

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Fernando Paim da Silva

**SISTEMA PARA GERENCIAMENTO DE FROTAS AUTOMOTIVAS
BASEADO EM IOT**

Santa Maria, RS
2021

Fernando Paim da Silva

SISTEMA PARA GERENCIAMENTO DE FROTAS AUTOMOTIVAS BASEADO EM IOT

Trabalho de Conclusão de Curso apresentado ao CURSO de BACHARELADO em CIÊNCIA DA COMPUTAÇÃO, Área de Concentração em Sistemas de Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em CIÊNCIA DA COMPUTAÇÃO**. Defesa realizada por videoconferência.

ORIENTADORA: Prof.^a Lisandra Manzoni Fontoura

©2021

Todos os direitos autorais reservados a Fernando Paim da Silva. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.


Fone (0xx) 55 99161 6747; End. Eletr.: fsilva@inf.ufsm.br

FERNANDO PAIM DA SILVA

SISTEMA PARA GERENCIAMENTO DE FROTAS AUTOMOTIVASBASEADO EM IOT

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**. Defesa realizada por videoconferência.

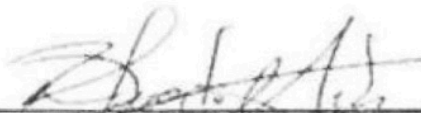
Aprovado em 12 de Fevereiro de 2021:



Lisandra Manzon Fontoura, Dr. (UFSM)
(Presidente/Orientador)



Marcelo Hoss, Dr. (UFMS) (videoconferência)



Roseclea Duarte Medina, Dr. (UFSM) (videoconferência)

Santa Maria, RS
2021

AGRADECIMENTOS

Agradeço imensamente a minha esposa por todo apoio, compreensão e amor dado durante todos os anos de minha graduação. Uma Mulher decidida mas repleta de delicadeza. Apesar de termos origens humildes, sempre nos dedicamos ao trabalho e as diversas tarefas fadigantes que nos trouxeram extraordinárias realizações, nunca privados de sonhar e conquistar.

Agradeço da mesma forma aos meus pais e minhas irmãs, que nunca deixaram de acreditar nas minhas capacidades, inclusive quando eu mesmo não as acreditava. Todos estiveram presentes e possuem um lugar em meu coração.

Agradeço a minha orientadora por sua enorme paciência e empatia. Essa mesma dedicação não é creditada somente a mim, mas a todos os alunos em todas as suas aulas. Sem dúvida é um exemplo.

Não só isso, mas também nos gloriamos nas tribulações, porque sabemos que a tribulação produz perseverança; a perseverança, um caráter aprovado; e o caráter aprovado, esperança. E a esperança não nos decepciona, porque Deus derramou seu amor em nossos corações, por meio do Espírito Santo que ele nos concedeu.

(Romanos 5:3-5 ACF)

RESUMO

SISTEMA PARA GERENCIAMENTO DE FROTAS AUTOMOTIVAS BASEADO EM IOT

AUTOR: Fernando Paim da Silva

ORIENTADORA: Lisandra Manzoni Fontoura

No Brasil, a maior parcela da frota automotiva é rodoviária e, com a situação atual das rodovias, a coleta de dados destes veículos enquanto se deslocam possibilita uma imensa quantidade de benefícios. Este trabalho apresenta o projeto e criação de uma solução IoT que fornece informações em tempo real de informações dos automóveis pertencentes a frota automotiva, como: quilometragem rodada desde o arranque do veículo, monitoramento de localização GPS e velocidade. Os dados são coletados em um dispositivo móvel e são enviados via conexão de Internet para o servidor que os armazena. A construção do módulo IOT é feita por meio de um dispositivo *smartphone* e um adaptador Bluetooth ELM327 conectado a porta OBD2 do veículo. O software desenvolvido roda no dispositivo móvel que conecta no adaptador ELM327, coleta os dados da ECU veicular como também a informação de localização pelos sensores do *smartphone*, efetua cache local e envia ao servidor para persistência. Os dados coletados foram comparados aos do veículo quando em movimento por meio de seu painel de instrumentos e o aplicativo foi capaz de capturar estes dados em um intervalo de 10 segundos. No servidor, os dados foram exibidos através de suas médias ou a diferença no caso da distância percorrida e, a rota veicular foi exibida em um mapa.

Palavras-chave: Monitoramento Veicular. Telemetria. Internet das Coisas. IoT. On-board diagnostic. OBD. OBDII. OBD2. GPS.

ABSTRACT

IOT-BASED AUTOMOTIVE FLEET MANAGEMENT SYSTEM

AUTHOR: Fernando Paim da Silva
ADVISOR: Lisandra Manzoni Fontoura

In Brazil, the largest portion of the automotive fleet is on the road and, with the current situation on the highways, the collection of data from these vehicles as they travel enables an immense amount of benefits. This work presents the design and creation of an IoT solution that provides real-time information on automobiles belonging to the automotive fleet, such as: mileage traveled since the vehicle was started, GPS location and speed monitoring. The data is collected on a mobile device and is sent via Internet connection to the server that stores it. The construction of the IOT module is done through a smartphone device and a Bluetooth adapter ELM327 connected to the vehicle's OBD2 port. The developed software runs on the mobile device that connects to the EML327 adapter, collects the data from the vehicle ECU as well as the location information by the smartphone sensors, performs local cache and sends it to the server for persistence. The collected data were compared to the vehicle's data when in motion using its instrument panel and the application was able to capture this data in an interval of 10 seconds. On the server, the data was displayed through their averages or the difference in the case of the distance traveled and the vehicular route was displayed on a map.

Keywords: Vehicle Monitoring. Telemetry. Internet of Things. IoT. On-board diagnostic. OBD. OBDII. OBD2. GPS.

LISTA DE FIGURAS

Figura 2.1 – Porta OBDII de um veículo.	21
Figura 2.2 – Adaptador ELM327 Bluetooth.	22
Figura 3.1 – Diagrama de casos de uso.	26
Figura 3.2 – Diagrama de atividade para ciclo de coleta de dados.	28
Figura 4.1 – Aplicativo React Native em execução.	35
Figura 4.2 – Adaptador ELM327 Bluetooth conectado ao Nissan March.	36
Figura 4.3 – Adaptador ELM327 Bluetooth conectado Chevrolet Montana.	36
Figura 4.4 – Tela de visualização dos dados.	37
Figura 4.5 – Tela de visualização da rota.	38

LISTA DE GRÁFICOS

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

<i>IoT</i>	Internet of Things
<i>ECU</i>	Engine Control Unit
<i>OBD</i>	On-Board Diagnostic
<i>OBD2</i>	On-Board Diagnostic Version 2
<i>GPS</i>	Global Positioning System
<i>API</i>	Application Programming Interface
<i>REST</i>	Representational State Transfer
<i>Covid</i>	Corona Virus Disease
<i>adhoc</i>	"Para este fim"
<i>Wi – Fi</i>	Wireless Fidelity
<i>WLAN</i>	Wireless local área network
<i>USB</i>	Universal Serial Bus
<i>PID</i>	Parameter Identification
<i>MIL</i>	Malfunction Indicator Lamp
<i>ISO</i>	Internacional Organization for Standardization

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVOS	15
1.1.1	Objetivos específicos	15
1.2	JUSTIFICATIVA	15
1.3	ORGANIZAÇÃO DO TEXTO	16
2	REFERENCIAL TEÓRICO	17
2.1	SITUAÇÃO DA FROTA AUTOMOTIVA NO BRASIL	17
2.2	GESTÃO DE FROTAS	18
2.3	<i>INTERNET OF THINGS</i> (IOT)	19
2.4	TECNOLOGIAS PASSÍVEIS DE UTILIZAÇÃO	20
2.4.1	OBD2	20
2.4.2	Adaptadores ELM327	21
2.4.3	Protocolos de comunicação Bluetooth	22
2.4.4	Python versão 3	22
2.4.5	JavaScript	23
2.5	TRABALHOS CORRELATOS	23
2.6	SOFTWARES DISPONÍVEIS SIMILARES	24
3	DESENVOLVIMENTO	25
3.1	ESPECIFICAÇÃO	25
3.1.1	Diagrama de casos de uso	25
3.1.2	Diagrama de atividades	26
3.2	IMPLEMENTAÇÃO	29
3.2.1	Ferramentas utilizadas	29
4	AVALIAÇÃO DOS RESULTADOS	30
4.1	COLETA DOS DADOS PELO SENSOR	30
4.1.1	Problema 1	32
4.1.2	Problema 2	33
4.1.3	Veículos Flex	34
4.1.4	Aplicativo móvel	34
4.2	ENVIO DOS DADOS E PERSISTÊNCIA PELA API	37
5	CONCLUSÃO	39
	REFERÊNCIAS BIBLIOGRÁFICAS	40

1 INTRODUÇÃO

Com a crescente integração de dispositivos à Internet e o aumento da necessidade por dados confiáveis e precisos, diversas áreas puderam se apropriar da tecnologia embarcada de módulos IoT (*Internet of Things*) para coleta de dados em tempo real.

