showing 1 to 10 of 100 entries

‹ 1 2 3 4 5 6 7 … 10 ›

Figura V.6: Relatórios de viagens em um mês específico, apresentando as linhas de histórico inseridas, a média de velocidade por viagens, distâncias totais percorridas e latitudes/longitudes do inicio e final d ecada viagem. Fonte: Biblioteca de imagens do Autor.



Figura V.7: Gráfico gerado pelas Plataforma identificando as viagens, por dia, com as distâncias percorridas para cada viagem identificada, sendo que, assim, alimenta-se o processo de treinamento da Plataforma e permite-se que o algoritmo de Machine Leraning possa classificar como sendo uma viagem padrão ou não. Fonte: Biblioteca de Imagens do Autor.

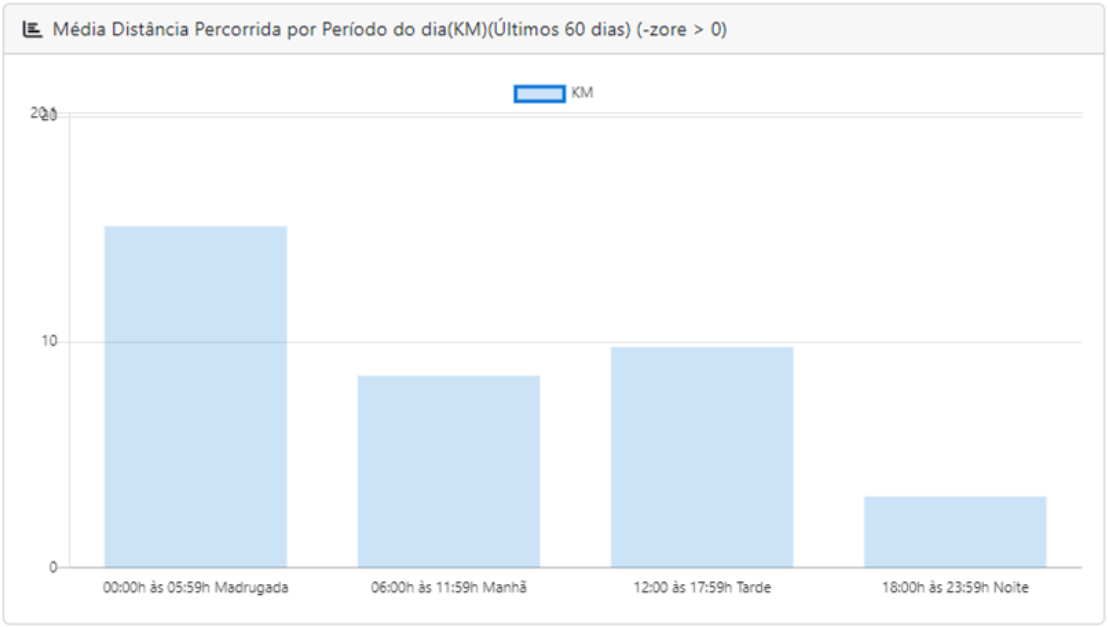


Figura V.8: Gráfico de média de distâncias percorridas em um dia (com vários períodos, tais como, pela manhã, tarde e noite), gerado pela Plataforma para fins de verificação do usuário (servindo para ver o quanto se gasta de quilometragem por um período determinado. Fonte: Biblioteca de imagens do Autor.

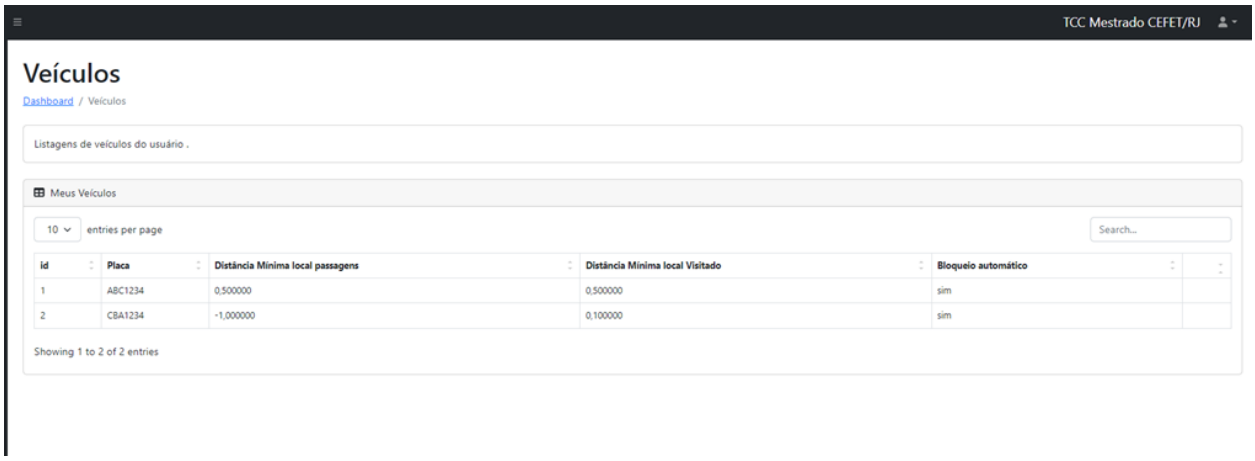


Figura V.9: Tela de cadastro de usuários e configuração de bloqueios Fonte: Biblioteca de imagens do Autor.

Na Figura V.10 , V.11 observa-se todos os dados coletados das 1.228 viagens totais examinadas no período de 11 meses, que geraram um total de 184.710 registros de vários dados nas tabelas do BD, inclusive os dados dos bloqueios automáticos “provocados” pela metodologia de testes.

Esses bloqueios são disponibilizados na forma do relatório visto na Figura V.12. Eles foram assim classificados pelo algoritmo de Machine Learning, que identifica dados inconsistentes com os padrão de rotas gravados no BD.

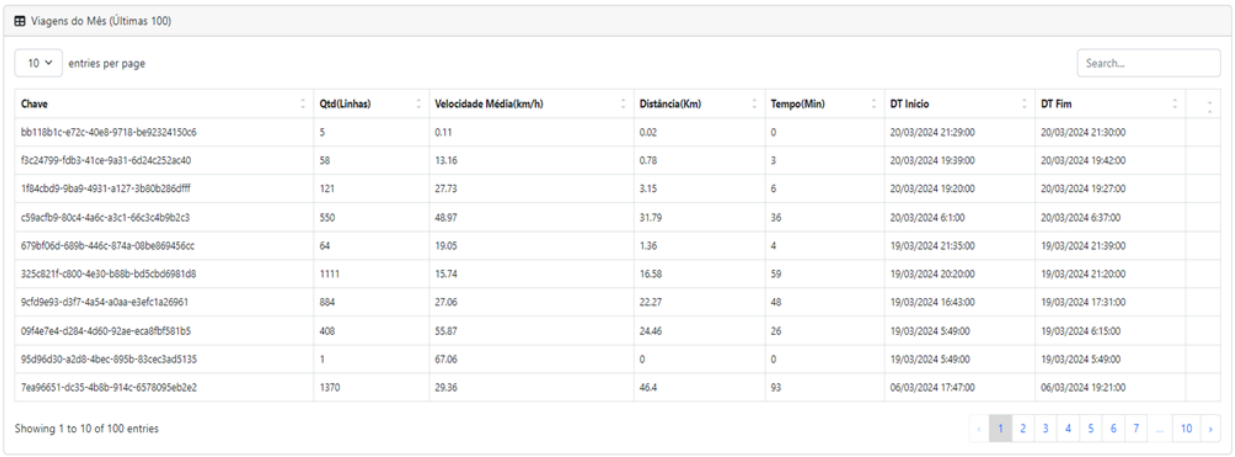


Figura V.10: Relatório gearado pela Plataforma apresentando as 1.228 viagens coletadas, que gerou 184.710 registros inseridos automaticamente no BD da Plataforma Fonte: Biblioteca de Imagens do Autor

