

UNIVERSIDADE DE CAXIAS DO SUL

LUCAS DE FARIA

**PROTÓTIPO PARA COLETA DE INFORMAÇÕES DA REDE CAN EM
AUTOMÓVEIS**

**Bento Gonçalves
2021**

Lucas de Faria

**PROTÓTIPO PARA COLETA DE INFORMAÇÕES DA REDE CAN EM
AUTOMÓVEIS**

Trabalho de Conclusão de Curso apresentado
como parte dos requisitos para obtenção do
título de Engenheiro Eletrônico.

Orientador:
Prof. Me. Ricardo Becker

**Bento Gonçalves
2021**

Ficha catalográfica.

DE FARIA, Lucas

Protótipo para coleta de informações da rede CAN em automóveis / Lucas de Faria; orientador: Ricardo Becker, Caxias do Sul: UCS, Área do Conhecimento de Ciências Exatas e Engenharia, 2021.

82 folhas, 30 cm.

Trabalho de conclusão do curso de Engenharia Eletrônica — Universidade de Caxias do Sul, Área do Conhecimento de Ciências Exatas e Engenharia, 2021

Inclui referências bibliográficas.

1. OBDII. 2. Google *Data Studio*. 3. *dashboard*. 4. Gestão de Frotas. I. Becker, Ricardo. II. Universidade de Caxias do Sul. III. Protótipo para coleta de informações da rede CAN em automóveis.

Lucas de Faria

**PROTÓTIPO PARA COLETA DE INFORMAÇÕES DA REDE CAN EM
AUTOMÓVEIS**

Trabalho de Conclusão de Curso apresentado
como parte dos requisitos para obtenção do
título de Engenheiro Eletrônico.

Orientador:
Prof. Me. Ricardo Becker

Aprovado em ____/____/____

Banca Examinadora

Prof. Me. Ricardo Becker (orientador)
Universidade de Caxias do Sul - UCS

Prof. Me. Patric Janner Marques
Universidade de Caxias do Sul - UCS

Prof. Me. Cesar Augusto Bernardi
Universidade de Caxias do Sul - UCS

Dedico esse trabalho aos meus pais e minha namorada que sempre me apoiaram e incentivaram em todos os momentos da jornada acadêmica.

AGRADECIMENTOS

À minha família, por me apoiar desde sempre e por compreender os momentos distantes em prol de um objetivo maior.

À Morgana, minha namorada, por todo o apoio e compreensão nessa fase final da jornada acadêmica.

Ao Professor Mestre Ricardo Becker pela orientação, incentivo e compreensão, juntamente com todo o aprendizado acolhido.

Aos colegas de graduação pela disposição e auxílio durante todo este processo.

“Um homem deve se ajustar a um novo objetivo assim que o antigo cai por terra.”
Sir Ernest Shackleton.

RESUMO

Dentro do mercado de frotas veiculares, cada vez mais os gestores têm reconhecido a importância de uma gestão de frotas eficaz e seus impactos nos resultados da sua empresa. A gestão de frotas veiculares se baseia no controle e análise das informações gerada pelos veículos durante o uso, e no comportamento dos motoristas na operação dos mesmos. Para realizar o acesso a esses dados, existem diversos dispositivos e serviços disponíveis no mercado para coleta de informações e apontamento de falhas. São encontrados dispositivos de *hardware* complexos para uso profissional em mecânicas automotivas, até serviços com supervisórios e aplicativos de *smartphone* para empresas com frotas de veículos. Dentro deste contexto, este trabalho apresenta o desenvolvimento de um sistema de coleta de informações do automóvel e disponibilização *online* dos dados. O objetivo principal deste trabalho, é detectar eventos adversos na condução, realizando a leitura das informações disponíveis na rede de comunicação de dados veicular como velocidade, RPM e temperatura do motor.

Com base na bibliografia, foi desenvolvido um protótipo capaz de acessar estas informações via porta OBDII do veículo, através do envio de pacotes contendo os códigos PID de interesse. Ao receber a resposta, verifica se o valor está dentro da faixa definida como limite para aquele parâmetro. Caso o valor ultrapasse o limite, é realizado o registro em um arquivo de texto. Através da conexão *Wi-Fi*, os dados presentes no arquivo de texto são publicados em uma planilha *online*. Por fim, utilizando o Google *Data Studio*, é feita a visualização dos dados em forma de *dashboard*. Para a validação do protótipo, foi realizada a comparação com os resultados obtidos através de ferramentas comerciais de proposta similares. O protótipo se mostrou eficaz na coleta e publicação das informações do veículo em comparação com aplicativos para *smartphone*, demonstrando ser capaz de obter o mesmo nível de informações deste tipo de solução. Comparando o *dashboard* desenvolvido com soluções comerciais já conhecidas no mercado, a solução proposta neste trabalho, como ferramenta gratuita, demonstrou ser capaz de fornecer informações semelhantes as obtidas no uso destes serviços consolidados.

Palavras-chave: OBDII. Google *Data Studio*. *dashboard*. Gestão de Frotas.

ABSTRACT

Within the vehicle fleet market, managers have increasingly recognized the importance of effective fleet management and its impact on their company's results. Vehicle fleet management is based on the control and analysis of information generated by vehicles during use and on the behavior of drivers in their operation. To access this data, there are several devices and services available on the market for collecting information and pointing out failures. Complex hardware devices are found for professional use in automotive mechanics, to supervisory services and smartphone applications for companies with vehicle fleets. Within this context, this work presents the development of a system for collecting information about the automobile and making available online of the data. The main objective of this work is to detect adverse events during driving, reading the information available in the vehicle data communication network, such as speed, RPM and engine temperature.

Based on the bibliography, a prototype capable of accessing this information via the vehicle's OBDII port was developed, by sending packages containing the PID codes of interest. Upon receiving the response, it checks whether the value is within the range defined as the limit for that parameter. If the value exceeds the limit, it is recorded in a text file. Through the *Wi-Fi* connection, the data present in the text file is published in a *online* spreadsheet. Finally, using Google *Data Studio*, data is visualized in the form of *dashboard*. To validate the prototype, a comparison was made with the results obtained using commercial tools with similar proposals. The prototype proved to be effective in collecting and publishing vehicle information compared to applications for *smartphone*, demonstrating to be able to obtain the same level of information from this type of solution. Comparing the *dashboard* developed with commercial solutions already known in the market, the solution proposed in this work, as a free tool, proved to be capable of providing information similar to that obtained using these consolidated services.

Keywords: OBDII. Google Data Studio. dashboard. Fleet management.