No Brasil, a maior parcela da frota automotiva é rodoviária e, com a situação atual das rodovias, a coleta de dados destes veículos enquanto se deslocam possibilita uma imensa quantidade de benefícios, como a identificação de diversos padrões importantes para a tomada de decisão empresarial e até mesmo governamental, no caso da existência de uma rede de compartilhamento destas informações (SINDIPEÇAS; ABIPEÇAS, 2019).

Os dados que podem ser coletados são inúmeros, por exemplo o tempo de viagem, velocidade média na via, consumo de combustível instantâneo e médio, quilometragem rodada, quantidade de combustível no tanque, tempo e quilometragem entre abastecimentos, entre outros. Além dos benefícios mais óbvios, esses dados podem ser importantes também para a detecção de possíveis fraudes ou até mesmo furtos, pois a coleta quase que instantânea pode contribuir para coibir ou mesmo impedir esses eventos. Ainda, pode-se utilizar estes dados para detecção de comportamentos do motorista, como identificar possíveis atos nocivos ao volante, prolongamento das horas dirigindo, abuso de velocidade em vias remotas e até mesmo dados de trânsito, como tempo de congestionamento, dias e horários de menor fluxo, etc.

Desde que os carros fabricados passaram a possuir um computador embarcado, conhecido como ECU (acrônimo para *Engine Control Unit*), cujo o objetivo é controlar de uma forma mais precisa a ignição do motor, diversos modelos passaram a contar com uma porta de conexão a este computador embarcado e um protocolo que facilita a obtenção dos dados de diagnóstico destes veículos, o qual foi chamado OBD (acrônimo para *On-Board Diagnostic*). Este protocolo se tornou obrigatório a partir de 1996 na Europa e Estados Unidos (MACHADO; OLIVEIRA, 2007).

A partir de 2009, de acordo com CONAMA (2004), a exigência da implantação do protocolo OBD versão 2 contempla a totalidade da frota brasileira para veículos fabricados ou importados. Esta resolução, apesar de visar o controle de emissões de poluentes, permitiu a fácil integração com a ECU destes veículos para a obtenção destes dados.

De posse destas informações, este trabalho visa desenvolver um sistema de controle de frota que use os dados automotivos coletados a partir de um módulo IoT, a ser desenvolvido usando hardware de baixo custo e comum ao mercado, para gerar informações relevantes a gestão de frota.

1.1 OBJETIVOS

Desenvolver um sistema de controle de frotas automotivas capaz de coletar dados através de um módulo IoT e avaliar esses dados gerando informações para o gerenciamento da frota.

1.1.1 Objetivos específicos

- Implementar a coleta de dados veiculares conforme protocolo OBD2 disponível no veículo pela ECU veicular e interpretados pelo módulo ELM327;
- Implementar a coleta de dados de GPS no dispositivo *smartphone*;
- Projetar e desenvolver o sistema de API REST;
- Implementar o envio dos dados coletadas ao servidor web;
- Analisar os dados gerando informações relevantes a gerência da frota.

1.2 JUSTIFICATIVA

O gestor responsável pela administração e controle da frota automotiva da empresa enfrenta diversos desafios no dia a dia e, muitas das informações necessárias para este controle não estão facilmente disponíveis ou dependem de terceiros. Desta forma, calcular os custos com as atividade de transporte automotivo ou mesmo o simples controle das próximas manutenções agendadas dos veículos acaba por ser uma dificuldade enfrentada (VALENTE et al., 2016).

Segundo (BERTAGLIA, 2020), o preço do produto final das organizações envolvidas com serviço de transporte, seja com veículos próprios ou terceirizados, depende diretamente do controle e do conhecimento dos ativos, bem como de seguir importantes práticas de negócios, pois é a melhor forma destas organizações se manterem competitivas e reduzirem seus custos.

Com abordagem deste trabalho visa usar simplesmente um *smartphone*, que muitas vezes já está disponível aos motoristas e, da conexão do mesmo com módulo Bluetooth OBD2 de baixo custo, este trabalho pode tornar mais democrática as soluções de gestão de frotas.

1.3 ORGANIZAÇÃO DO TEXTO

O presente trabalho está organizado da seguinte maneira: o capítulo 2 trata sobre o referencial teórico relevante para o entendimento da situação atual da gestão de frota no Brasil, além de explicar as tecnologias usadas no decorrer da pesquisa. No capítulo 3 são tratados os detalhes do desenvolvimento do protótipo, no capítulo 4 são apresentados e avaliados os resultados obtidos, e para finalizar, no capítulo 5 é apresentada a conclusão do trabalho realizado.

2 REFERENCIAL TEÓRICO

Neste capítulo são apresentadas as motivações para o desenvolvimento deste trabalho, descrevendo a importância e o significado de tecnologias embarcadas, bem como a IoT. Também são descritas algumas tecnologias passíveis de utilização no decorrer deste trabalho.

2.1 SITUAÇÃO DA FROTA AUTOMOTIVA NO BRASIL

No Brasil, de acordo com (MINISTÉRIO DA INFRAESTRUTURA, 2020), em outubro de 2020, temos uma frota nacional composta por mais 107 milhões de veículos entre seus mais diversos tipos. Porém, para se ter uma ideia da quantidade de veículos automotores que possuem potencial para a coleta de dados de acordo com os requisitos deste trabalho, deste total podemos selecionar os que foram fabricados posterior à 2010 e obtemos um total de 46.834.150 veículos. O relatório divulgado é dividido em arquivos que são categorizados por diversos indicadores, porém seu cruzamento para que possamos efetuar uma seleção dos dados mais precisa aos alvos deste estudo se torna impossível pela falta de informações que vinculem seus dados. Além destes, não fica claro quais veículos são circulantes e fazem parte da mobilidade urbana brasileira.

Com intuito de se ter uma ideia mais precisa da situação da frota brasileira de veículos que possam ser alvos deste trabalho, podemos observar o relatório divulgado pelo Sindicato Nacional da Indústria de Componentes para Veículos Automotores (Sindipeças) relativo ao ano de 2019, no qual a frota circulante brasileira é estimada em cerca de 45,9 milhões, sem considerar os veículos de categoria Motocicletas. De acordo com o relatório publicado, sabe-se que a idade média dos veículos no ano de 2019 é de 9 anos e 8 meses, portanto podemos concluir que a ampla maioria dos veículos foi fabricada posterior a 2010 e, necessariamente, implementa o protocolo OBD2.

Observa-se que a frota brasileira possui um crescimento estimado de 2,5% para o ano de 2019 e de 3% para 2021, em relação aos anos imediatamente anteriores, porém as projeções feitas foram calculadas antes dos reflexos da pandemia de Covid-19. Além destes dados, é interessante ressaltar que desde 2010 houve uma queda na taxa de variação anual da frota e a partir de 2016 um novo crescimento da taxa, porém menor que em anos anteriores, ou seja, percebe-se que houve um envelhecimento da frota deste 2010 até o momento, o que reflete uma diminuição na venda de modelos novos comparando com o ano de 2010 onde se teve o maior pico entre os anos disponíveis no relatório. A frota brasileira está em sua maioria concentrada nos estados de São Paulo, Minas Gerais, Paraná, Rio de Janeiro e Rio Grande do Sul, com sua distribuição de maior à menor

respectivamente nesta ordem e, representam um total de 64,3% de todos os autoveículos circulantes do País no ano de 2019 (SINDIPEÇAS; ABIPEÇAS, 2019).

2.2 GESTÃO DE FROTAS

Gestão de frotas entende-se pela atividade de reger, administrar ou gerenciar um conjunto de ativos, mais especificamente veículos de uma mesma empresa ou organização. A tarefa em si é muito ampla e passa por diversos serviços, por exemplo manutenção mecânica, cálculo de custos, dimensionamento ou mesmo a roteirização. A tarefa de calcular precisamente um padrão de participação da frota no patrimônio e nos custos das empresas não é algo simples, pois esta parcela varia conforme cada caso e, depende por exemplo, dos objetivos da organização (se carga própria, de terceiros, passageiros, entre outros), da forma como os veículos são incorporados e contabilizados na frota (se por financiamento, aluguel ou *leasing*) e ainda da natureza da operação (VALENTE et al., 2016).