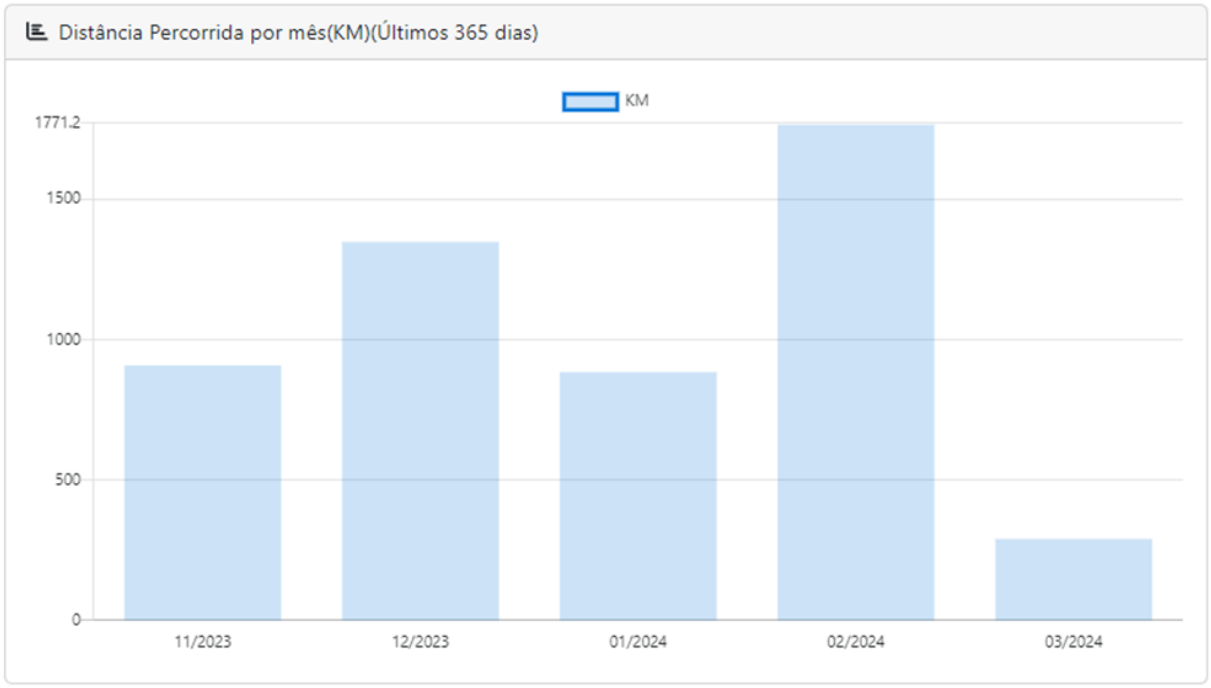


Figura V.11: Apresentação gráfica das distâncias percorridas pelo veículo controlado pela Plataforma, sendo indicado mês a mês. Dentro de um período especificado (no caso 11 meses) Fonte: Biblioteca de imagens do Autor.

```
1 •  SELECT    DATE_FORMAT(CONVERT( data_log  , DATE) ,   '%d/%m/%Y' ) data , mensagem , count(0) qtd_ocorrencias
2     ,(select count(0) from  vw_viagens ) qtd_viagens
3     ,(select count(0) from  historico ) qtd_linhas_historico
4       FROM  ocorrencias
5     group by  DATE_FORMAT(CONVERT( data_log  , DATE) ,   '%d/%m/%Y' )   , mensagem  order by data_log   desc
6
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

data	mensagem	qtd_ocorrencias	qtd_viagens	qtd_linhas_historico
02/03/2024	Bloqueado automaticamente, veículo passou por local sinalizado com ocorrências de subtração de veículos	2	1241	185556
02/03/2024	Bloqueado automaticamente, usuário por local sinalizado por terceiros como de ocorrências	1	1241	185556
02/03/2024	Bloqueado automaticamente, usuário nunca visitou o local	112	1241	185556
02/03/2024	Bloqueado automaticamente, usuário nunca passou próximo do local	88	1241	185556

Figura V.12: Relatório de bloqueio disponibilizados nos 11 meses de testes da Plataforma, gerado pela procedures/métodos do BD, mostrando os bloqueios (e os tipos, se automático ou não) durante o período Fonte: Biblioteca de Imagens do Autor

Um aspecto visto nos testes foi que a confirmação de como que a Plataforma sinaliza sobre situações de bloqueio, podendo a mesma efetua-lo automaticamente ou só avisando ao usuário para que esse tome a decisão, utilizando as mensagens adequadas, tais

como:

• Bloqueio automático: Identifica uma situação de que o usuário nunca passou no local observado, ou seja, o veículo é bloqueado por se encontra em um lugar estranho ao armazenado no BD.

• Bloqueios automático: Veículo passou perto de um local inexistente no cadastro, mas com uma proximidade considerada, pelo treinamento, como suspeita.

• Bloqueio automático: Veículo se encontra em um local que não foi visto como início ou término

de alguma outra viagem cadastrada no BD.

• Bloqueio automático: Veículo passou por local sinalizado com ocorrências de subtração de veículos.

• Bloqueio Não-Automático: Nesse modo, os avisos são feitos ao usuário, sem haver bloqueio automático, mas deixando a opção para o usuário emitir um sinal de bloqueio, as causas são as mesmas de bloqueio automático.

V.1.3 Comparações de aferição de medição de velocidade média do veículo

Um fenômeno relevante observado, foi que durante os teste em campo, observou-se que as velocidades médias, aferidas pelo odômetro veicular e o protótipo divergiam. Em geral, o protótipo lia uma velocidade 4,88% menor do que a velocidade mostrada no odômetro do veículo, como pode ser visto nas figura V.5 e V.13 que exibem essas comparações.

Após exaustivos testes, observou-se que na verdade os odômetros veiculares possuem uma calibragem um pouco superior, que as velocidades reais lidas no medidores. Como pode inclusive ser visto na figura V.13, a velocidade aferida pelo protótipo e por medidores de velocidades de trânsito, são equivalentes, desmistificando inclusive a crença popular de que esses radares possuem alguma margem inferior a medição real.



Figura V.13: Cena mostrando que a Plataforma identificou 39 Km/H em comparação com o que o radar identificou, no dia 30/03/2024, as 8h do amanhã, Barra da Tijuca-RJ Fonte: Biblioteca de imagens do Autor.

V.1.4 Comparações de aferição de distância

Também foram feitas verificações se se o uso da Fórmula de Haversine, para o cálculo de distância das viagens estariam precisos. Como evidência desta situação de precisão, a Tabela 8 exibe a amostragem destas comparações de uma viagem, de modo que o resultado obtido pela Plataforma dos parâmetro de latitude e longitude, de origem e destino, gerada pelo GPS da Plataforma, pudesse ser relacionado aos dados extraídos pelo aplicativo Google Maps, da Google.

Tabela V.1: Comparação dos dados obtidos pela GPS da Plataforma com os extraídos do Google Maps da viagem de 17/02/2024

Data de 17/02/2024	Latitude Origem	Longitude Origem	Latitude Destino	Longitude Destino	Distância
Obtido do GPS da Plataforma	22.828150	42.112068	22.764268	43.364084	157.024042
Obtido do Google Maps	22.567159	41.012769	22.794200	43.404083	154.024040

Pode-se notar que a precisão dos dados da Plataforma em relação ao obtidos pelo Google Maps forma bastante adequados. Há um desvio bem pequeno nas medições, mantendo-se praticamente similares os valores de longitude, latitude (origem e destino) e a distância percorrida com uma diferença percentual da ordem de menos de 1%.

Essa diferença entre o obtido pela Plataforma e o calculado pelo Google Maps gerou uma diferença de cerca de precisão de 3 km, sendo a medida da Plataforma em cerca de 157 km. A hipótese para essa diferença é atribuída pelos métodos de cálculo do Google Mapas e da Plataforma.