LISTA DE FIGURAS

Figura 1 – Visão genérica do gerenciamento em um veículo FIAT.	20
Figura 2 – Bits dominantes e recessivos barramento CAN.	21
Figura 3 – Quadro CAN 2.0A e 2.0B.	22
Figura 4 – Formato quadro de mensagens CAN 2.0A.	22
Figura 5 – Estrutura do código de falha DTC.	24
Figura 6 – Estrutura Mensagem de Pedido.	25
Figura 7 – Estrutura Mensagem de Reposta.	25
Figura 8 – Camada física de acordo com a norma ISO-11898-2.	27
Figura 9 – Camadas TCP/IP.	31
Figura 10 – Supervisor em <i>Labview</i> para visualização de parâmetros.	36
Figura 11 – Gráfico de relação entre Velocidade e Rotação do motor.	37
Figura 12 – Gráfico da relação entre a variação do RPM e desvio da aceleração longitudinal.	38
Figura 13 – Distribuição de clientes de serviços de gerenciamento de frotas.	40
Figura 14 – Tamanho dos clientes.	40
Figura 15 – Tipo de veículos com serviços de monitoramento.	41
Figura 16 – Diagrama da arquitetura proposta.	44
Figura 17 – Diagrama Etapa 1.	48
Figura 18 – Circuito para interface entre o microcontrolador e porta OBDII do veículo.	49
Figura 19 – Fluxograma do algoritmo de validação da detecção de evento e escrita no SD <i>Card</i>	54
Figura 20 – Fluxograma da Etapa 3.	55
Figura 21 – Compartilhamento do barramento SPI.	55
Figura 22 – Fluxograma de funcionamento do algoritmo implementado para leitura e publicação dos dados na planilha <i>online</i>	57
Figura 23 – <i>dashboard</i> desenvolvido utilizando os dados publicados no Google <i>SpreadSheet</i> como banco de dados.	59
Figura 24 – Visualização 2D do protótipo desenvolvido.	60
Figura 25 – Visualização 3D do protótipo desenvolvido.	60
Figura 26 – Protótipo montado no veículo Ford Ka Sport para teste de integração.	62
Figura 27 – Fluxograma completo do algoritmo desenvolvido para o microcontrolador principal.	62
Figura 28 – Trajeto percorrido durante os testes de validação.	64
Figura 29 – Dados coletados pelo protótipo e exibidos no <i>dashboard</i> Google <i>Data Studio</i>	66
Figura 30 – Dados coletados no RaceChorno.	67
Figura 31 – Ferramenta OBDII Autoool X60.	74
Figura 32 – <i>Script</i> desenvolvido para página <i>web</i> responsável por receber os dados do ESP8266 parte 1.	75

Figura 33 – <i>Script</i> desenvolvido para página <i>web</i> responsável por receber os dados do ESP8266 parte 2	76
Figura 34 – Conexão do protótipo com a porta OBDII.	77
Figura 35 – <i>dashboard</i> da solução Carrorama.	82

LISTA DE TABELAS

Tabela 1 – Endereços OBDII para CAN2.0A 11 <i>bits</i>	26
Tabela 2 – Descrição dos pinos conector J1962.	27
Tabela 3 – Métodos HTTP.	33
Tabela 4 – Tabela de classificação de motorista conforme troca de marcha.	38
Tabela 5 – Comparativo entre trabalhos na área.	39
Tabela 6 – Comparativo dos serviços prestados por empresas nacionais do setor.	42
Tabela 7 – Descrição dos campos contidos na mensagem NMEA-0183 RMC	50
Tabela 8 – Códigos PID disponíveis de acordo com modelo e fabricante do veículo testado.	51
Tabela 9 – Tabela de comparação entre dados GPS do <i>Smartphone</i> e protótipo	52
Tabela 10 – Resultado teste comparativo Aplicativo RaceChrono e Protótipo.	65
Tabela 11 – Códigos PID OBDII para requisições de <i>status</i>	78

LISTA DE SIGLAS

ACU	-	<i>Airbag Control Unit</i>
ACK	-	<i>Acknowledge</i>
BCM	-	<i>Body Control Module</i>
BSS	-	<i>Base Transceiver System</i>
CAN	-	<i>Controller Area Network</i>
CAN-L	-	Sinal baixo do barramento CAN
CAN-H	-	Sinal alto do barramento CAN
CCM	-	<i>Central Control Module</i>
CRC	-	<i>Cycle Redundancy Check</i>
CSV	-	<i>Comma-Separated Values</i>
DLC	-	<i>Data Length Code</i>
DTC	-	<i>Diagnostic Trouble Code</i>
ECU	-	<i>Electronic Control Unit</i>
ECM	-	<i>Electronic Control Module</i>
ETSI	-	<i>European Telecommunications Standards Institute</i>
EPROM	-	<i>Erasable Programmable Read Only Memory</i>
GPS	-	<i>Global Positioning System</i>
GEM	-	<i>General Electronic Module</i>
GSM	-	<i>Global System for Mobile</i>
GPRS	-	<i>General Packet Radio Service</i>
HA	-	<i>Hard Accelerating</i>
ID	-	Identificação
I2C	-	<i>Inter-Integrated Circuit</i>
IP	-	<i>Internet Protocol</i>
ISO	-	<i>International Organization for Standardization</i>
Mbps	-	Megabits por segundo
MEMS	-	<i>Microelectromechanical systems</i>
MS	-	<i>Mobile Station</i>
NSS	-	<i>Network Switching System</i>
NMEA	-	<i>National Marine Electronics Association</i>
OBDII	-	<i>On Board Diagnostic</i>
OMS	-	<i>Operations and Maintenance System</i>

PCM - *Powertrain Control Module*
PLMN- *Public Land Mobile Network*
PID - *Parameter Identification*
RPM - *Revoluções por minuto do motor*
RTR - *Remote Transmition Request*
SD Card - *Secure Digial Card*
SCM - *Suspension Control Module*
SC - *Soft Cruising*
SOF - *Start-of-Frame*
SAE - *Society of Automotive Engineers*
SID - *Service Identification*
SPI - *Serial Peripheral Interface*
TCM - *Transmission Control Module*
UART - *Universal Asynchronous Receiver/Transmitter*
Wi-Fi - *Wireless Fidelity*