Segundo (BERTAGLIA, 2020), as organizações envolvidas com serviço de transporte, seja com veículos próprios ou terceirizados, precisa controlar e conhecer de forma efetiva os seus ativos e seguir importantes práticas de negócios, pois é a melhor forma de se manterem competitivas e reduzirem seus custos logísticos e, por consequência, o preço de seu produto final, ficando evidente a importância da correta administração. A administração dos transportes tem como um de seus elementos mais importantes a gestão das frotas, principalmente se for sua atividade final da cadeia de abastecimento. Em muitos casos, a movimentação de carga tem um peso significativo na formação dos custos logísticos e na qualidade do serviço prestado. Ao longo dos anos, organizações que possuíam frotas menores cresceram, passando a aumentar a quantidade de veículos. No entanto, esse aumento não significa que sua operação seguiu sendo realizada de forma mais eficiente, podendo em muitos casos estar aumentando seus custos de forma excessiva e desnecessária, ou ainda perdendo qualidade na prestação de seu serviço.

A gestão de frota possui diversos aspectos relevantes que precisam serem levados em conta como a administração dos ativos, mão de obra, consumo de combustível, manutenção mecânica, estoque de peças de reposição, planejamento de atividades administrativas, eficiência da frota, indicadores de desempenho e produtividade, unitização de cargas, entre outras. Todas esses fatores combinados justificam o uso da tecnologia da informação na administração de transportes. Atualmente, com o avanço da tecnologia, diversas ferramentas importantes foram criadas para fornecer suporte as tarefas do processo de transporte, como exemplo podemos destacar os sistemas de otimização de rotas, de rastreamento de frota, bem como a telemática (BERTAGLIA, 2020).

2.3 INTERNET OF THINGS (IOT)

Atualmente vivemos uma crescente revolução tecnológica com a criação constante de objetos inteligentes, sejam pulseiras ou palmilhas que são capazes de contar quantos passos você andou e compartilhar com seus amigos, ou sejam sistemas inteligentes que acendem as luzes e aquecem o jantar quando você se aproxima de casa. Todas essas tecnologias estão relacionados ao conceito de *Internet of Things* que vem sendo construído (MAGRANI, 2018).

Toda essa tecnologia que vem sendo criado está mudando à forma como interagimos com o mundo a nossa volta (MAGRANI, 2018). De posse de todas estas informações, coletadas por estes dispositivos de forma autônoma, podemos rastrear e contabilizar tudo de forma que seja possível reduzir muito o desperdício, a perda e o custo (ASHTON, 2009).

De acordo com (ASHTON, 2009), por volta de 1999, Kevin Ashton propôs o termo *Internet of Things* (com acrônimo IoT) ou Internet das Coisas, que acabou por se popularizar dez anos depois com a publicação de seu artigo "That 'Internet of Things' Thing" para o RFID Journal. Segundo (ASHTON, 2009), a falta de tempo das pessoas em suas rotinas trará a necessidade da conexão dos inúmeros dispositivos diferentes de diversas maneiras.

Segundo (MAGRANI, 2018), o termo *Internet of Things* (com acrônimo IoT) ou Internet das Coisas, não possui um único conceito que possa ser considerado unânime, porém podemos entender IoT como sendo um ecossistema de computação que interconecta objetos físicos à Internet por meio de sensores, objetivando tornar o cotidiano das pessoas mais fácil, simplificando seus processos diários, sendo estes sensores onipresentes ou também chamados de Ubíquos. Ainda, (MAGRANI, 2018) encontra um entendimento comum entre os conceitos, a hiperconectividade entre estes computadores, sensores e objetos físico.

Segundo ITU-T (2012), também temos que IoT possui uma definição ampla com implicações tecnológicas e sociais, sendo que do ponto de vista técnico, IoT pode ser vista como sendo uma infraestrutura global para a socialização da informação possibilitando serviços avançados, interconectando coisas (físicas e virtuais), com base na tecnologias de informação e comunicação interoperáveis existentes e em constante evolução. Ainda segundo ITU-T (2012), temos que a identificação, coleta de dados, processamento e comunicação são recursos necessários para esta oferta de serviços a todos tipos de aplicativos, porém garantindo a segurança e a privacidade.

Em 2020, estima-se que a quantidade de objetos inteligentes interconectados dobrará, passando dos 25 bilhões para 50 bilhões. Na economia, estima-se um cenário mais impressionante ainda, com a movimentação de 11 trilhões de dólares em 2025. Com essas estimativas tão animadoras, a IoT está na mira do setor privado e da gestão pública, pois apresenta uma possível solução para problemas de eficiência produtiva, poluição e

até mesmo criminalidade. Ainda, na área da saúde pode-se melhorar a interação médico e paciente através do monitoramento remoto constante através de dispositivos de saúde interconectados (MAGRANI, 2018).

Porém, essa coleta, armazenamento, transmissão e compartilhamento rotineira de dados por estes dispositivos, muitos destes dados particulares e íntimos, podem trazer riscos a privacidade e segurança destes usuários (MAGRANI, 2018).

2.4 TECNOLOGIAS PASSÍVEIS DE UTILIZAÇÃO

Nesta seção é apresentado uma breve apresentação das tecnologias que poderão ser utilizadas no decorrer da implementação deste trabalho.

2.4.1 OBD2

Criado em 1970 pela agência de proteção ambiental dos Estados Unidos da América (EPA), batizado por *on board diagnosis* (OBD), é um protocolo que tem o intuito de controlar a dispersão de gases pelos automóveis. Pouco tempo depois passou a controlar também o consumo de combustível e a ignição, devido aos limites cada vez mais rígidos sobre as emissões. Durante sua criação, uma das maiores preocupações era garantir que o veículo mantivesse o controle de suas emissões ao longo de toda sua vida útil, permitindo a partir de então que as autoridades pudessem fiscalizar através dos equipamentos corretos. Ainda, junto com todas estas funcionalidades, surgiu a necessidade de um processo chamado de auto-diagnóstico que facilitava os trabalhos de manutenção veicular a partir dos dados de falhas armazenados no dispositivo pelo protocolo (MACHADO; OLIVEIRA, 2007).



Figura 2.1 – Porta OBDII de um veículo.

2.4.2 Adaptadores ELM327

Segundo (ELECTRONICS, 2016), a grande maioria dos automóveis atuais possui o sistema de teste e diagnóstico OBD presente e, apesar da sua padronização na comunicação e transferência de dados, nenhum deles é utilizável diretamente em computadores ou mesmo em *smartphones*. Para suprir essa necessidade o dispositivo ELM327 foi projetado pela Elm Electronics para funcionar como uma ponte entre essas portas OBD e a interface RS232.

Os adaptadores baseados no chip ELM327 possuem diversas variações de modelos, preços e formato de conexão, sendo elas Wi-Fi, USB e Bluetooth. São dispositivos de baixo custo que se conectam diretamente na porta OBD veicular e fornecem informações em tempo real sobre o veículo (OLIVEIRA, 2017).



Figura 2.2 – Adaptador ELM327 Bluetooth.

2.4.3 Protocolos de comunicação Bluetooth

O protocolo Bluetooth é uma tecnologia de comunicação sem fio por ondas de rádio facilmente integrado em dispositivos eletrônicos e, com baixo custo e potência. Capaz de eliminar o uso de fios e cabos entre dispositivos, a tecnologia consegue comunicar criando redes *ad hoc* entre vários dispositivos à pequenas distância. Opera na faixa de frequência de 2,4 GHz e foi concebido para não competir com as tecnologias de redes sem fio WLANs, apesar de operarem na mesma frequência (KLEINSCHMIDT, 2004).

2.4.4 Python versão 3

Concebida por Guido van Rossum entre os anos 1989 e 1990, a linguagem Python recebeu este nome por motivo do seriado de comédia da BBC Monty Python's Flying Circus, o qual o criador era fã. A linguagem é um projeto de desenvolvimento de software de código-fonte aberto, no entanto Guido van Rossum ainda possui um papel definitivo nas decisões da evolução da linguagem. Projetada especialmente para tornar os programas muito legíveis, Python também é uma linguagem de uso geral (PERKOVIC, 2016).

Python é uma linguagem que suporta múltiplos paradigmas, sendo eles imperativo, orientado a objetos e funcional. Também é uma linguagem de alto nível, de tipagem

dinâmica e forte, gerenciamento automático de memória e escopo léxico. Implementa estruturas de dados embutidas como tuplas, listas e dicionários as quais aumentam muito a expressividade do código. Ainda, é uma linguagem onde a indentação define a forma do bloco de código, o que não é uma característica muito comum (CRUZ, 2015).