A conclusão é que o uso da Fórmula de Haversine pela Plataforma pode fornecer dados bem precisos, com uma margem de erro menor que 2% e cuja diferença pode acontecer por problemas de consideração de ponto inicial e final dados pelas ferramentas de medição externas à Plataforma, que tendem a ser mais imprecisos. Um exemplo pode ser visto na A Figura V.14, que apresenta a rota identificada pelo Google Maps.

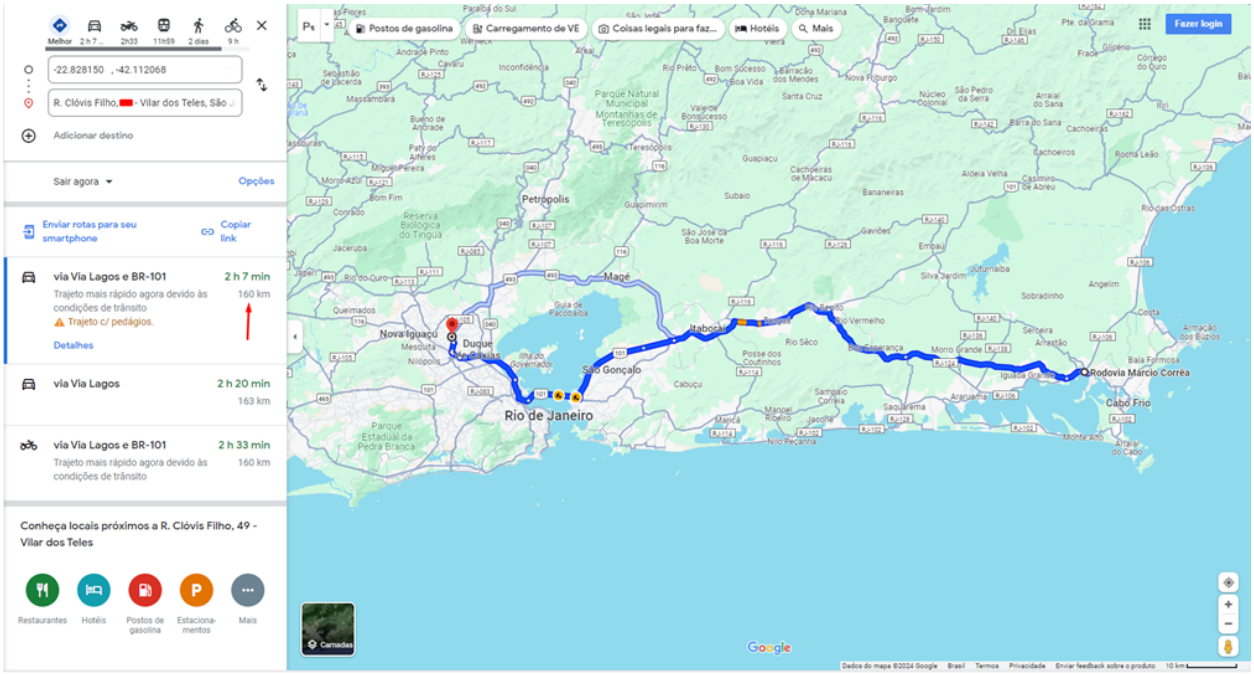


Figura V.14: Rota do dia 17/02/2024 vista através da ferramenta Google Maps, no qual o Maps mediu 154 km e a Plataforma mediu 157 km de distância percorrida . Fonte: Elaboração Própria

V.1.5 Classificações de padrões de viagens

De modo a permitir uma classificação mais precisa dos padrões de viagens e do uso do veículo monitorado, aplicou-se, nesta pesquisa, técnicas de agrupamento e remoção de outlieirs (valores discrepantes) através de aplicação de desvio padrão dos dados, identificando os pontos de dados que estão fora de um certo número de desvios padrão da média, para “aparar” e eliminar esse outliers.

Uma vez eliminado os outliers ou sua influência nos demais cálculos executados pela Plataforma, para melhor análise dos padrões de viagens e movimentações, foram utilizados algoritmos de treinamento baseado em algoritmos de Naïve Bayes, adequado ao uso de Machine Leraning de classificação, no qual se utiliza de dados de treinamento Arar and Ayan [2017] .

O modelo foi treinado usando um conjunto de dados rotulados, obtidos pela Plataforma e transformados nos registros do BD da mesma. Esse treinamento para classificação dos dados serve para se aplicar às situações de bloqueio visto no item V.1.2, que mostra as classificações previstas para execução pela Plataforma.

Como exemplo dos métodos de treinamento, através dos testes aplicados, notou-se uma maior frequência de uso do veículo monitorado durante as terças e quartas feiras. Esse resultado reflete a realidade, visto que é o dia que o veículo é usado para deslocamento de caso até o trabalho, um percurso de aproximadamente 20 km. Tal análise pode ser observado nos gráficos gerados pela Plataforma e apresentados pelo Aplicativo Web vistos nas Figuras V.15 e V.16.



Figura V.15: Gráfico gerado pela Plataforma que apura distância média percorrida em dois meses consequentes, identificando as corridas por dia da semana. Fonte: Biblioteca de imagens do Autor.

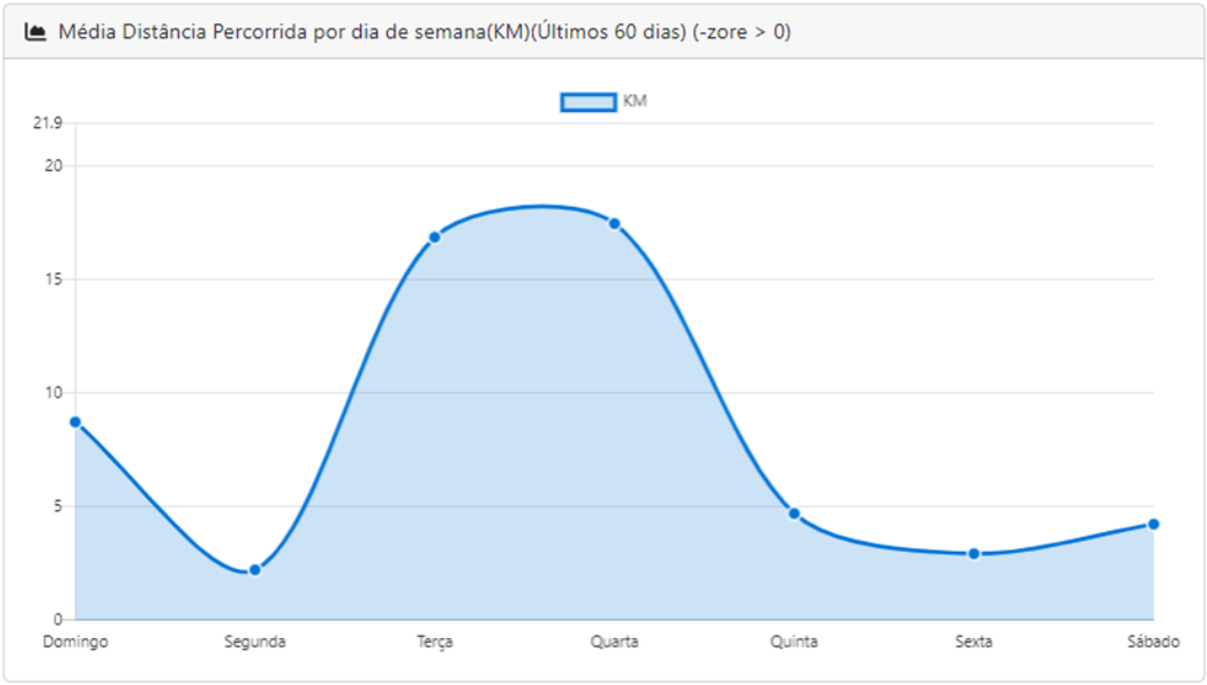


Figura V.16: Outro gráfico gerado pela Plataforma mostrando a distância média percorrida em outros dois meses consequentes, confirmando que os dias da semana de maior movimentação do veículo são terça e quarta, Fonte: Biblioteca de imagens do Autor.

Outro indicador utilizado pela Plataforma para identificar a frequência de uso e auxiliar na

eventual descoberta de fuga dos padrões, é aquele que usa o período do dia (manhã, tarde ou noite) das viagens. O resultado de gráficos gerados para analisar períodos/dia pode ser observado nas Figura V.17 e V.18.

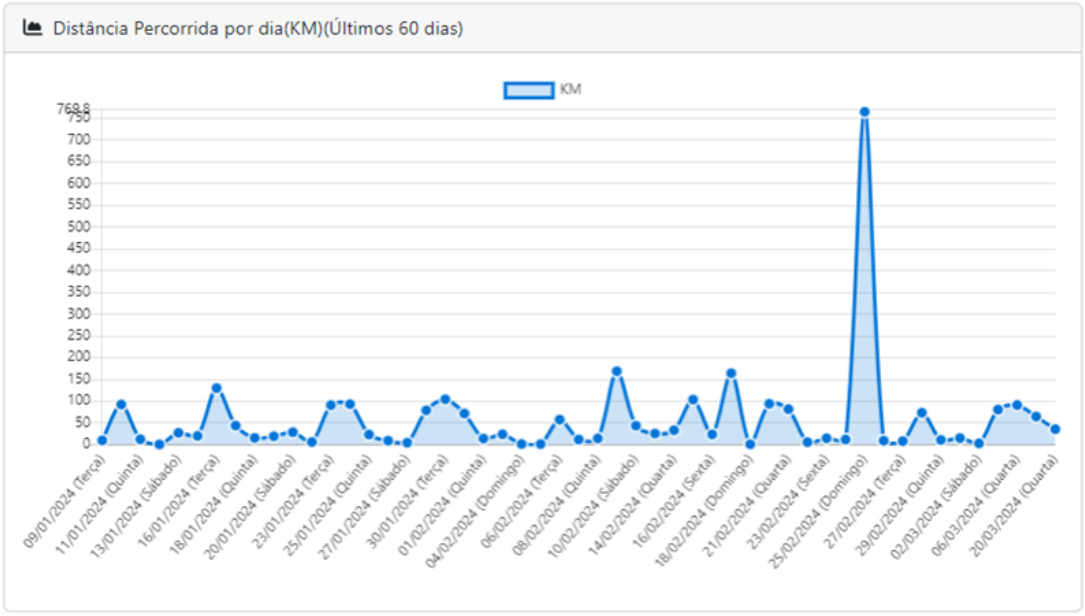


Figura V.17: Grafico apresentando do mês de fevereiro de 2024 (por dias d euso do veículo), indicando quel dia houve mais viagens pela parte da manhã, no caso no 25/02/2024. Fonte Biblioteca de imagens do autor.

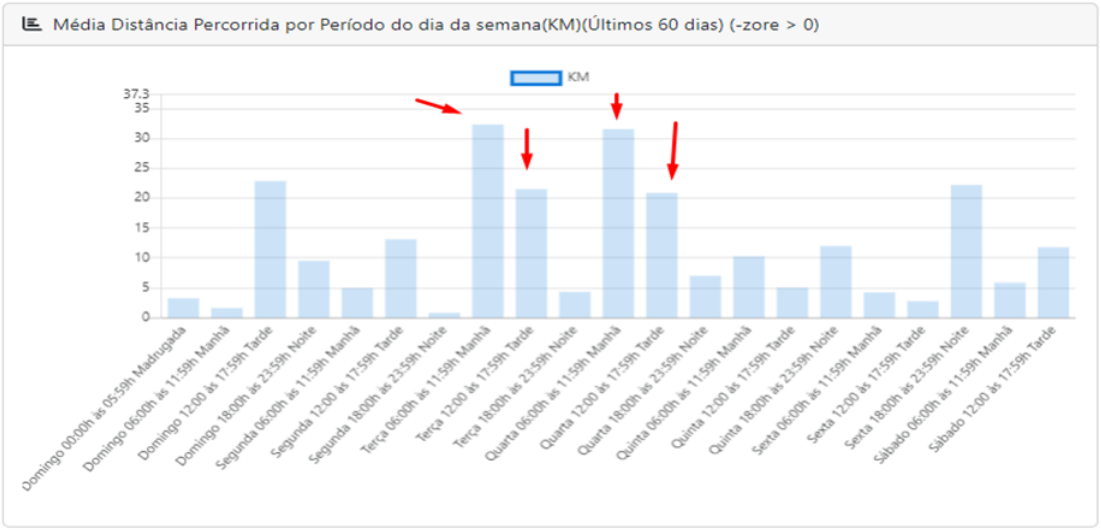


Figura V.18: Gráfico apresentando aa distância média percorrida no mês de fevereiro de 2024, por dia da semana e período do dia. Fonte Biblioteca de imagens do autor.

Capítulo VI Considerações Finais

Diante do observado em trabalhos relacionados, nota-se que as abordagens mais comuns para o problema de evitar, prevenir ou até mesmo facilitar a recuperação de veículos roubados ou furtados, perpassam pela dependência de alguma ação do usuário após o fato ocorrido, o que nem sempre é comum.

Por outro lado, há ainda a necessidade de se utilizar de tecnologias como Machine Learning para acurar o aumento da segurança para condutores de veículos através de aprendizado de seus próprios padrões de uso de veículo, assim como com ocorrências reportadas anteriormente por perfil de outros usuários, o que também classifica a presente solução, como de caráter colaborativo. Observa-se então, que a presente abordagem, embasou-se na necessidade de entregar uma plataforma com nível maior de automatização, objetividade e de caráter mais popular.

Cabe salientar, que ao longo do desenvolvimento do projeto e das pesquisas em si, houve uma grande dificuldade de se encontrar trabalhos científicos correlatos ao presente trabalho, motivo pelo qual as referências encontradas acabam sendo de componentes específicos, isto é, obteve-se tabalhos com algumas características de partes específicas usadas para a Plataforma, sem, contudo ter-se visto algum trabalho com solução similar completa.