SUMÁRIO

1	INTRODUÇÃO	16
1.1	OBJETIVO GERAL	17
1.2	OBJETIVOS ESPECÍFICOS	17
1.3	RESTRICÇÕES E RISCOS	18
1.4	DESCRIÇÃO GERAL DO TRABALHO	18
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	ECU	19
2.2	<i>CONTROLLER AREA NETWORK</i>	20
2.2.1	Protocolo CAN	20
2.2.2	Formato quadro de mensagens	21
2.3	DIAGNÓSTICO VEICULAR SOBRE O PROTOCOLO CAN	23
2.3.1	SID - <i>Service Identification</i>	23
2.3.2	Mensagem de Pedido - <i>Request Message</i>	25
2.3.3	Mensagem de Resposta - <i>Response Message</i>	25
2.3.4	Mensagem de Início de Sessão	25
2.3.5	Conector padrão OBDII	26
2.3.6	Conexão física com Barramento CAN	27
2.3.6.1	Transceptor CAN	28
2.3.6.2	Controlador CAN	28
2.4	SISTEMA DE LOCALIZAÇÃO VIA SATÉLITE	28
2.5	ENVIO DE DADOS VIA INTERNET	29
2.5.1	GSM	29
2.5.2	GPRS	30
2.5.2.1	Tarifação serviço GPRS	30
2.5.3	Redes <i>Wireless Wi-Fi</i>	30
2.5.3.1	TCP/IP	31
2.5.3.2	Cliente-Servidor	32
2.6	TRABALHOS RECENTES NA ÁREA	34
2.6.1	Aquisição de dados e visualização em tempo real	34
2.6.2	Simulação de Rede CAN via <i>software</i>	34
2.6.3	Aplicação focada no gerenciamento da frota	35
2.7	SOLUÇÕES COMERCIAIS RELACIONADAS COM O TEMA	40
3	METODOLOGIA APLICADA	43
3.1	METODOLOGIA DE PROJETO	43
3.1.1	Etapa 1 - Coleta de dados	44
3.1.2	Etapa 2 - Processamento	45
3.1.3	Etapa 3 - Envio	45
3.1.4	Etapa 4 - Análise dos dados	46
3.1.5	Testes funcionais	46
3.1.6	Testes de Integração	46
3.1.7	Testes de Validação	47
4	DESENVOLVIMENTO DO PROTÓTIPO DE COLETA DE INFORMAÇÕES ATRAVÉS DA PORTA OBDII	48

4.1	ETAPA 1 - COLETA DE DADOS	48
4.1.1	Comunicação com o veículo	48
4.1.2	Leitura e interpretação dos dados GPS	50
4.1.3	Testes Funcionais Etapa 1	50
4.2	ETAPA 2 - PROCESSAMENTO	52
4.2.1	Definição dos parâmetros máximos dos PID	53
4.2.2	Algoritmo de comparação	53
4.2.3	Testes Funcionais Etapa 2	54
4.3	ETAPA 3 - ENVIO	54
4.3.1	Registro dos dados no SD Card	54
4.3.2	Publicação dos dados no Google Spreadsheet	56
4.3.3	Testes Funcionais Etapa 3	57
4.4	ETAPA 4 - ANÁLISE DOS DADOS	57
4.4.1	Análise dos dados utilizando Google Data Studio	57
4.4.2	Testes Funcionais Etapa 4	58
4.5	DESENVOLVIMENTO DO <i>HARDWARE</i> DO PROTÓTIPO	59
4.5.1	Validação do <i>Hardware</i>	59
4.6	TESTES DE INTEGRAÇÃO	61
4.6.1	Resultados Testes de Integração	61
4.7	TESTES DE VALIDAÇÃO	63
4.7.1	Cenário de testes	64
4.7.2	Resultados dos Testes de Validação	65
5	CONSIDERAÇÕES FINAIS	68
	REFERÊNCIAS	71
	APÊNDICE A - FERRAMENTA OBDII AUTOMOTIVA AUTOOL X60	74
	APÊNDICE B - <i>SCRIPT</i> PARA PUBLICAR NA PLANILHA GOOGLE <i>SPREADSHEET</i>	75
	APÊNDICE C - CONEXÃO DO PROTÓTIPO COM O VEÍCULO	77
	ANEXO A - TABELA PID	78
	ANEXO B - <i>DASHBOARD</i> DA FERRAMENTA DE GESTÃO VEICULAR <i>CARRORAMA</i>	82

1 INTRODUÇÃO

A Associação Brasileira das Locadoras de Automóveis (ABLA) voltou a registrar crescimento na demanda por aluguel de carros por motoristas de aplicativos de transporte, como Uber, 99 Pop e Cabify. Segundo a entidade, entre abril e maio de 2020, com a intensificação da pandemia de Corona Vírus no País e as quarentenas decretadas por estados e municípios, houve redução de 80 por cento (ou 160 mil unidades) do total de 200 mil veículos que estavam dedicados aos serviços de transporte. Em julho, as locadoras detectaram aumento de 25 por cento na procura por esse tipo de aluguel (REDAÇÃO AB, 2020). A população que trabalha em veículos, como os motoristas de aplicativo, taxistas e motoristas profissionais aumentou 29,2 por cento em 2018 e chegou a 3,6 milhões, com 810 mil pessoas a mais em relação a 2017 (Cristiane Crelier, 2019). A formação de mini frotas utilizando as bases dos serviços de transporte particular tem crescido em todo o país, trazendo aos motoristas de aplicativo a possibilidade de terceirizar os serviços e aumentar os lucros.

Dentro desse contexto, cada vez mais os gestores têm reconhecido a importância de uma gestão de frotas eficaz e seus impactos nos resultados da sua empresa. Para que essa gestão seja eficiente, diversas soluções tecnológicas são desenvolvidas, a fim de oferecer melhorias na forma como as rotinas são conduzidas (Equipe Unidas, 2019). A gestão da frota de veículos envolve a administração de veículos motorizados, como carros, vans e caminhões, permitindo que as empresas que dependem do transporte no negócio minimizem os riscos associados à melhoria da eficiência do transporte, segurança, produtividade e redução dos custos gerais de transporte e pessoal (Gowda; Gopalakrishna, 2015). Se os motores desses veículos não apresentarem uma manutenção rigorosa, podem apresentar um consumo de combustível maior e, consequentemente, elevar as taxas de emissões de poluentes a níveis acima do esperado (Chardsutthi; Achariyasombat; Adsavakulchai, 2010).

Os motores de carros modernos geralmente possuem uma unidade de controle que é responsável pela otimização do desempenho e controle da maior parte do motor. Esta unidade de controle é conhecida como ECU (*Electronic Control Unit*). A aquisição de dados em veículos nasceu a quase 50 anos com a necessidade de adequar os veículos a normas de emissões de poluentes cada vez mais rígidas, mas também traz dados importantes como falhas em sistemas e componentes, e comportamentos na direção (Andria et al., 2015).

De um modo geral, existem três fontes principais de dados para a pesquisa do comportamento de dirigir: a primeira são os dados de vídeo, que geralmente usam instrumentos como gravador ou telefone celular para coletar dados de vídeo de tráfego. A segunda categoria vem principalmente dos dados do barramento CAN, incluindo a velocidade do carro, direção, freio, informações de posicionamento do GPS (*Global Positioning System*) e outros dados, que registram totalmente o estado de funcionamento do veículo. O terceiro tipo de dado são os dados coletados por sensores durante o processo de direção dos veículos (Lin et al., 2018).