2.4.5 JavaScript

A linguagem JavaScript foi criada pela empresa Netscape, agora Mozilla, no início da fase da web, sendo que o seu nome é uma marca registrada e licenciada pela Sun Microsystems, agora Oracle. Quando enviada para a European Computer Manufacturers Association (ECMA) com intuito de padronização, a linguagem passou a ter o nome ECMAScript em sua versão padronizada. JavaScript é uma linguagem de alto nível, interpretada, dinâmica e fracamente tipada, sendo muito conveniente para estilos de programação orientada a objetos e funcional. Sua sintaxe é derivada da linguagem Java, da herança baseada em protótipos de *self* e das funções de primeira classe de *Scheme* (FLANAGAN, 2013).

2.5 TRABALHOS CORRELATOS

Durante a pesquisa sobre as informações necessárias ao trabalho, alguns trabalhos correlatos encontrados podem ser destacados, sendo descritos a seguir.

A monografia (ALMEIDA; FARIA, 2013) produziu um aplicativo *desktop* na plataforma LabVIEW®. Seu foco difere deste trabalho pois seu intuito era obter os dados sobre as falhas da DTC e armazená-las localmente.

A monografia (FAGUNDES; SILVA; ASSIS, 2015) trabalhou com a coleta dos dados OBD2 e de localização de forma similar a este trabalho, porém seu foco foi produzir um software para a plataforma de hardware EVK2, diferenciado-se deste trabalho por não ser feito em um aplicativo React Native.

A dissertação (OLIVEIRA, 2017) produziu um aplicativo móvel com intuito de capturar os dados de emissão de gases poluentes. Os dados coletados foram exclusivamente armazenados no dispositivo móvel, diferentemente deste trabalho.

2.6 SOFTWARES DISPONÍVEIS SIMILARES

Outros aplicativos que se integram a ECU por meio da porta OBD2 estão disponíveis nas lojas de aplicativos dos sistemas operacionais móveis. Entre as versões gratuitas encontradas, não foram encontrados versões com a função de monitoramento remoto. Também podem ser encontrados aplicativos para somente rastreamento de localização, no entanto sem a integração com ECU veicular. Todos os aplicativos encontrados e capazes de se conectar com o adaptador ELM327 foram construídos somente com o intuito de servirem como um *scanner* automotivo para detecção de falhas.

Existem ainda empresas que fornecem serviços e aplicativos com foco muito similar ao deste trabalho, porém são aplicativos licenciados e necessitam da aquisição de uma assinatura e/ou de um dispositivo OBD legado. Entre essas empresas, algumas fornecem um aplicativo de rastreamento veicular que também é capaz de se integrar com a ECU veicular, porém sua versão para testes gratuitos é disponível somente para empresas e requerem um cadastramento. As empresas verificadas foram: Cobli, Azul Tecnologia. Outras empresas similares também foram encontradas, porém sem testes gratuitos ou necessitam de um rastreador legado ou comercial, sendo elas: Bsoft, Smart localiza e Tiger. Também vale destacar que não fica claro nas suas páginas web se os softwares destas empresas são compatíveis com adaptadores ELM327.

Ainda, diversas outras empresas foram encontradas que fornecem software de rastreamento veicular com período de testes gratuitos, porém não se integram com a ECU do veículo e, os testes gratuitos também estão disponíveis somente para empresas. Estas foram: Infleet, Logistrack.

Entre as empresas pesquisadas, nenhuma forneceu um período de testes gratuito para pessoa física, sendo que ambas trabalham somente com outras empresas.

Este trabalho visou a criação de um sistema de monitoramento de frota de baixo custo e com hardware comercial simples disponível amplamente no mercado, desta forma, seu foco difere destes apresentados aqui.

3 DESENVOLVIMENTO

Este trabalho visa projetar e implementar um aplicativo *mobile* para *smartphone* com sistema operacional Android que seja capaz de se conectar ao adaptador EML327 por conexão Bluetooth e, através da troca de comandos seriais, obter dados em tempo real do veículo, bem como obter dados dos próprios sensores de localização do dispositivo. Os dados coletados foram enviados pelo dispositivo ao servidor que os armazenará. No servidor também é possível visualizar, por meio de uma página web, as informações coletadas conforme são recebidas, servindo como uma central de monitoramento para controle destes veículos.

Para o desenvolvimento deste trabalho, os custos envolvidos foram somente com a compra do adaptador ELM327 no valor de R\$26,90. O aparelho *smartphone* e o servidor não envolveram custos de aquisição ou licença, sendo que o *smartphone* usado era de uso pessoal do autor e o servidor utilizado foi hospedado na plataforma Heroku que fornece o serviço de forma gratuita.

3.1 ESPECIFICAÇÃO

Esta seção demonstra a partir de diagramas os requisitos do sistema web e do aplicativo a serem desenvolvidos.

3.1.1 Diagrama de casos de uso

Visando descrever as funcionalidades do aplicativo foi elaborado um diagrama de casos de uso. O usuário do sistema é o Gestor da Frota e as funcionalidades associadas a ele são:

A funcionalidade "Controlar a quantidade de quilômetros rodados pela frota" permite que o gestor da frota controle a distância percorrida em cada rota, de forma que o mesmo possa verificar possíveis incongruências.

A funcionalidade "Auditar a rota percorrida pelos veículos da frota" permite também que o gestor da frota possa verificar possíveis desvios de rotas.

A funcionalidade "Auditar a velocidade de um veículo em qualquer momento" permite ao gestor da frota verificar possíveis excessos de velocidade ou mesmo identificar momentos de congestionamento no trânsito.

A funcionalidade "Detectar possíveis fraudes com o máximo de informações" permite ao gestor de frota ter as informações precisas e em tempo real ao seu favor, de forma a

permitir que o mesmo audite a frota e seus custos.

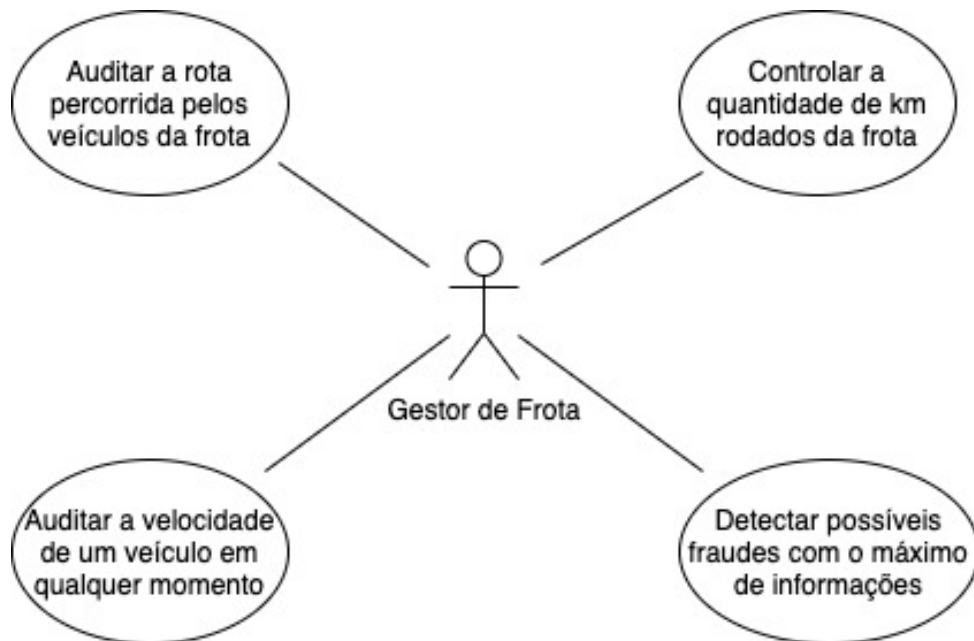


Figura 3.1 – Diagrama de casos de uso.

3.1.2 Diagrama de atividades

Afim de descrever as etapas da coleta de dados pelo aplicativo foi elaborado um diagrama de atividades. As etapas da coleta descritas no diagrama são:

Coleta dados GPS: Quando o aplicativo solicita a API sistema operacional do *smartphone* a latitude e longitude capturadas pelos sensores de GPS.

Verifica conexão Bluetooth com o Adaptador: Verifica se o aplicativo tem conexão aberta com o módulo ELM327 por meio do Bluetooth.

Escreve comando AT Serial na conexão Bluetooth: Momento em que o aplicativo envia ao adaptador ELM327 um comando.

Solicita no conexão: Ocorre quando a conexão com o adaptador ELM327 foi perdida para abrir uma nova conexão.

Interpreta comando, requisita à ECU e retorna o dado na interface serial por Bluetooth: Etapa realizada pelo adaptador ELM327 quando recebe um novo comando.