Esta dificuldade também pode ser enxergada como uma grande oportunidade para este trabalho preencher esta lacuna de trabalhos científicos com a presente abordagem. Outra lacuna observada, foi a dificuldade de encontrar soluções de IOT com custo baixo, o que se entende, que com essa pesquisa, acabou trazendo um resultado promissor, mostrando que é possível entregar solução com um grau de inteligência e de custo baixo.

Destaca-se também o fato que consultando atores do mercado corporativo, como seguradoras e cooperativas de seguro, também observou-se uma aderência deste trabalho a este mercado de seguros, porém com uma segunda aplicação, que seria a de verificar os padrões de uso, para validar o risco que determinado motorista tem de sofrer algum sinistro e atuar na prevenção , tanto de risco de furto ou roubo, objeto inicial deste trabalho, assim como prevenir acidentes causados por uma condução imprudente por exemplo.

Como solução colaborativa, a medida que mais usuários venham a aderir o uso da ferramenta, mais sinalizações de usuários de áreas de risco são imputadas e portanto, reflete-se em mais segurança para os usuários da aplicação.

Tendo em vista que a solução, além de coletar dados de localização, também coleta e armazena parâmetros de velocidade, abre-se um universo de possibilidades de aplicabilidade tanto para esta solução em si, quanto para os dados gerados, que podem ser compartilhados com outras aplicações ou com entidades de segurança pública, oferecendo mais um apoio no enfrentamento aos altos índices de ocorrências do tipo estudado por este objeto.

Contudo, entende-se, que também há espaço para se inserir outros indicadores para acurar acerca da tomada de decisão de bloquear o veículo, assim como, incrementar os demais módulos, visando evoluir o protótipo desta aplicação , em um produto final, pronto para ser distribuído ao mercado.

Em se tratando de um trabalho que envolve ciência de dados, os usos futuros dos dados produzidos e/ou capturados, sofrem proteção da LGPD, portanto, dever-se-á levar em consideração essa proteção, sendo importante ter-se a total ciência dos usuários que os dados compartilhados devem apenas ser de dados não sensíveis, ou ainda o uso consentido pelos responsáveis legais dos dados, como prevê a lei LGPD em seu 11º artigo.

Um outro fator relevante , e portanto, de extrema pertinência, é que componentes tecnológicos são impactados por uma rápida obsolescência.Com o seguimento veicular não poderia ser diferente, portanto, este se encontra em grande mudança devido ao impacto dos veículos de propulsão elétrica.

Dito isto, este projeto se encontra aderente a essa nova tendência, tendo em vista que independente da propulsão , sejam eles térmicos ou elétricos, e de novas tecnologias que ainda possam ser exploras, ainda assim possuem componentes vitais a seu funcionamento.Estes componentes por sua vez, podem então ser chaveados com esta plataforma, por tanto, serem controlados de acordo com a presente solução abordada.

Por fim, acredita-se que o objetivo deste estudo foi atingido, pois foi construído uma solução robusta, adaptativa, escalável, que automatiza a tomada de decisão de bloqueio, que geram dados valiosos que poderão ser usados na própria evolução da solução, assim como em políticas públicas. As características apresentadas, levam a crer que a solução preenche de algum modo uma lacuna no meio acadêmico/científico.

Referências Bibliográficas

Arar, F. and Ayan, K. A feature dependent naive bayes approach and its application to the software defect prediction problem. *Applied Soft Computing*, 59:197–209, 2017. 64

Bala Dhanalakshmi, S., Deepa, N., M, A., and S, A. Auto guiding system for campus. In *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, pages 1440–1444, 2022. 27, 28

Barbosa, F. J. Uma abordagem para aprimorar a eficiência e a escalabilidade do back-end, 2023. 25, 26

Bazzi, C. L. Introdução a banco de dados, 2018. 30

Black, J. E., Holmes, D. E., and Carr, L. M. A geography of marine plastics. *Irish Geography*, 53(1):58–91, 2020. 21

Bowman, M. Mobile web development: When, why, and how. *Toptal Blog*, 2023. 22

Braga, A., de Oliveira, O. N., Pinto, W. A., and Pedrosa, W. A. Using machine learning algorithms for car theft prediction. *Expert Systems with Applications*, 128:130–140, 2019. 24

Cruz, L. R., Delatore, F., and Prado, A. C. Estudo do sistema de gps com hardware arduino para implementação em veículos agrícolas. In *VIII Simpósio de Iniciação Científica, Didática e de Ações Sociais da FEI*, 2018. 39

da República, P. Lei geral de proteção de dados pessoais (lgpd), 2018. 43

Dauni, P., Firdaus, M., Asfariani, R., Saputra, M., Hidayat, A., and Zulfikar, W. Implementation of haversine formula for school location tracking. volume 1402, 2019. 28, 29