Para realizar o acesso a esses dados, existem diversos dispositivos e serviços disponíveis no

mercado para coleta de informações e apontamento de falhas, desde dispositivos de *hardware* complexos para uso profissional em mecânicas automotivas, até serviços com supervisórios e aplicativos de celular para empresas com grandes frotas de veículos. Uma das opções no mercado atualmente é a da Creare Sistemas® que realiza a leitura do barramento CAN (*Controller Area Network*) e fornece ao cliente um ambiente virtual para acompanhamento dos dados através de relatórios personalizados (Creare Sistemas, 2020), outra opção de destaque é o serviço do Carrorama®, que dispõe de um pacote de planos para os donos de frotas dividido em pacote Starter e Elite, onde no primeiro tem-se acesso a informações limitadas e no plano Elite, relatórios mais completos, como monitoramento em tempo real, relatórios de consumo e rastreamento (Carrorama, 2019). No mercado de plataformas de rastreamento e telemetria, encontram-se diversas soluções que atendem a grandes frotas como as citadas anteriormente, mas que acabam inviáveis para pequenos empresários e donos de minifrotas como as formadas por motoristas de aplicativos por serem serviços pagos através de mensalidade e, com valores que variam de acordo com o número de carros por frota.

Dentro do contexto apresentado, este trabalho é inspirado no artigo *Design and Development of On-Board Diagnostic (OBD) Device for Cars* (SAWANT; MANE, 2018) e almeja propor um estudo sobre os dados disponíveis no barramento CAN acessados pela porta OBDII, de forma a fornecer uma solução simples e barata ao pequeno gestor de frota, que seja capaz de monitorar seus veículos mantendo sua manutenção em dia e seus motoristas treinados para uma condução otimizada.

1.1 OBJETIVO GERAL

O objetivo geral do trabalho é a implementação de um protótipo para coleta de informações integrado à rede de comunicação interna de veículo através da porta OBDII.

1.2 OBJETIVOS ESPECÍFICOS

Para atingir o objetivo geral, são listados os objetivos específicos do trabalho de forma a alcançar os resultados esperados:

- Implementar um algoritmo de leitura dos parâmetros de temperatura do líquido de arrefecimento, rotação do motor e velocidade através do acesso à rede CAN pela porta de comunicação OBDII, gerando o registro quando um desses parâmetros ultrapasse o valor determinado;
- Ler informações de geolocalização através de um módulo GPS para integrar as informações de latitude, longitude, hora e data coletadas, aos registros de eventos;
- Realizar o registro dos dados em um cartão de memória, e quando tenha conexão de rede *Wi-Fi* (*Wireless Fidelity*), publicar os dados salvos na memória para uma planilha

eletrônica *online* integrada com um *dashboard* na plataforma *Google Data Studio*;

- Comparar o desempenho da solução com relação a aquisição dos dados, com ferramentas similares e de baixo custo encontradas no mercado.

1.3 RESTRIÇÕES E RISCOS

Algumas restrições devem ser impostas e alguns riscos devem ser avaliados, uma vez que o desenvolvimento do trabalho deve se limitar aos objetivos anteriormente especificados. O sistema implementado não tem como finalidade a visualização em tempo real dos parâmetros coletados do veículo, somente a consulta dos dados previamente coletados. Vale ressaltar também que, o dispositivo não faz alterações nos parâmetros da ECU, limitando-se somente a realizar requisições de informações disponíveis no barramento CAN. O sistema proposto tem a aplicação restrita a veículos do tipo automóvel, com tecnologia de combustível gasolina ou flex, com ano de fabricação superior a 2010 e com unidade de controle eletrônico compatível com o protocolo OBDII. Foram realizados testes de validação em um número limitado de veículos, priorizando utilizar modelos de fabricantes distintos a fim de validar a aplicação genérica do sistema.

1.4 DESCRIÇÃO GERAL DO TRABALHO

Para a apresentação do projeto, esta documentação será fracionada na seguinte forma de estrutura:

- O segundo capítulo está constituído por uma pesquisa bibliográfica que abrange os conceitos básicos para a consolidação do trabalho desenvolvido neste projeto.
- O terceiro capítulo está constituído pela metodologia utilizada no desenvolvimento do trabalho.
- No quarto capítulo, são demonstrados o desenvolvimento, os testes funcionais de cada etapa e os testes de integração. Os testes validação e resultados do trabalho são descritos ao final deste capítulo.
- No quinto capítulo as contribuições deixadas por este documento e os respectivos trabalhos futuros, dos quais poderão surgir novos desafios acadêmicos.

2 FUNDAMENTAÇÃO TEÓRICA

Para estabelecer uma compreensão do contexto e fundamentar a metodologia utilizada durante o processo de desenvolvimento do trabalho, o presente capítulo tem por objetivo expor conceitos que englobam aspectos teóricos e práticos relacionados ao tema em análise e que fundamentam o desenvolvimento proposto na metodologia na sequência.

2.1 ECU

Ao contrário do que muitos pensam, o conceito de gerenciamento eletrônico do motor não é novo. Em 1939, já era testado o primeiro sistema de injeção direta de gasolina em aviões pela Bosch. Em 1954, era fabricado o primeiro veículo com injeção eletrônica, o Mercedes Benz 300 SL. No Brasil, o VW Gol GTI, em 1989, foi o primeiro carro com injeção eletrônica. Era um sistema analógico com um único bico injetor, mas que mudaria para sempre o rendimento dos carros (CAPELLI, 2010).

A ECU (*Electronic Control Unit*), é constituída basicamente de um microprocessador que recebe dados de sensores e realiza a comparação com parâmetros presentes na memória EPROM (*Erasable Programmable Read Only Memory*) (DIAS, Anderson, 2012). De acordo com Capelli (2010) os sensores são componentes instalados em vários pontos do veículo e servem para enviar dados para a ECU, como por exemplo, temperatura do motor, rotação, demanda do acelerador, etc. Atuadores recebem as informações da ECU e atuam no motor, alterando principalmente o volume de combustível injetado e parâmetros de ignição. Na Figura 1 são mostrados alguns exemplos de sensores e atuadores.

Em um veículo moderno, a ECU é subdividida em módulos. Em carros modernos de alta performance podem haver mais de 100 módulos de controle (Stoltzfus, Justin, 2020). Os módulos mais comuns encontrados são (Equipe Munic.io, 2018):

- PCM (*Powertrain Control Module*): Módulo de controle do trem de força;
- TCM (*Transmission Control Module*): Módulo de controle da transmissão;
- ACU (*Airbag Control Unit*): Módulo de controle do *Airbag*;
- CCM (*Central Control Module*): Módulo de controle central;
- GEM (*General Electronic Module*): Módulo eletrônico geral;
- BCM (*Body Control Module*): Módulo de controle do chassi;
- SCM (*Suspension Control Module*): Módulo de controle da suspensão;

Figura 1 – Visão genérica do gerenciamento em um veículo FIAT.