Interpreta o valor retornado e classifica conforme cabeçalho: Etapa quando o adaptador ELM327 responde ao comando dado, sendo necessário verificar o cabeçalho da resposta que descreve o comando interpretado pelo adaptador e então verificar se interpretado corretamente.

Aplica a função conforme a classificação do dado convertendo de hexadecimal para

o formato correto: Etapa onde se converte os dados recebidos para o formato adequado, como exemplo graus celsius, porcentagem e decimal.

Armazena o valor obtido com a conversão: Momento em que é salvo em estrutura de dados da memória primária os dados convertidos.

Se todos os comandos desta rodada de monitoramento foram realizados: Verifica se todas as leituras de dados foram feitas e, se todas realizadas, segue adiante, senão retorna ao início.

Monta JSON com todos os dados da rodada: Etapa que cria um string JSON com todos os dados da rodada e do *buffer* para futuro envio.

Salva dados em arquivo de backup: Antes de enviar os dados, abre o arquivo de backup dos dados e acrescenta ao final a string JSON criada.

Envia para o o servidor por POST HTTP: Faz uma requisição HTTP do tipo POST e envia a string JSON ao servidor.

Recebe JSON de dados: Momento em que o servidor recebe os dados enviados.

Salva em banco de dados: Quando o servidor salva os dados recebidos.

Retorna o resultado da operação: Quando o servidor retorna a confirmação.

Verifica se envio teve sucesso: Verifica a resposta do servidor para decidir se houve sucesso. Se houve sucesso, retorna ao início.

Adiciona ao *buffer* para envio na próxima rodada: Caso o envio de dados ao servidor tenha falhado, armazena em estrutura de dados da memória primária os dados para ser enviado novamente na próxima rodada.

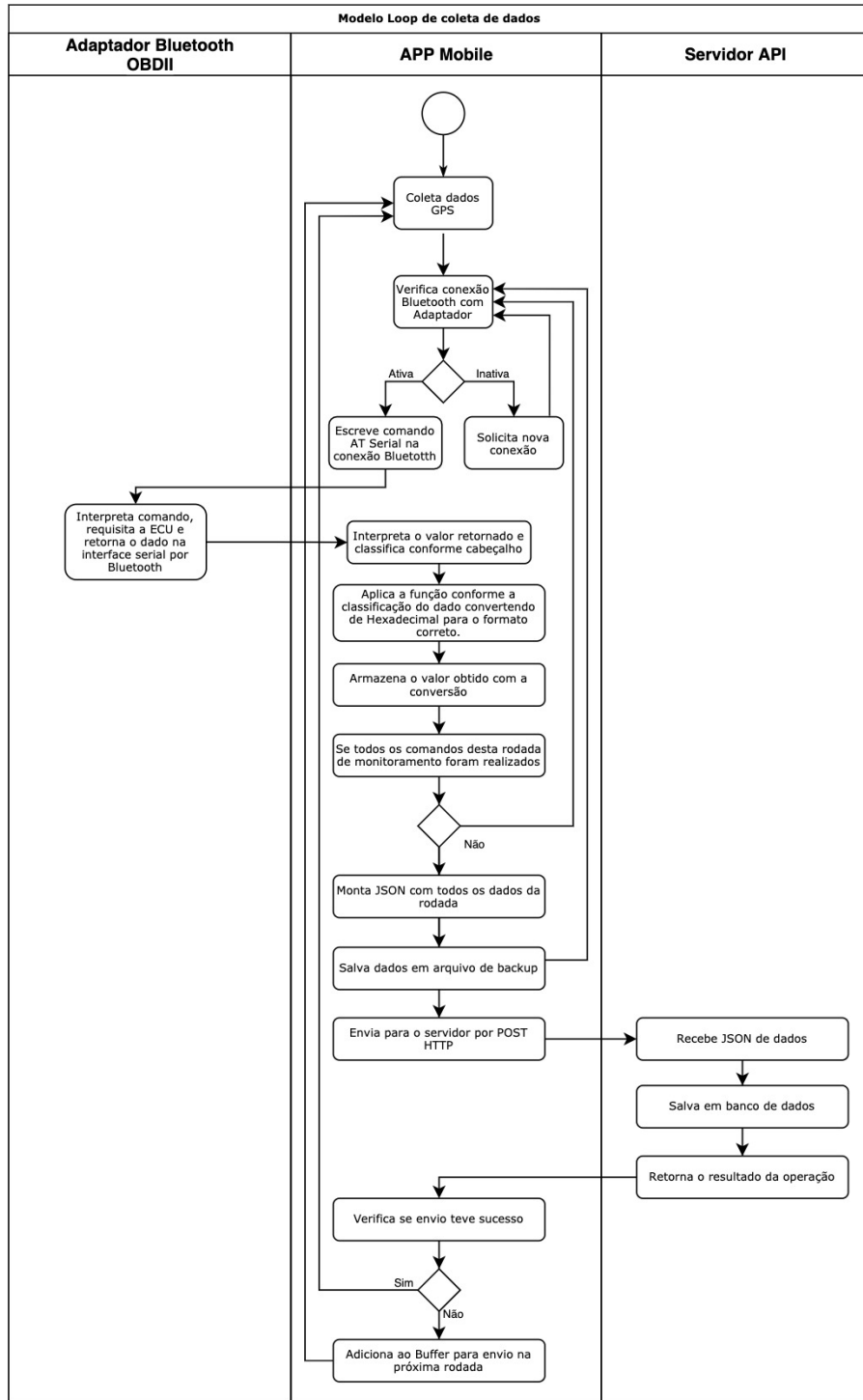


Figura 3.2 – Diagrama de atividade para ciclo de coleta de dados.

3.2 IMPLEMENTAÇÃO

Esta seção descreve as ferramentas utilizadas na construção do aplicativo móvel.

3.2.1 Ferramentas utilizadas

O aplicativo *mobile* foi desenvolvido em linguagem JavaScript ECMA6. No servidor, a linguagem escolhida também foi a mesma, porém com o uso do ambiente de execução Javascript server-side Node.JS.

Para o desenvolvimento do aplicativo em *smartphone* android foi utilizado o *framework* React Native na versão 0.63.4, o qual é capaz de compilar aplicativos nativos para os sistemas operacionais Android e iOS. O foco deste trabalho está em desenvolver somente aplicativos Android, apesar da possibilidade de desenvolvimento de aplicativos para sistema iOS Apple pelo *framework* React Native (FACEBOOK, 2021).

Para a comunicação Bluetooth foi utilizado o plugin React Native Bluetooth Serial Next na versão 1.2.3. A documentação do mesmo encontra-se presente no repositório Git público do mesmo (NUTTAWUTMALEE, 2019).

Para a recuperação dos dados de localização do dispositivo foi utilizado o *plugin* react-native-community/geolocation na versão 2.0.2. A documentação do mesmo encontra-se presente no repositório Git público do mesmo (COMMUNITY, 2020).

O servidor foi implementado usando o *framework* Express na versão 4.17.1, sendo que apesar de não estar no lado do cliente, sua linguagem é também JavaScript, o que é permitido pelo uso do ambiente de execução Javascript server-side Node.JS (HOLLOWAY-CHUK; STRONGLOOP et al., 2021).

Para a persistência dos dados foi utilizado o banco de dados NoSQL MongoDB, que também é implementado em JavaScript. A versão utilizada foi a 4.4.3 (ENTERPRISE; COMMUNITY, 2021).

4 AVALIAÇÃO DOS RESULTADOS

Neste capítulo são descritos os procedimentos realizados durante os testes e o conhecimento obtido.

4.1 COLETA DOS DADOS PELO SENSOR

A coleta dos dados ocorreu em um intervalo de 10 segundos com uma ótima precisão. Algumas dificuldades foram encontradas com relação a forma de comunicação com o adaptador ELM327, sendo necessário o envio de um caracter *Carriage Return* ao término da string em Hexadecimal via interface serial para a correta interpretação. O caracter de *Line Feed* pode ser enviado em conjunto, mas o mesmo não é necessário para que o adaptador entenda o fim do comando, como demonstrado a seguir.

```
1 this.state.deviceOBD.write('ATZ\r');
```

Listing 4.1: Envio de comando PID com *Carriage Return*

Outra questão importante na forma de comunicação com o adaptador, porém esta já conhecida por estar explícita na documentação do fabricante, foi a necessidade de requisitar um único comando por vez, ou seja, cada dado a ser coletado é requisitado e respondido de forma sequencial e o tempo de resposta é variável. Diversos comandos são possíveis, porém cada fabricante opta por aceitar ou não cada comando, sendo que muitos fabricantes não implementam os comandos do protocolo em sua totalidade, no entanto colocam esses em outros PIDs para consulta.