Desai, M. and Phadke, A. Internet of things based vehicle monitoring system. In *2017 Fourteenth International Conference on Wireless and Optical Communications Networks (WOCN)*, pages 1–3, 2017. 22

Donohue, R. et al. Time series proportional symbol maps with leaflet and jquery. *Cartographic Perspectives*, 76, 2020. 40

Escalante, M., García-De-León, F., Dillman, C., et al. Genetic introgression of cultured rainbow trout in the mexican native trout complex. *Conservation Genetics*, 15:1063–1071, 2014. 35

Farias, Menelau, and Soares. Adoção de aplicativos de transporte e seu impacto na economia colaborativa., 2022. 16

FBSP, F. B. d. S. P. Anuário brasileiro 2022: Os crimes patrimoniais no brasil: entre novas e velhas dinâmicas, 2022. 16, 17

FGV. Brasil tem 424 milhões de dispositivos digitais em uso, revela a 31ª pesquisa anual do fgvcia, 2020. 16

Garcia, C. M. and Abilio, R. Integração entre sistemas utilizando web services rest e soap: Um relato prático, 2017. 26

Garey, M. and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. 0-7167-1044-7, United States of América, 13 edition, 1979. 20, 27

Harbinarka, S. Car accident information sending tool booktitle="tugas akhir thesis, university of technology. turkey, 2022. 17

Hizham, F. A. and Ginardi, R. V. H. Development of information retrieval method and haversine formula to determine clinic recommendation in jember. In *2021 International Conference on ICT for Smart Society (ICISS)*, Dhaka, Bangladesh, 2021. 20, 27

Irúrtia, A. and Iglesias, X. *A formação de jovens atletas: da detecção do talento à intervenção federativa*. Institut Nacional d'Educació Física de Catalunya, Barcelona, 2009. 35

Khan, S. U., Alam, N., Jan, S. U., and Koo, I. S. Iot-enabled vehicle speed monitoring system. *Electronics*, 11(4):614, 2022. 22

Lee, S., Tewolde, G., and Kwon, J. Design and implementation of vehicle tracking system using gps/gsm/gprs technology and smartphone application. In *IEEE World Forum on Internet of Things (WF-IoT)*, pages 353–358, 2014. 19, 20

Letto, R. et al. Software defined radio-based gps spoofing attack model on road navigation system. In *Advances in Intelligent Systems and Computing*, volume 1340. Springer, 2022. 17

Leão, A. A. P., Fritz, C. K., Dias, M. R. M. G., Carvalho, J. A. R., Mascarenhas, L. P. G., Cat, M. N. L., Radominski, R., and Nesi-França, S. Bone mass and dietary intake in children and adolescents with type 1 diabetes mellitus. *Journal of Diabetes and its Complications*, 34(6):107573, 2020. 35

Mattar, J. and Ramos, D. K. *Metodologias de pesquisa para trabalhos educativos: quantitativa, qualitativa e mistas*. Ed Edições 70, 2 edition, 2021. 53

Monawar, T., Mahmud, S. B., and Hira, A. Anti-theft vehicle tracking and regaining system with automatic police notifying using haversine formula. In *4th International Conference on Advances in Electrical Engineering*, pages 775–779, Dhaka, Bangladesh, 2017. 21

Monk, S. *Projetos com Arduino e Android: Use seu Smartphone ou Tablet para controlar o Arduino*. Bookman, Porto Alegre, 2014. 39

Mounika, A. and Chepuru, A. Iot based vehicle tracking and monitoring system using gps and gsm. *International Journal of Recent Technology and Engineering (IJRTE)*, 8(2S11):2399–2403, 2019. 19, 20, 22

Neves, A. L. M. and Domingues, D. S. *Manual da Metodologia de Pesquisa Científica*. Escola de Aperfeiçoamento de Oficiais - ESAO, 2007. 49

Nichat, M. Landmark based shortest path detection by using a* algorithm and haversine formula, 2013. 28

Qadir, A. M. and Cooper, P. Gps-based mobile cross-platform cargo tracking system with web-based application. In *8th International Symposium on Digital Forensics and Security*, pages 353–358, 2020. 21

Tsindeliani, D., Povstyana, Y., Lishchyna, N., and Yashchuk, A. Latency reduction in real-time gps tracking in android and the web-based gps monitoring system. In *2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, pages 1–7, 2022. 21, 22

Tursunbayev, Y., Lee, Y.-S., Chang, H.-Y., and Shin, Y.-S. A review of machine learning models for vehicle theft prediction. *IEEE Access*, 9:37723–37737, 2021. 24

Tuyisenge, J. C., v. Ishimwe, Iradukunda, J. P., and (2021), D. M. Development of security starting system for vehicles based on iot. In *International Conference on Information Technology (ICIT)*, pages 505–510, 2021. 20, 21

Verka Jovanović, Vladmir Lazović, N. M. Sql query execution time between two singidunum university locations. In *Synthesis 2015 - International Scientific Conference of IT and Business-Related Research*, pages 624–628, 2015. 29, 30

W3C.BR. Boa prática 23: Disponibilizar dados por meio de uma api, 2020. 22

Zheng, Q., Yu, Z., Zheng, X., Zheng, Y., and Huang, G. A machine learning approach for predicting car thefts. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3868–3877, 2019. 24

ANEXO A – Código fonte do módulo veículo (C para ESP32)