Fonte: Adaptado de (CAPELLI, 2010).

2.2 CONTROLLER AREA NETWORK

A rede CAN (*Controller Area Network*) ou área de trabalho controlada foi desenvolvida por Robert Bosch em 1986, como uma possível solução para simplificar a rede cabeada de veículos, que com o aumento de componentes eletrônicos demandavam maior espaço físico, maior custo e complexidade de projeto do chicote elétrico (CAPELLI, 2010). Por meio da rede CAN é possível interligar todos os módulos veiculares citados no item 2.1.

2.2.1 Protocolo CAN

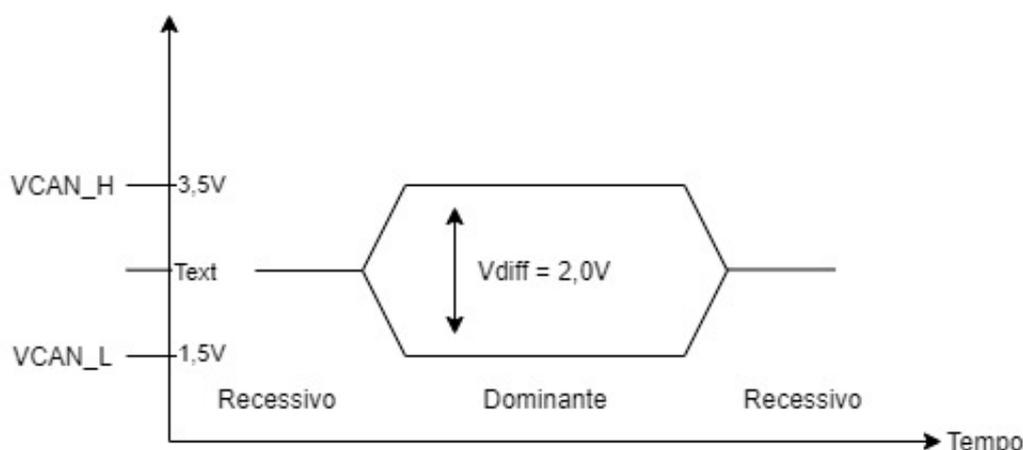
Segundo (GUIMARAES, 2007) o protocolo CAN é classificado como protocolo de comunicação serial síncrono. O sincronismo entre os módulos conectados à rede é feito em relação ao início de cada mensagem lançada ao barramento, evento esse que ocorre em intervalos de tempo conhecidos e regulares. O protocolo trabalha no conceito de multi mestre, onde todos os módulos podem se tornar mestres em determinado momento e escravos em outro e, suas mensagens são enviadas em regime *multicast*, ou seja, cada mensagem é enviada para todos os módulos existentes na rede.

Ainda conforme (GUIMARAES, 2007) cada *bit* é transmitido por um valor de tensão específico e constante. A velocidade de transmissão de dados é inversamente proporcional ao

comprimento do barramento, onde a maior taxa de transmissão é de 1 Mbps considerando um barramento de 40 metros.

O barramento de dados pode ser constituído de três maneiras com um, dois ou quatro fios. As redes com dois e quatro fios, trabalham com os sinais de dado denominados CAN-H (CAN *High*) e CAN-L (CAN *Low*). No caso de barramento com quatro fios, além dos sinais de dados, fazem parte do barramento dois fios da alimentação que é fornecida pela bateria do veículo. Os dados enviados através da rede devem ser interpretados pela análise da diferença de potencial entre os fios CAN-H e CAN-L, utilizando o conceito de par diferencial que ajuda a atenuar os efeitos causados por interferências eletromagnéticas. Conforme a Figura 2, os dados não são representados por *bit 0* e *bit 1*, mas sim por *bits* dominantes e recessivos, seguindo os níveis de tensão estabelecidos para cada um (GUIMARAES, 2007).

Figura 2 – Bits dominantes e recessivos barramento CAN.

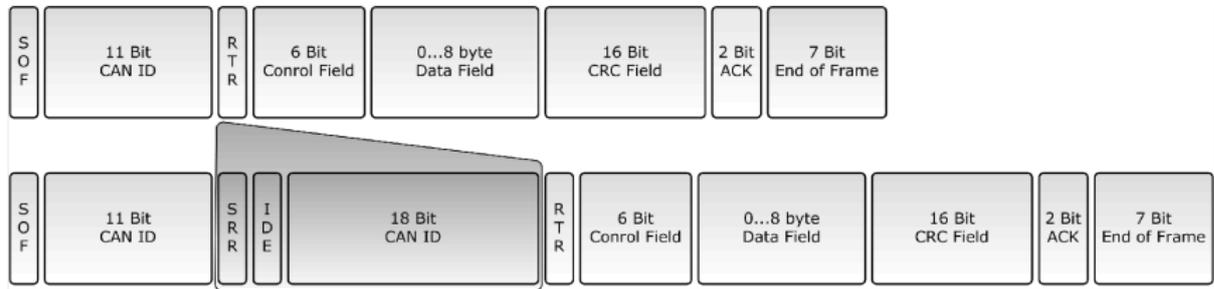


Fonte: Adaptado de (GUIMARAES, 2007).

2.2.2 Formato quadro de mensagens

Existem dois formatos de mensagens no protocolo CAN, formato CAN 2.0A e CAN 2.0B. O formato CAN 2.0A contém um identificador de 11 *bits*, já o formato CAN 2.0B um identificador de 29 *bit* (GUIMARAES, 2007). O número de *bits* do ID (identificação), determina a quantidade de mensagens suportada, onde no padrão de 11 *bit* é igual a 2048 e com 29 *bit* 537 milhões de mensagens na rede (ELETRONICS, 2017). O uso do formato em 11 *bits* pode apresentar um número limitado de mensagens para determinadas aplicações, mas por outro lado o uso do padrão em 29 *bits* aumenta o tempo de transmissão, o que também pode ser crítico dependendo do caso. Na Figura 3, é demonstrada a diferença entre os quadros de mensagens e em seguida, cada campo é descrito.

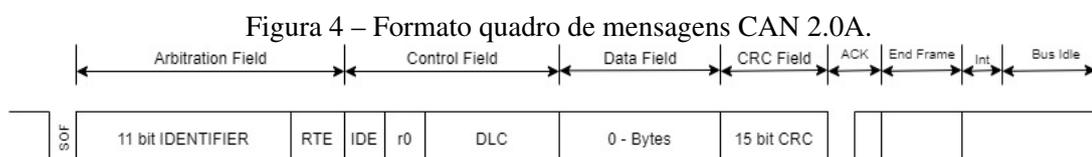
Figura 3 – Quadro CAN 2.0A e 2.0B.