Essa customização do protocolo realizada pelos fabricantes não é publicada para a comunidade, sendo que muitos dos fabricantes desejam que uma parceria comercial seja realizada para então haver a troca de conhecimento. Para este trabalho em especial, os comandos faltantes foram substituídos por comandos similares que eram mais comumente encontrados. Um exemplo foi o comando que retorna a quilometragem do odômetro do veículo. Este comando não era implementado em nenhum dos veículos testados, porém existe um comando que retorna a quilometragem desde o último *reset* da MIL, o que é feito geralmente durante as manutenções. A partir desta quilometragem pode-se calibrar a quilometragem percorrida desde a fabricação acrescentando a diferença quando comparada com a quilometragem do odômetro. No entanto, para este caso não houve a necessidade de calibragem, pois bastou efetuar a diferença desta quilometragem entre o seu término em relação ao seu início na rota monitorada.

Ainda, existem formas de se descobrir esses comandos efetuando uma análise dos comandos trafegados no barramento OBDII da ECU veicular, porém esta técnica é

repleta de incertezas e possui uma grande complexidade na análise. Uma outra forma é a análise de outros aplicativos gratuitos capazes de buscar estes dados, monitorando sua conexão ao adaptador ELM327, de forma a simular uma conexão Bluetooth. Esta técnica é mais simples, porém é necessário um estudo sobre a possível violação dos direitos de uso destes aplicativos. Como o escopo deste trabalho não visa tal aprofundamento, os requisitos foram atendidos com outros comandos similares presentes.

Os dados coletados em tempo real da ECU foram: a versão do protocolo ISO praticado pelo veículo, estado da conexão ao adaptador, RPM do motor, velocidade do veículo, odômetro, nível do tanque de combustível e porcentagem de álcool presente na gasolina. Ainda foram coletados os dados de GPS através do sensor do *smartphone*, sendo eles a latitude e a longitude. Cabe destacar que o comando que retorna a porcentagem de álcool na gasolina estava presente em um dos dois veículos testados e o nível do combustível não era implementado em nenhum, ficando estes comandos configurados e preparados para possíveis novos testes em outros veículos.

Os teste ocorreram em quatro veículos com interface OBDII, sendo eles um Nissan March 1.0 2016, uma Chevrolet Montana 1.6 2016, um Honda Fit 1.5 2009 e uma Ford Ecosport 1.6 2004. Em todos os carros a coleta de dados ocorreu de forma constante e sem falhas na comunicação, porém houveram dois problemas.

4.1.1 Problema 1

A primeira dificuldade encontrada foi com relação a função de conversão dos dados de odômetro, que são recebidos em formato Hexadecimal como String, sendo que a linha da conversão acabava fazendo somente um Parse desta String considerando como se o dado coletado estivesse já em base decimal, como pode se ver abaixo a forma errada e sua correção.

```
1 // exemplo de fragment: "41 31 21 B3"
2 let odometer = fragment.slice(6, 15).replace(/\n|\r| /gi, '');
3 odometer = parseInt(odometer, 10);
4 // retorna 21
```

Listing 4.2: Forma errada de conversão.

```
1 // exemplo de fragment: "41 31 21 B3"
2 let odometer = fragment.slice(6, 15).replace(/\n|\r| /gi, '');
3 odometer = parseInt(odometer, 16);
4 // retorna 8627
```

Listing 4.3: Forma errada de conversão.

Infelizmente, este erro de digitação foi notado após vários percursos monitorados, porém sendo corrigido e testado com sucesso, mas as coletas no veículo Chevrolet Montana 1.6 não puderam serem refeitos, apesar de que o problema enfrentado estava somente relacionado a forma de conversão e não com problemas relacionados a falhas. As rotas monitoradas nos demais veículo foram refeitas e o problema não ocorreu novamente.

Como estes dados de odômetro coletados foram enviados ao servidor e salvos no arquivo de backup somente em sua forma convertida, impacta que não existe uma forma de recuperá-los. A função `parseInt` do JavaScript, que recebe uma String e retorna um Number é configurada para conversão para decimal, considera somente os algarismo de 0 à 9, ignorando as letras que compõem a base hexadecimal. Um exemplo pode ser visto abaixo:

```
1 let odometer = parseInt('21A9', 10);
2 // retorna 21
```

Listing 4.4: Demonstração do retorno da função `parseInt`

Apesar deste infortúnio, os demais dados foram corretamente coletados. Os dados foram coletados, salvos em backup de arquivo e enviados ao servidor em formato JSON. Um exemplo de um momento de salvamento é exemplificado a seguir:

```

1 {
2   "collectTimestamp": "2021-02-05T01:10:48.038Z",
3   "bluetoothIsEnable": true,
4   "carDescription": "MARCH1.02016\r",
5   "elmVersion": "ELM327 v2.1\r",
6   "isoProtocol": "ISO 15765-4 (CAN 11/500)\r",
7   "connectObd": true,
8   "egnitionStatus": 173,
9   "fuelLevel": null,
10  "currentLatitude": "-29.7183354",
11  "currentLongitude": "-53.8082999",
12  "rpm": {
13    "value": 1344.5,
14    "lastRead": "2021-02-05T01:10:38.088Z"
15  },
16  "speed": {
17    "value": 15,
18    "lastRead": "2021-02-05T01:10:38.797Z"
19  },
20  "odometer": {
21    "value": 8627,
22    "lastRead": "2021-02-05T01:10:39.560Z"
23  },
24  "throttlePos": {
25    "value": "18.82",
26    "lastRead": "2021-02-05T01:10:40.260Z"
27  },
28  "ethanolPercent": {
29    "value": "21.18",
30    "lastRead": "2021-02-05T01:10:40.948Z"
31  }
32 }

```

Listing 4.5: Desmotração do retorno da função parseInt

4.1.2 Problema 2

O segundo problema encontrado foram nos teste efetuados no veículo Ford Ecosport 1.6 de ano 2004. Este veículo possui ano muito abaixo do ano que se tornou obrigatório a presença de interface OBD2 nos veículos brasileiros, a saber, ano de 2009. Apesar deste veículo implementar a versão nova do procolo OBD, o mesmo não reconheceu o

comando utilizado para monitorar o odômetro. Como este trabalho não previa que veículos anteriores a 2009 fossem compatíveis, este problema não afeta seus resultados.

4.1.3 Veículos Flex

No desenvolvimento do aplicativo foi adicionado o comando que retorna a porcentagem de álcool presente na gasolina, sendo que este indicador é somente possível em veículos flex, pois nestes modelos é muito importante que a ECU conheça esse dado para então mensurar a quantidade de combustível a ser injetada no motor. Nos veículos testados este dado foi reconhecido por somente um dos modelos, Nissan March. Este comando foi adicionado de forma extra ao projeto do aplicativo, pois o mesmo não era necessário em nenhum dos casos de uso. Desta forma, este não foi considerado como sendo um problema que afete os resultados.

4.1.4 Aplicativo móvel

O aplicativo desenvolvido para efetuar a coleta e envio dos dados, desenvolvido em React Native, foi compilado para o sistema Android 7.1.1 e utilizando em aparelho Motorola G4 Play, que apesar de ser um modelo antigo produzido a partir do ano de 2016, se demonstrou compatível com todas as funcionalidades utilizadas no código fonte. Sua interface não foi desenvolvida com intuito de ser esteticamente agradável, mas sim ser um protótipo capaz de demonstrar as potencialidades deste trabalho. Apesar disso, é possível que trabalhos futuros possam estilizar e melhorar a aparência. Uma demonstração do aplicativo em funcionamento por ser vista a seguir:

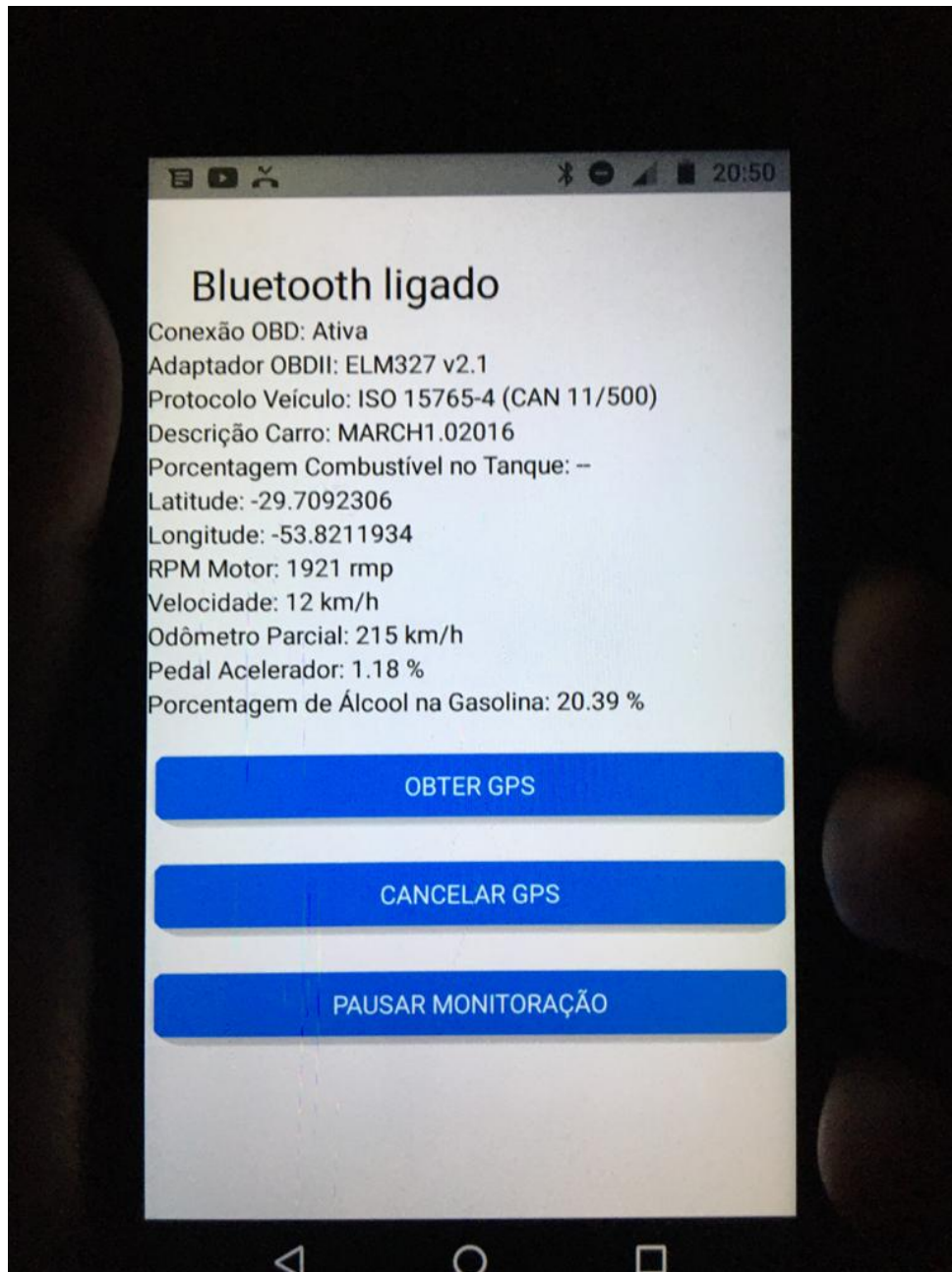


Figura 4.1 – Aplicativo React Native em execução.

Também foram feitos testes em um dispositivo *smartphone* Apple modelo Iphone 6s. O aparelho possui incompatibilidade com a versão Bluetooth do adaptador ELM327, o que impossibilitou o pareamento e comunicação dos mesmos. Apesar da comunicação com o adaptador Bluetooth não ocorrer, o aplicativo compilou e executou sem erros e foi capaz de coletar os dados de localização geográfica e de efetuar a comunicação com um computador por Bluetooth, sendo monitorado no decorrer dos testes. É possível que modelos de versão superior ao *smartphone* Apple usado sejam compatíveis com o adaptador ELM327.

Algumas fotos dos veículos estão demonstradas a seguir:



Figura 4.2 – Adaptador ELM327 Bluetooth conectado ao Nissan March.



Figura 4.3 – Adaptador ELM327 Bluetooth conectado Chevrolet Montana.

4.2 ENVIO DOS DADOS E PERSISTÊNCIA PELA API

O envio de dados para o servidor ocorreu por meio da conexão de Internet do dispositivo *smartphone* de forma satisfatória, de forma que em momentos que não era possível a conexão de Internet móvel, o aplicativo armazenou os dados não enviados em um buffer e no retorno da conexão os dados foram efetivamente enviados. Como sabe-se que existem trechos viários que a conexão móvel pode falhar ou mesmo não existir, esse armazenamento temporário dos dados foi previamente planejado. Todavia, esses dados são persistidos em arquivo no próprio dispositivo para possíveis coletas.

O servidor desenvolvido além de conter uma API de comunicação para os dados enviados pelo aparelho móvel conta com uma interface web para a exibição dos dados. Os dados processados de cada rota são exibidos em uma tabela com suas médias ou diferenças. Também quando o usuário clica nas linhas de cada rota é redirecionado para outra página onde o trajeto do veículo é renderizado para análise.

Como explicado, o problema enfrentado pelo erro do código fonte inviabilizou o cálculo do percurso percorrido pelo veículo Chevrolet Montana, e apesar de ter sido corrigido, as rotas não puderam serem refeitas por se tratar de um veículo cedido temporariamente aos testes. A seguir podem ser vistos os resultados obtidos e interfaces do sistema web desenvolvido. Vale lembrar que as rotas do segundo veículo testado possuem seu cálculo de distância percorrida afetado pelo problema no código. As rotas do primeiro veículo puderam serem refeitas e comprovaram a correção do erro de conversão dos dados.

Visualiza dados

Rota	Veículo	Horário de Início	Velocidade Média	RPM Médio	Tempo de Percurso	Distância Percorrida	Álcool na gasolina
1	Nissan March 1.0 2016	4/1/2021 21:43	30.1 km/h	1567 rpm	6 min	3 km	20.61 %
2	Nissan March 1.0 2016	4/1/2021 22:4	28.1 km/h	1461 rpm	6 min	3 km	20.57 %
3	Honda Fit 1.5 2009	6/1/2021 14:23	23.8 km/h	1436 rpm	7 min	2 km	0.00 %
4	Ford EcoSport 1.6 2004	6/1/2021 16:57	18.5 km/h	899 rpm	5 min	0 km	0.00 %
5	Ford EcoSport 1.6 2004	6/1/2021 17:3	27.7 km/h	1371 rpm	4 min	0 km	0.00 %
6	Chevrolet Montana 1.6 2010	2/1/2021 8:33	17.4 km/h	1238 rpm	7 min	0 km	0.00 %
7	Chevrolet Montana 1.6 2010	2/1/2021 8:42	12.2 km/h	829 rpm	58 min	0 km	0.00 %
8	Chevrolet Montana 1.6 2010	2/1/2021 10:28	29.3 km/h	1665 rpm	15 min	0 km	0.00 %
9	Chevrolet Montana 1.6 2010	2/1/2021 12:20	13.9 km/h	1251 rpm	6 min	0 km	0.00 %
10	Chevrolet Montana 1.6 2010	2/1/2021 14:27	19.8 km/h	1423 rpm	4 min	0 km	0.00 %
11	Chevrolet Montana 1.6 2010	2/1/2021 14:46	21.7 km/h	1433 rpm	4 min	0 km	0.00 %
12	Chevrolet Montana 1.6 2010	3/1/2021 8:34	19.5 km/h	1386 rpm	19 min	0 km	0.00 %
13	Chevrolet Montana 1.6 2010	3/1/2021 9:6	31.2 km/h	1581 rpm	6 min	0 km	0.00 %

Figura 4.4 – Tela de visualização dos dados.