```
  #include <RCSwitch.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <TinyGPS++.h>
#include <DNSServer.h>
#include <WebServer.h>
#include <WiFiManager.h>
#include <ArduinoJson.h>
#include <TM1637Display.h>

//GPS
#define RXD2 16
#define TXD2 17
HardwareSerial neogps(1);
TinyGPSPlus gps;
//GPS


RCSwitch mySwitch = RCSwitch();
int oldvalue;
int value;
unsigned long tempoChavelida = millis();
unsigned long tempoReconectarWifi = millis();



//const char* ssid       = "ANALUISA";
//const char* password   = "11082016";

//const char* ssid       = "SPEED";
//const char* password   = "Emb159753";


//const char* ssid       = "Brilho nos Olhos";
//const char* password   = "Vitus@0308!#";

const char* ssid       = "jeugenio";
const char* password   = "12345678";

int LED_VIVO = 2;
int LED_COMANDO = 15;
int BOTAO_CONFIG_WIFI = 2;
```

```
43 String latitude_lida = "";
44 String longitude_lida = "";
45 String mensagem_lida = "";
46 double velocidade_lida = 0;
47 String chave_viagem = "";
48
49
50
51
52 WiFiManager wifiManager;//Objeto de manipula  o do wi-fi
53
54
55 //Rele de corte de Bomba
56 int PINO_RELE_COMANDO = 23;
57 int PINO_RELE_GND = 21;
58 int PINO_RELE_5V = 22;
59 //Rele de corte de Bomba
60
61
62 //Display 7 seguimentos
63 #define PINO_DISPLAY_CLK  26
64 #define PINO_DISPLAY_IO  27
65 TM1637Display display = TM1637Display(PINO_DISPLAY_CLK, PINO_DISPLAY_IO);
66 int PINO_DISPLAY_GND = 14;
67 int PINO_DISPLAY_5V = 12;
68 //Display 7 seguimentos
69
70
71
72 //Wifi Mobile
73 //callback que indica que o ESP entrou no modo AP
74 void configModeCallback (WiFiManager *myWiFiManager) {
75   Serial.println("Entrou no modo de configura  o");
76   Serial.println(WiFi.softAPIP()); //imprime o IP do AP
77   Serial.println(myWiFiManager->getConfigPortalSSID()); //imprime o SSID criado da
       rede
78 }
79
80 //Callback que indica que salvamos uma nova rede para se conectar (modo esta  o)
81 void saveConfigCallback () {
82   Serial.println("Configura  o salva");
83 }
84 //Wifi Mobile
```

```
85
86
87
88
89  void setup() {
90    Serial.begin(9600);
91
92    //S rial  M dulo GPS
93    neogps.begin(9600, SERIAL_8N1, RXD2, TXD2);
94
95    //mySwitch.enableReceive(0);  // Receiver on interrupt 0 => that is pin #2 //
        Arduino pino 2
96    pinMode(GPIO_NUM_35, INPUT);
97    mySwitch.enableReceive(digitalPinToInterrupt(GPIO_NUM_35)); // pino 35 esp32 e
        pino 2 no arduino
98
99
100
101
102   pinMode(PINO_DISPLAY_5V, OUTPUT);
103   pinMode(PINO_DISPLAY_GND, OUTPUT);
104   delay(50);
105   digitalWrite (PINO_DISPLAY_GND, LOW);
106   digitalWrite (PINO_DISPLAY_5V, HIGH);
107   delay(150);
108   //display.setBrightness(0x0f);
109  //display.showNumberDec(-1, false);
110
111
112
113   pinMode(PINO_RELE_COMANDO, OUTPUT);
114   pinMode(PINO_RELE_GND, OUTPUT);
115   pinMode(PINO_RELE_5V, OUTPUT);
116   delay(50);
117   digitalWrite (PINO_RELE_COMANDO, LOW);
118   digitalWrite (PINO_RELE_GND, LOW);
119   digitalWrite (PINO_RELE_5V, HIGH);
120
121
122   Serial.println("Ligado.");
123   pinMode (LED_VIVO, OUTPUT);
124   pinMode (LED_COMANDO, OUTPUT);
125  //Bot o de conectar wifi
```

```
126   pinMode(BOTAO_CONFIG_WIFI,INPUT);
127   digitalWrite (BOTAO_CONFIG_WIFI, LOW );
128
129  digitalWrite (LED_VIVO,   LOW );
130  digitalWrite (LED_COMANDO,  HIGH );
131
132  delay(500);
133  digitalWrite (LED_VIVO, HIGH );
134  digitalWrite (LED_COMANDO,  HIGH );
135
136  delay(500);
137
138  digitalWrite (LED_VIVO, HIGH );
139  digitalWrite (LED_COMANDO,  LOW );
140  delay(500);
141  Serial.print("..");
142  delay(500);
143
144 /* Fazer coenectar por paibel
145    //callback para quando entra em modo de configura  o AP
146   wifiManager.setAPCallback(configModeCallback);
147   //callback para quando se conecta em uma rede, ou seja, quando passa a trabalhar
        em modo esta  o
148   wifiManager.setSaveConfigCallback(saveConfigCallback);
149   */
150
151
152   conectarWifi();
153
154    //Teste Remover
155   //EnviarLocalizacao("1234","4321", "Latitude:-22.76-Longitude:-43.36-Velocidade
       :0.04-Satelites:9-Altitude:33.00");
156
157 }
158
159
160 void conectarWifi(){
161   delay(500);  //Atraso de 10 milissegundos
162    //connect to WiFi
163   Serial.printf("Conectando em %s ", ssid);
164   WiFi.mode(WIFI_STA);
165   WiFi.begin(ssid, password);
166   while (WiFi.status() != WL_CONNECTED) {
```

```
167       Serial.print("Conectado no wifi: ");
168       delay(500);
169       Serial.print(".");
170       Serial.print(WiFi.status());
171       digitalWrite (LED_VIVO,  !digitalRead(LED_VIVO));
172       digitalWrite (LED_COMANDO,  !digitalRead(LED_COMANDO) );
173
174   }
175    IPAddress ip = WiFi.softAPIP(); //Endere o de IP
176    Serial.print("Endere o de IP: "); //Imprime o endere o de IP
177    Serial.println(ip);
178    Serial.print("Conectado WIFI:");
179    Serial.println(ssid);
180
181 }
182
183
184 void reconectarWifi(){
185
186   if ((WiFi.status() != WL_CONNECTED) ) {
187     Serial.print(millis());
188     Serial.println("Reconectando no WiFi...");
189     delay(500);
190     digitalWrite (LED_VIVO,  !digitalRead(LED_VIVO));
191     digitalWrite (LED_COMANDO,  !digitalRead(LED_COMANDO) );
192     WiFi.disconnect();
193     WiFi.reconnect();
194   }
195 }
196
197 void DesligarRele(){
198   Serial.println("Deligando rele");
199   digitalWrite (PINO_RELE_COMANDO, HIGH);
200
201 }
202
203
204
205 void EnviarLocalizacao(String latitude , String longitude, String Mensagem ,
      String chave_viagem ){
206
207
208           Serial.print("Enviando localiza  o: ");
```