Fonte: Adaptado de (ELETRONICS, 2017).

- SOF (*Start-of-Frame*): é o *bit* que identifica o início de uma mensagem;
- Campo de Arbitragem - (*Arbitration Field*):
 CAN ID: 11/29 *bits* que definem a prioridade da mensagem.
 RTR (*Remote Transmission Request*): Faz a requisição de mensagem remota para outro módulo, distinguindo entre uma requisição de dados ou os dados propriamente ditos.
- Campo de controle - (*Control Field*):
 IDE: Quando for *bit* dominante, configura para o padrão CAN2.0A 11 *bits*. Quando recessivo, adiciona mais 18 *bits* ao quadro, configurando em 29 *bits*.
 DLC: (*Data Length Code*) que são 4 *bits* para indicar o número de *bytes* do campo de dados.
- Campo de Dados - (*Data Field*): até 8 *bytes* que carregam a informação transmitida.
- CRC (*Cycle Redundancy Check*): Campo composto de 16 *bits* dedicados à detecção de erros. São utilizados 15 *bits* para implementação do código de falhas e 1 *bit* recessivo é usado para delimitar o campo.
- ACK (*Acknowledge*): Campo de confirmação consiste em 1 *bit* que indica se a mensagem foi recebida corretamente pelo nó de destino e, 1 *bit* recessivo delimitador.
- EOF (*End Of Frame*): 7 *bits* indicando o fim da mensagem no barramento.

Na Figura 4, é apresentado visualmente o quadro de mensagens utilizando o padrão CAN2.0A de 11 *bits*.



Fonte: Adaptado de (ELETRONICS, 2017).

2.3 DIAGNÓSTICO VEICULAR SOBRE O PROTOCOLO CAN

O automóvel é uma máquina em constante evolução através do tempo. Embora seja visto como um meio de transporte, a forma como é realizado esse transporte tem sido cada vez mais questionada e regulamentada pelos órgãos ambientais. Essas regulamentações em forma de leis no passado, surgiram para iniciar um movimento de controle de emissões de gases. Hoje desempenham um papel muito maior que isso, desenhando o automóvel que teremos no futuro em todos os aspectos, indo além da quantidade de gases que ele poderá emitir.

Conforme Guimarães (2007), o termo diagnóstico veicular representa as funções ou ferramentas que permitem a programação ou verificação do funcionamento de cada módulo eletrônico existente em um veículo. Dentro desse conceito, a indústria automobilística e várias organizações de padronização industriais vêm trabalhando no desenvolvimento de protocolos padrão de diagnóstico veicular. No Brasil, utiliza-se tanto as normas ISO (*International Organization for Standardization*) e SAE (*Society of Automotive Engineers*).

A norma ISO 15031, especifica os aspectos de comunicação via OBDII entre o veículo e o equipamento de diagnóstico externo em termos de interpretação de dados e tempos de resposta. É importante ressaltar que a comunicação não é feita diretamente com os sensores, mas sim com uma ou várias ECU presente no veículo. A comunicação se dá através de uma troca de mensagens no formato pedido e resposta (CUNHA, 2011). A mesma norma define ainda dois parâmetros importantes para serem definidos na mensagem de pedido e resposta:

- **SID - *Service Identification*** - Identificação de serviço solicitado. A identificação de serviço consiste num número hexadecimal entre 01 e 09. Na seção 2.3.1 são descritos cada um dos serviços em detalhes.
- **PID - *Parameter Identification*** - Identificação dos parâmetros que se pode solicitar à ECU. Esses parâmetros são identificados por um número Hexadecimal entre 00 e FF. Exemplo: PID 05 - *Engine Coolant Temperature* (Temperatura do líquido de arrefecimento do motor).

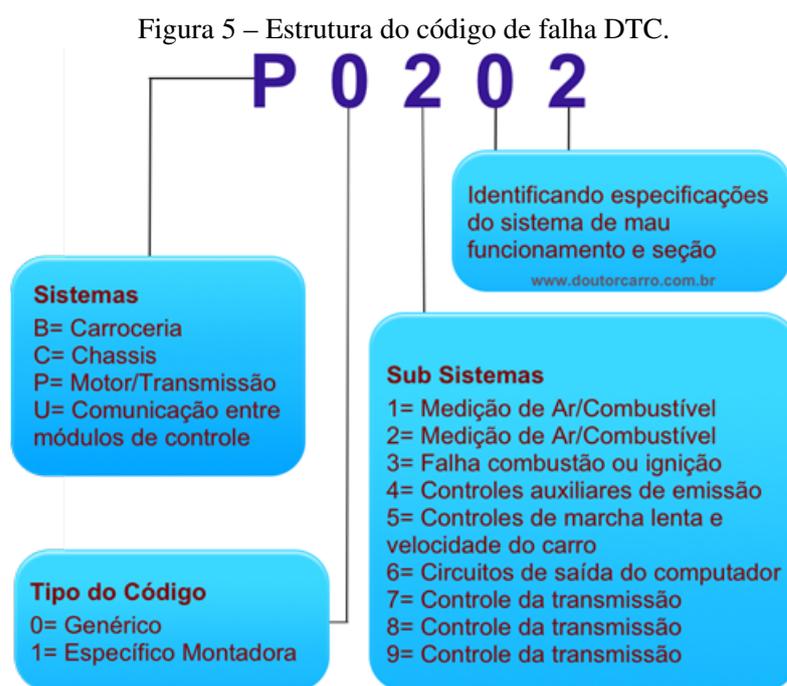
Na Tabela 11 no Anexo A, é apresentada a lista de PIDs disponíveis e suas respectivas descrições.

2.3.1 SID - *Service Identification*

- **SID 01:** O código de serviço 01 solicita os dados em tempo real da ECU conforme o PID solicitado.
- **SID 02:** No SID 02, os dados do sensor requisitado são registrados no momento que a falha acontece.

- SID 03: Realizando uma requisição para o SID 03, são mostrados os códigos DTC (*Diagnostic Trouble Code*). Para representar as falhas registradas nos módulos automotivos, o OBDII utiliza 5 dígitos (COSTA, 2018). O primeiro dígito informa o sistema de origem do código de falha e é representado por uma letra.
 - a) B - *Body System*: Carroceria;
 - b) C - *Chassis*: Chassi;
 - c) P - *Powertrain*: Trem de força (Motor e transmissão);
 - d) U - *Network*: Comunicação entre módulos;

Os dígitos restantes são formados por números. O primeiro identifica o tipo do código (0 para genérico e 1 para específico da montadora). O segundo refere-se ao subsistema onde a falha se encontra e os demais dígitos, descrevem o problema, conforme exemplificado na Figura 5. Os códigos podem ainda ter dígitos no formato hexadecimal na segunda, terceira e quarta posição.