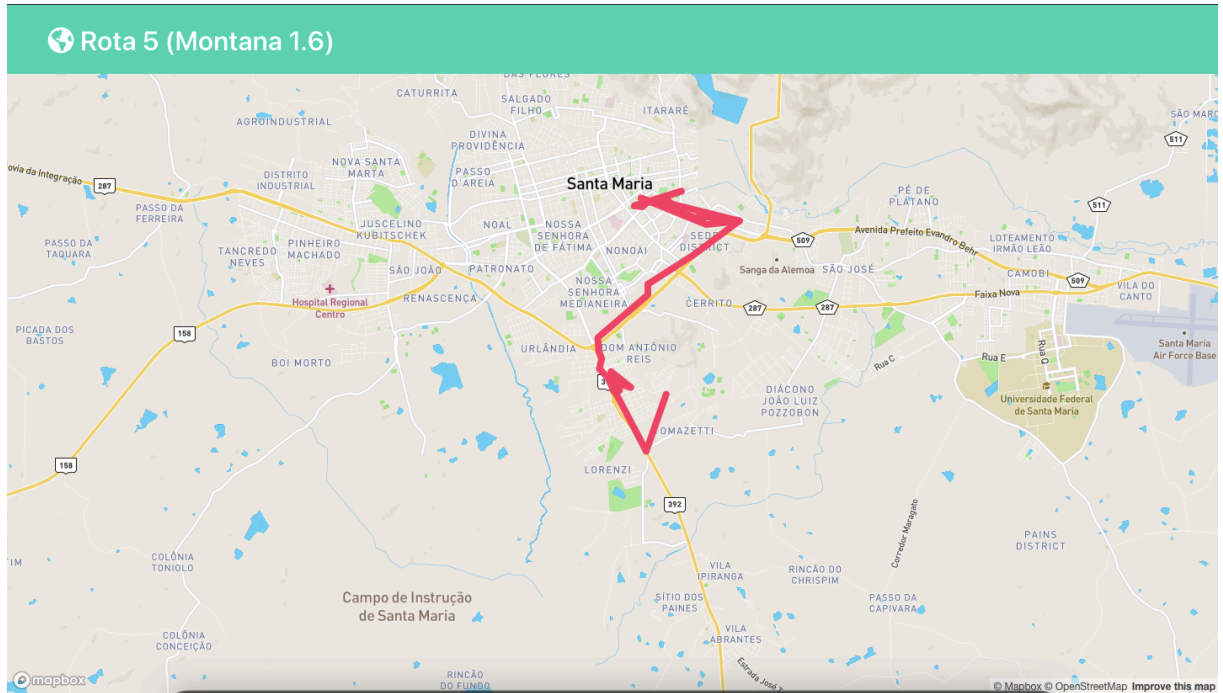


Figura 4.5 – Tela de visualização da rota.

Como durante os testes ficou claro que nem todos os comandos eram implementados por todas as montadoras de veículo, alguns comandos adicionais incluídos na implementação tiveram seu retorno como sendo NO DATA. Estes casos foram tratados e os comandos foram salvos e enviados como sendo do tipo null. O comando que deveria retornar a porcentagem de combustível na tanque não foi reconhecido em nenhum dos modelos de carros testados e o comando de porcentagem de álcool misturado à gasolina foi reconhecido somente no veículo Nissan March 1.0.

5 CONCLUSÃO

As novas tecnologias presentes no nosso dia a dia nos permitem a coleta de diversos dados que podem ser processados e analisados para os mais diferentes fins. A tecnologia IoT vem conquistando cada vez mais espaço do mercado e possui um grande potencial de crescimento, permitindo que diversos equipamentos ou objetos possam ser monitorados remotamente através dos sensores presentes ou mesmo serem controlados através de atuadores.

O presente trabalho se propôs a coletar esses dados veiculares para permitir ao gestor de frota uma maior precisão no seu dia a dia e também criticar os dados coletados por fontes humanas. O objetivo foi alcançado com a coleta de dados de forma remota. Os algoritmos desenvolvidos podem ser melhorados a fim de se obter outros dados pertinentes para diversos fins em futuros trabalhos.

Algumas dificuldades foram encontradas no decorrer do trabalho, porém todas puderam ser superadas de forma a se alcançar o objetivo proposto.

Espera-se que novos veículos desenvolvidos passem em um futuro não tão distante a terem novas formas para se coletar dados que sejam mais simples e democráticas, no entanto fica concluído que já é possível coletar estes dados atualmente.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, E. L. de; FARIA, F. F. de. **Scanner OBD-II em Plataforma LabVIEW**. 2013. Monografia (Tecnologia em Eletrônica Automotiva), Faculdade de Tecnologia Fatec Santo André, São Paulo - SP. Disponível em: <<http://fatecsantoandre.edu.br/arquivos/TCC232.pdf>>.

ASHTON, K. That "internet of things" thing. **RFiD Journal**, v. 22, n. 7, p. 97–114, 2009.

BERTAGLIA, P. R. **Logística e gerenciamento da cadeia de abastecimento**. São Paulo: Saraiva Educação, 2020. 23, 276-302 p.

COMMUNITY react-native. **@react-native-community/geolocation**. 2020. Acesso em Jan. 2021. Disponível em: <<https://github.com/react-native-geolocation/react-native-geolocation>>.

CONAMA. **Dispõe sobre os requisitos para adoção de sistemas de diagnose de bordo - OBD nos veículos automotores leves objetivando preservar a funcionalidade dos sistemas de controle de emissão**. [S.l.], 2004. Resolução 354, 62,63 p. Diário Oficial da União, Ministério do Meio Ambiente, Brasília, nº 239, de 14 dez. 2004. Seção 1, p. 62-63. Disponível em: <<http://www2.mma.gov.br/port/conama/legiabre.cfm?codlegi=456>>.

CRUZ, F. **Python: Escreva seus primeiros programas**. São Paulo: Casa do Código, 2015. 1-9 p.

ELECTRONICS, E. **ELM327: OBD to RS232 Interpreter**. [S.l.], 2016. Acesso em: mai. 2020. Disponível em: <<https://www.elmelectronics.com/wp-content/uploads/2016/07/ELM327DS.pdf>>.

ENTERPRISE, M.; COMMUNITY. **MongoDB Community Server**. 2021. Acesso em Jan. 2021. Disponível em: <<https://docs.mongodb.com/manual/>>.

FACEBOOK, I. **React Native**. 2021. Acesso em Jan. 2021. Disponível em: <<https://reactnative.dev/docs/getting-started>>.

FAGUNDES, F. A. V.; SILVA, G. L. da; ASSIS, M. A. S. **Sistema de monitoramento automotivo remoto**. 2015. Monografia (Tecnologia em Eletrônica Automotiva), Faculdade de Tecnologia Fatec Santo André, São Paulo - SP. Disponível em: <<http://fatecsantoandre.edu.br/arquivos/TCC342.pdf>>.

FLANAGAN, D. **JavaScript : o guia definitivo**. Porto Alegre: Bookman, 2013. 1-4 p.

HOLLOWAYCHUK, T.; STRONGLOOP; OTHERS. **Express.js**. 2021. Acesso em Jan. 2021. Disponível em: <<https://expressjs.com/pt-br/4x/api.html>>.

ITU-T. **Overview of the Internet of things**. [S.l.], 2012. Recomendação Y.2060. Acesso em: mar. 2020. Disponível em: <<https://www.itu.int/en/ITU-T/publications/Pages/recs.aspx>>.

KLEINSCHMIDT, J. H. **REDES BLUETOOTH: MODELAGEM, DESEMPENHO E APLICAÇÕES**. 2004. Dissertação (Mestrado em Informática Aplicada), Pontifícia Universidade Católica do Paraná, Curitiba - PR. Disponível em: <https://www.pggia.pucpr.br/pt/arquivos/mestrado/dissertacoes/2004/2004_joakleinchmidt.pdf>.

MACHADO, A. S. L.; OLIVEIRA, B. R. R. O sistema obd (on-board diagnosis). **Mestrado em Automação e Sistemas, Instituto Superior de Engenharia do Porto, Porto - Portugal**, p. 97–114, 2007. Acesso em Nov. 2020. Disponível em: <http://ave.dee.isep.ipp.pt/~mjf/act_lect/SIAUT/Trabalhos\%202007-08/Trabalhos/SIAUT_OBD.pdf>.

MAGRANI, E. **A internet das coisas**. Rio de Janeiro: FGV Editora, 2018. 192 p.

MINISTÉRIO DA INFRAESTRUTURA. **Divulgação de dados estatísticos sobre a frota nacional de veículos**. 2020. Acesso em Nov. 2020. Disponível em: <<https://www.gov.br/infraestrutura/pt-br/assuntos/transito/conteudo-denatran/frota-de-veiculos-2020>>.

NUTTAWUTMALEE. **react-native-bluetooth-serial-next**. 2019. Acesso em Jan. 2021. Disponível em: <<https://github.com/nuttawutmalee/react-native-bluetooth-serial-next>>.

OLIVEIRA, J. C. M. G. de. **Soft sensor veicular para medição de emissões de gases**. 2017. Dissertação (Mestrado em Engenharia Elétrica e de Computação), Universidade Federal do Rio Grande do Norte, Natal - RN. Disponível em: <https://repositorio.ufrn.br/jspui/bitstream/123456789/24834/1/JulioCesarMeloGomesDeOliveira_DISSERT.pdf>.

PERKOVIC, L. **Introdução à computação usando Python : um foco no desenvolvimento de aplicação**. Rio de Janeiro: LTC Livros Técnicos e Científicos Editora Ltda, 2016. 1-95 p.

SINDIPEÇAS, S. N. da Indústria de Componentes para V. A.; ABIPEÇAS, A. B. da Indústria de A. **Relatório da Frota Circulante**. [S.l.], 2019. Disponível em: <<https://abinteligencia.com.br/relatorio-da-frota-circulante-2020/>>.

VALENTE, A. M. et al. **Gerenciamento de Transporte e Frotas**. São Paulo: Cengage Learning, 2016. 1-36 p.