```
209         HTTPClient http;
210         //http.begin("http://192.168.1.11/MeuVeiculo/api/
    VeiculoGravarLocalizacao"); //Specify the URL
211         http.begin("https://ideiaspara.com.br/MeuVeiculoApi/api/
    VeiculoGravarLocalizacao"); //Specify the URL
212         http.setAuthorization("XXXXXXXXX");
213         http.addHeader("Content-Type", "application/json");
214 
215         //int httpCode = http.GET();    //Make the request
216         String Conteudo = "{'veiculo_id': 1,  'usuario_id': 1,  'latitude': '" +
     latitude + "',  'longitude': '" + longitude +"',  ' data_atualizacao':
    '2023-09-25T02:04:59.225Z', 'veiculo_mensagem_io':'"  + Mensagem + "'  , '
    chave_viagem':'"  + chave_viagem + "'  }";
217         int httpCode = http.POST(Conteudo);
218          Serial.println(Conteudo);
219 
220         if (httpCode > 0) { //Check for the returning code
221 
222             String payload = http.getString();
223             Serial.println(httpCode);
224             Serial.println(payload);
225 
226 
227             //VerificaComando Desligar Rel
228             DynamicJsonDocument doc(1024);
229             deserializeJson(doc, payload);
230 
231 
232 
233             bool desativar_veiculo = (bool)doc["desativar_veiculo"];
234             String i_chave_viagem = doc["chave_viagem"];
235             chave_viagem = i_chave_viagem;
236 
237             Serial.print("chave_viagem");
238             Serial.println(chave_viagem);
239 
240 
241             Serial.print("Desativar");
242             Serial.println(desativar_veiculo);
243             if(desativar_veiculo && velocidade_lida == 0){
244               DesligarRele();
245             }
246              //VerificaComando Desligar Rel
```

```
247
248             }
249
250           else {
251             Serial.println("Error no HTTP request");
252           }
253
254           http.end();
255  }
256
257
258 void ObterLocalizacao(){
259
260   boolean temDados = false;
261
262    while (neogps.available())
263     {
264       if (gps.encode(neogps.read()))
265       {
266         temDados = true;
267       }
268    }
269
270    if(temDados){
271
272      // if (gps.location.isValid() ==  1)
273      // {
274
275         Serial.print("Lendo gps: ");
276         latitude_lida = String(gps.location.lat(),6);
277         longitude_lida = String(gps.location.lng(),6);
278         velocidade_lida =  gps.speed.kmph();
279         display.setBrightness(0x0f);
280         display.showNumberDec(velocidade_lida, false);
281
282
283         Serial.print("Lat: ");
284         Serial.println(gps.location.lat(),6);
285         Serial.print("Lng: ");
286         Serial.println(gps.location.lng(),6);
287         Serial.print("Velocidade: ");
288         Serial.println(gps.speed.kmph());
289         Serial.print("Satelite:");
```

```
290         Serial.println(gps.satellites.value());
291         Serial.print("Altitude:");
292         Serial.println(gps.altitude.meters(), 0);
293 
294         Serial.print("Data: ");
295         Serial.print(gps.date.day()); Serial.print("/");
296         Serial.print(gps.date.month()); Serial.print("/");
297         Serial.println(gps.date.year());
298 
299         Serial.print("Hora: ");
300         Serial.print(gps.time.hour()); Serial.print(":");
301         Serial.print(gps.time.minute()); Serial.print(":");
302         Serial.println(gps.time.second());
303         Serial.println("####------#####");
304 
305         mensagem_lida = "Latitude:" +  String(gps.location.lat(),6) + "-Longitude:
    " +  String(gps.location.lng(),6) + "-Velocidade:" + String(gps.speed.kmph()) +
     "-Satelites:" + String(gps.satellites.value()) + "-Altitude:" + String(gps.
    altitude.meters());
306         Serial.println("mensagem_lida" +  mensagem_lida);
307 
308 
309       // }
310       // else
311       // {
312       //   Serial.println("Sem sinal GPS");
313       // }
314   }
315   else
316   {
317       Serial.println("Nenhum dado de gps");
318   }
319 }
320 
321 void loop() {
322 
323 
324 /* Fazer coenectar por paibel
325     //Se o bot o for pressionado
326    if (digitalRead(BOTAO_CONFIG_WIFI) == HIGH) {
327       Serial.println("Abertura Portal"); //Abre o portal
328       digitalWrite (LED_VIVO, HIGH );
329       digitalWrite (LED_COMANDO,  HIGH );
```

```
330
331     wifiManager.resetSettings();       //Apaga rede salva anteriormente
332     if(!wifiManager.startConfigPortal("IDEASFOR", "12345678") ){ //Nome da Rede
    e Senha gerada pela ESP
333       Serial.println("Falha ao conectar"); //Se caso n o conectar na rede
    mostra mensagem de falha
334       delay(2000);
335       ESP.restart(); //Reinicia ESP ap s n o conseguir conex o na rede
336     }
337     else{       //Se caso conectar
338       Serial.println("Conectado na Rede!!!");
339       ESP.restart(); //Reinicia ESP ap s conseguir conex o na rede
340     }
341   }
342   */
343
344   if(WiFi.status()== WL_CONNECTED){ //Se conectado na rede
345
346       digitalWrite (LED_VIVO, LOW );
347       digitalWrite (LED_COMANDO, HIGH );
348
349     // digitalWrite(32,HIGH); //Acende LED AZUL
350   }
351   else{ //se n o conectado na rede
352
353     digitalWrite (LED_VIVO,   LOW );
354     digitalWrite (LED_COMANDO,  HIGH );
355
356     delay(500);
357     digitalWrite (LED_VIVO, HIGH );
358     digitalWrite (LED_COMANDO,  HIGH );
359
360     delay(500);
361
362     digitalWrite (LED_VIVO, HIGH );
363     digitalWrite (LED_COMANDO,  LOW );
364     delay(500);
365     Serial.print("..");
366     // wifiManager.autoConnect();// Fun   o para se autoconectar na rede
367   }
368
369
370
```

```
371  //digitalWrite (LED_VIVO, LOW );
372  //digitalWrite (LED_COMANDO, HIGH );
373
374 if((millis() - tempoChavelida) > 500){ // N o le repetido
375
376    tempoChavelida  = millis();
377    ObterLocalizacao();
378    if(latitude_lida != "" && longitude_lida != ""){
379       EnviarLocalizacao(latitude_lida,longitude_lida, mensagem_lida, chave_viagem)
       ;
380    }
381 }
382
383 if((millis() - tempoReconectarWifi) > 5000){
384   tempoReconectarWifi  = millis();
385   reconectarWifi();
386 }
387
388 }
389
```

ANEXO B – código fonte do endpoint "VeiculoGravarLocalizacao", construído sob o código fonte C.