Fonte: Adaptado de (COSTA, 2018).

- SID 04: É utilizado para excluir os códigos de falha armazenados, fazendo com que a luz indicador de falha na ECU presente no painel, se apague.
- SID 05: Através do SID 05 são realizados testes no sensor de oxigênio.
- SID 06: Permite acessar os códigos de diagnóstico de subsistemas que não são monitorados constantemente, como por exemplo o sistema de catalisador.

- SID 07: É utilizado para requisitar DTC que não foram confirmados pela ECU. A forma de uso é similar ao SID 03. Ex.: P000 - Sem códigos pendentes, P0301 - Falha na ignição.
- SID 08:
Através do SID 08 é solicitado o controle dos sistemas de bordo por um equipamento externo.
- SID 09: Informa a ferramenta utilizada as informações de identificação do veículo.

2.3.2 Mensagem de Pedido - *Request Message*

Conforme descrito anteriormente, o protocolo CAN é baseado em pedido e resposta. A Figura 6 demonstra exemplo de quadro de mensagem enviado do equipamento externo para o veículo, solicitando informações, conforme o padrão OBDII.

Figura 6 – Estrutura Mensagem de Pedido.

CAN Data Field							
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Número de bytes adicionais	SID	PID	Bit Stuffing				

Fonte: Adaptado de (CUNHA, 2011).

2.3.3 Mensagem de Resposta - *Response Message*

As mensagens enviadas do veículo para o equipamento externo de diagnóstico, tem a estrutura apresentada na Figura 7.

Figura 7 – Estrutura Mensagem de Resposta.

CAN Data Field							
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Número de bytes adicionais	SID + 0x40	PID	Data A	Data B Opcional	Data C Opcional	Data D Opcional	Bit Stuffing

Fonte: Adaptado de (CUNHA, 2011).

2.3.4 Mensagem de Início de Sessão

A norma ISO 15031 especifica que todas as ECUs devem responder a pelo menos os códigos SID 01 (Dados em tempo real da ECU do trem de força) e o PID 00 (PID suportados [01 - 20]). Esse será o primeiro quadro a ser enviado no início da comunicação.

Conforme (CUNHA, 2011), em um sistema OBDII que utilize o protocolo CAN2.0A 11 *bits*, cada ECU tem um endereço para qual o equipamento externo envia o pedido e, um endereço com o qual responde pedidos. O de resposta é composto pela soma do endereço recebido e o valor oito conforme visto na Tabela 1.

Tabela 1 – Endereços OBDII para CAN2.0A 11 *bits*.

CAN ID (HEX)	Descrição
7DF	Endereço funcional(<i>Broadcast</i>)
7E0	Endereço Físico ECU 1 - Pedido
7E8	Endereço Físico ECU 1 - Resposta
7E1	Endereço Físico ECU 2 - Pedido
7E9	Endereço Físico ECU 2 - Resposta
7E2	Endereço Físico ECU 3 - Pedido
7EA	Endereço Físico ECU 3 - Resposta

Fonte: O Autor.

O tempo de resposta máximo de uma mensagem é definido na norma 15031 no item 5.2.2.6. Define-se no item que o tempo máximo entre o envio de uma mensagem de pedido e a recepção de uma mensagem de resposta, deve ser de 50 ms. Desta forma evita-se a espera por tempo indeterminado pela resposta de um PID não suportado.

2.3.5 Conector padrão OBDII

Com a popularização do gerenciamento eletrônico de motores (injeção eletrônica) e as normas de emissões de gases poluentes mais rígidas, foi criado o padrão OBD para diagnóstico veicular. O padrão inicial determinava que os fabricantes criassem um sistema capaz de diagnosticar e identificar falhas nos componentes do sistema de gerenciamento, que provocassem um aumento nos níveis de emissão de poluentes, com o objetivo de auxiliar no processo de identificação das causas (COSTA, 2018).

A primeira versão desse sistema, conhecido como OBDI, não possuía uniformidade tanto em protocolo quando em conector, onde cada fabricante automotivo desenvolvia seus produtos com seus próprios critérios e normas. Além da falta de uniformidade, o sistema se mostrou ineficiente na identificação do elemento defeituoso. Então em 1988, a SAE propôs a padronização dos sistemas existentes, culminando no atualmente utilizado, padrão OBDII (MACHADO, Antônio, 2008).

De acordo com (SAWANT; MANE, 2018) a interface OBDII contém quatro ou cinco protocolos de sinais, mas a maioria dos fabricantes utiliza apenas um tipo. O conector padrão para o sistema é o conector J1962 que atende aos requisitos da norma ISO 15031-3 (2004). O conector apresenta 16 contatos, entre os contatos, nove tem funções bem definidas e o restante é determinado pelo fabricante do veículo (Lin et al., 2018). Na Tabela 2 são descritos os pinos e posições no conector J1962.

Tabela 2 – Descrição dos pinos conector J1962.

Pino	Descrição	Pino	Descrição
1	Específico fabricante	9	Específico fabricante
2	J1850 Bus +	10	J1850 Bus -
3	Específico fabricante	11	Específico fabricante
4	Chassis GND	12	Específico fabricante
5	GND Sinal	13	Específico fabricante
6	CAN High	14	CAN Low
7	ISO-9141-2 Linha K	15	ISO-9141-2 Linha L
8	Específico fabricante	16	+12V Bateria

Fonte: Adaptado de (SAWANT; MANE, 2018).

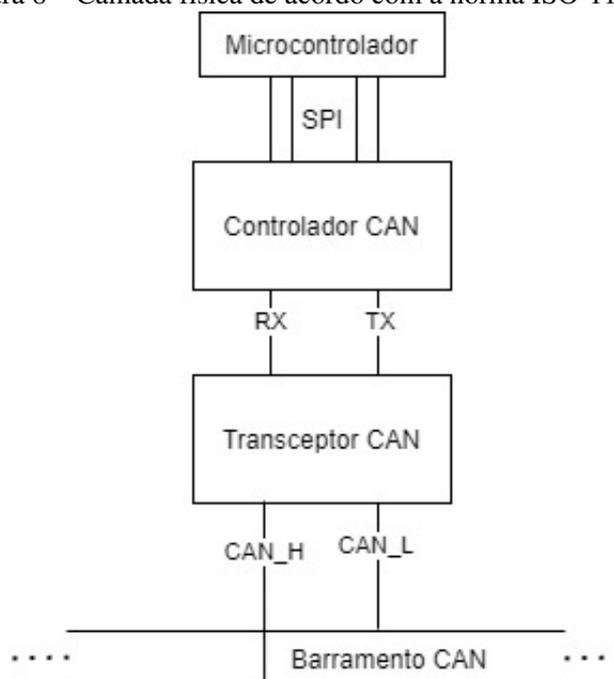
Através do conector, é possível acessar informações dos módulos eletrônicos do veículo conectando um equipamento externo.