```
1       /// <summary>
2       /// M todo usado pelo m dulo  ve culo para gravar a  ltima
    localiza  o
3       /// </summary>
4       /// <param name="localizacao"></param>
5       /// <returns></returns>
6       [HttpPost]
7       [System.Web.Http.Route("api/VeiculoGravarLocalizacao")]
8       public HttpResponseMessage VeiculoGravarLocalizacao(LocalizacaoDTO
    localizacao)
9       {
10          HttpResponseMessage response = new HttpResponseMessage();
11          AuthenticationHeaderValue authHeader = Request.Headers.Authorization;
12          string token = "";
13          if (authHeader != null)
14          {
15              token = authHeader.ToString();
16          }
17          string Tipo = authHeader is null ? "" : authHeader.Scheme;
18          if (!new VeiculoRepositorio().ValidarAutenticacao(token, Tipo).
    RetornoOK)
19          {
20              return Request.CreateResponse(HttpStatusCode.Unauthorized, new
    VeiculoRepositorio().ValidarAutenticacao(token, Tipo));
21          }
22
23          try
24          {
25              LocalizacaoOcorrenciasDTO locVeiculo = new
    LocalizacaoOcorrenciasDTO();
26              VeiculosDTO veiculo = new VeiculoRepositorio().VeiculoObter(new
    VeiculosDTO {  usuario_id  = localizacao.usuario_id,  id = localizacao.
    veiculo_id });
27
28              if (string.IsNullOrEmpty(localizacao.chave_viagem))
29              {
30                  if(veiculo.id != 0)
31                  {
32                      TimeSpan tempo = DateTime.Now - veiculo.data_atualizacao ;
33                     if(  tempo.TotalMinutes <= 5)
34                      {
```

```
35                          localizacao.chave_viagem = veiculo.
    veiculo_chave_viagem; // Feito para recuperar o ultimo token
36                      }
37                      else
38                      {
39                          localizacao.chave_viagem = Guid.NewGuid().ToString();
    // Feito para gerar o primeiro guid da viagem
40                      }
41                  }
42                  else
43                  {
44                      localizacao.chave_viagem = Guid.NewGuid().ToString(); //
    Feito para gerar o primeiro guid da viagem
45                  }
46
47              }
48              locVeiculo.chave_viagem = localizacao.chave_viagem;
49
50              locVeiculo.desativar_veiculo = false;
51              locVeiculo.localizacao = new VeiculoRepositorio().
    GravarLocalizacao(localizacao);
52              locVeiculo.ocorrencias = new VeiculoRepositorio().
    ListarOcorrencias(localizacao);
53              if (locVeiculo.ocorrencias != null && locVeiculo.ocorrencias.Count
     > 0)
54              {
55                  locVeiculo.desativar_veiculo = true;
56              }
57
58
59              // Chana de forma ass ncrona a l gica de previs o
60              Task.Run(() => new VeiculoRepositorio().VerificaBloqueio(
    localizacao));
61              // Chana de forma ass ncrona a l gica de previs o
62
63              response = Request.CreateResponse(HttpStatusCode.OK, locVeiculo);
64
65              response.Headers.CacheControl = new CacheControlHeaderValue();
66          }
67          catch (Exception ex)
68          {
69              throw new HttpResponseException(Request.CreateResponse(
    HttpStatusCode.InternalServerError, ex));
```

```
70
71              }
72
73              return response;
74          }
75
76
77
```

ANEXO C – código fonte da procedure SetVeiculoLocalizacao em TSQL

```
1     CREATE DEFINER=‘jeugenio25_MeuVeiculoUser‘@‘%‘ PROCEDURE ‘
      SetVeiculoLocalizacao‘(IN p_usuario_id int , IN p_veiculo_id int ,  IN
      p_latidude varchar(254)  , IN p_longitude varchar(254) , IN
      p_veiculo_mensagem_io varchar(254) , IN p_chave_viagem varchar(254)  )
2  BEGIN
3
4  DECLARE latitude_ant    decimal(18,6);
5  DECLARE longitude_ant   decimal(18,6);
6  DECLARE qtd_viagens   int ;
7
8  DECLARE p_data_inicio DATE;
9  DECLARE  p_data_fim DATE;
10
11 DECLARE distancia_dia_semana    decimal(18,6);
12 DECLARE var_veiculo_distancia_percorrida    decimal(18,6);
13
14
15
16    select   count(0) into  qtd_viagens  from  historico where  chave_viagem =
      p_chave_viagem order by id desc  limit 1  ;
17
18    IF  qtd_viagens is not null or  qtd_viagens =   0 THEN
19
20    update veiculos set
21
22    veiculo_distancia_percorrida =    0
23
24    where id = p_veiculo_id ;
25
26
27    END IF;
28
29
30    select   CAST(latitude as decimal(11,4))  ,  CAST( longitude as decimal(11,4))
       into  latitude_ant , longitude_ant from  historico where  chave_viagem =
      p_chave_viagem order by id desc  limit 1  ;
31
32    update veiculos set latitude = CASE WHEN CAST(p_latidude as decimal(11,4))  <>
       0  THEN   p_latidude ELSE latitude END  ,
33    longitude = CASE WHEN CAST(p_longitude as decimal(11,4))  <>  0  THEN
      p_longitude ELSE longitude END
34   , veiculo_chave_viagem = p_chave_viagem
```

```
35    , data_atualizacao = current_timestamp() ,
36     veiculo_codigo =  case when veiculo_codigo is null then   UUID() else
      veiculo_codigo end ,
37     veiculo_mensagem_io = p_veiculo_mensagem_io ,
38     veiculo_distancia_percorrida = veiculo_distancia_percorrida +   HAVERSINE(
      latitude_ant,  longitude_ant,  CAST(p_latidude as decimal(11,4)) ,  CAST(
      p_longitude as decimal(11,4)) )
39
40     where id = p_veiculo_id ;
41
42 IF   CAST(p_latidude as decimal(11,4))  <>  0 THEN
43
44     insert into historico ( veiculo_id , latitude , longitude, veiculo_mensagem_io
      , velocidade, chave_viagem, distancia_percorrida )
45     select p_veiculo_id  , p_latidude,  p_longitude,    p_veiculo_mensagem_io ,
      CAST(SUBSTRING_INDEX(  SUBSTRING_INDEX(veiculo_mensagem_io, "Velocidade:", -1)
      ,  "-", 1)  as decimal(18,2)) ,
46     p_chave_viagem , veiculo_distancia_percorrida from  veiculos where id =
      p_veiculo_id ;
47
48
49     select veiculo_distancia_percorrida into var_veiculo_distancia_percorrida
        from  veiculos where id = p_veiculo_id ;
50
51    -- values( p_veiculo_id ,  p_latidude , p_longitude, p_veiculo_mensagem_io,
52    -- CAST(SUBSTRING_INDEX(  SUBSTRING_INDEX(veiculo_mensagem_io, "Velocidade:",
      -1) ,  "-", 1)  as decimal(18,2)), p_chave_viagem    ) ;
53
54     select id veiculo_id, p_usuario_id usuario_id,  latitude, longitude ,
      data_atualizacao , p_veiculo_mensagem_io ,
55     CAST(SUBSTRING_INDEX(  SUBSTRING_INDEX(veiculo_mensagem_io, "Velocidade:", -1)
       ,  "-", 1)  as decimal(18,2))  velocidade,
56     p_chave_viagem  from  veiculos where id = p_veiculo_id ;
57
58
59     -- Bloco de c digo que identifica quantas viagens se aproximam do  da
      localiza  o
60
61     -- Bloco de c digo que identifica quantas viagens se aproximam do  da
      localiza  o
62
63 END IF;
64
```

```
65 -- testes
66  -- Lugar Visitado
67  -- call SetVeiculoLocalizacao (1, 1, '-22.776440', '-43.361248', 'Latitude
       :-22.776440-Longitude:-43.361248-Velocidade:0.61-Satelites:7-Altitude:53.20','
       efe9722e-f270-4217-8121-24e9079a43a4') ;
68  -- Lugar Nunca Visitado
69  -- call SetVeiculoLocalizacao (1, 1, '-23.5489', '-46.6388', 'Latitude:-23.5489-
       Longitude:-46.6388-Velocidade:0.61-Satelites:7-Altitude:53.20','efe9722e-f270
       -4217-8121-24e9079a43a4') ;
70
71
72     /* call SetVeiculoLocalizacao (1, 1, '12345678', '87654321') ;  */
73     /* call SetVeiculoLocalizacao (1, 1, '0.00', '0.00', 'Latitude:0.00-Longitude
       :0.00-Velocidade:0.00-Satelites:0-Altitude:0.00') ;  */
74
75 END
76
77
```