2.3.6 Conexão física com Barramento CAN

Para que o sinal vindo do barramento CAN possa ser interpretado como um pacote de dados por um microcontrolador, faz-se necessário implementar a conexão física entre o barramento CAN e o processador prevista na norma ISO-11898-2 (*High-speed medium access unit*).

Conforme visto na Figura 8, a norma especifica a camada física do protocolo, que consiste em uma etapa com interface analógica e digital.

Figura 8 – Camada física de acordo com a norma ISO-11898-2.



Fonte: Adaptado da norma ISO-11898-2.

2.3.6.1 Transceptor CAN

O transceptor CAN é responsável por conectar-se fisicamente ao barramento convertendo os níveis analógicos de tensão para níveis digitais. No processo de transmissão de uma mensagem, o controlador emite dados seriais para a entrada lógica do transceptor. O estado recessivo ou dominante correspondente é emitido nos pinos CAN-H e CAN-L. No recebimento de uma mensagem do barramento, o transceptor recebe estados dominantes ou recessivos nos mesmos pinos CAN-H e CAN-L, esses estados são reproduzidos como níveis lógicos para o controlador para receber quadros (MICROCHIP, 2002).

2.3.6.2 Controlador CAN

O controlador implementa as especificações do protocolo CAN, interpretando os quadros recebidos tanto no formato 11 *bits* ou 29 *bits*, e transmitindo ao microcontrolador através da comunicação SPI (*Serial Peripheral Interface*) (MICROCHIP, 2019).

2.4 SISTEMA DE LOCALIZAÇÃO VIA SATÉLITE

O Sistema de Posicionamento Global, conhecido por GPS ou NAVSTAR-GPS (*Navigation Satellite with Time And Ranging*), é um sistema de rádio navegação desenvolvido pelo Departamento de Defesa dos Estados Unidos da América, visando inicialmente ser o principal sistema de navegação do exército americano (CARVALHO, 2009).

Segundo (FRENZEL, 2013), "o Sistema de Posicionamento Global, também conhecido por Navstar, é um sistema de navegação baseado em satélites que pode ser utilizado por qualquer pessoa com um receptor apropriado para identificar sua posição na terra".

Conforme Carvalho (2009), o modelo adotado pelo GPS fundamenta-se na determinação entre um ponto denominado receptor e os pontos de referência, que são os satélites. Cada satélite emite um sinal codificado que o identifica. O cálculo de cada posição é obtido com a interseção das medidas realizadas pelos satélites. Com a intersecção de apenas três satélites é possível calcular a posição em duas dimensões (latitude e longitude). Para a identificação da posição de pontos, o sistema utiliza-se das coordenadas dos seus satélites.

Carvalho (2009) define que o sistema GPS é dividido em três segmentos funcionais, sendo eles Segmento Espacial, Segmento de Controle e Segmento de Usuário.

O Segmento Espacial é composto por 24 satélites. Os satélites encontram-se em 6 planos de órbitas circulares, com ponto de cruzamento nas longitudes 0, 60, 120, 180, 240 e 300, numa altitude de aproximadamente 20.200 km, em períodos de 12 horas e inclinação de 55°. Esse arranjo garante que se tenha pelo menos quatro satélites visíveis a qualquer hora do dia em qualquer lugar do planeta.

O Segmento de Controle é constituído por estações terrestres que tem o objetivo de monito-

rar, corrigir e garantir o funcionamento do sistema.

O Segmento do Usuário é constituído pelos receptores, e está associado as aplicações do sistema, com diversos tipos de receptores e métodos de posicionamento.

De acordo com Forouzan (2013), O GPS opera com base em um princípio denominado tri-lateração que tem os termos triangulação como sinônimo. O princípio da triangulação, permite determinar uma localização de um ponto na Terra, se distância que se encontra de três satélites e a posição de cada satélite for conhecida, desta forma pode ser calculada por um receptor GPS. O receptor GPS precisa, então, determinar a sua distância com relação a pelo menos três satélites de GPS (centros das esferas). A medição da distância é realizada usando um princípio conhecido como *one-way ranging*, ou medição unidirecional. Cada um dos 24 satélites transmite sincronicamente um sinal complexo, sendo que o sinal de cada satélite apresenta um padrão único. O computador no receptor mede o atraso entre os sinais provenientes dos satélites e sua cópia dos sinais originais para determinar as distâncias até os satélites.

De acordo com o site (GEO, 2012) os dispositivos eletrônicos que utilizam a tecnologia GPS, utilizam o padrão NMEA-0183 (*National Marine Electronics Association*). O padrão define características elétricas do sinal, protocolo de transmissão, sincronismo e formatos específicos de sentenças de transmissão.

2.5 ENVIO DE DADOS VIA INTERNET

2.5.1 GSM

Segundo Oliveira 2004, o GSM consiste em um padrão desenvolvido pela ETSI (*European Telecommunications Standards Institute*) na tentativa de uniformizar os sistemas celulares europeus existentes. Considerando uma tecnologia 2G, o GSM provê a transferência de voz e dados através de comutação por circuito. De acordo com (DI LUCIA SANTOS, 2008), A rede é formada por interfaces abertas e padronizadas, seguindo sua principal intenção, montar uma arquitetura mais abrangente possível. A rede é estruturada para que seja possível a integração entre componentes de diferentes fabricantes. Os componentes da arquitetura são divididos em quatro grupos. O conjunto desses grupos é chamado de PLMN (*Public Land Mobile Network*) rede móvel pública terrestre, e é implantado por uma operadora. Estes grupos são:

- MS - *Mobile Station* - Estação Móvel: Formado pelo próprio aparelho celular, computador ou qualquer outro sistema de comunicação de voz ou dados. Necessita de um cartão SIM, que guarda seu registro de rede.
- BSS - *Base Transceiver System* - Sistema de estação base: Comunica com as estações móveis e envia informações para os sistemas de comutação de rede, NSS.
- NSS - *Network Switching System* - Sistema de comutação de rede: Processa informações através de interfaces e protocolos e gerencia o banco de dados.

- OMS - *Operations and Maintenance System* - Sistema de operação e manutenção: Responsável por comandar os grupos de componentes.

2.5.2 GPRS

Conforme (SVERZUT, 2005), nas últimas décadas a evolução tecnológica tem provocado profundas transformações na sociedade moderna, influenciando drasticamente os hábitos e costumes das pessoas. Analisando essa evolução, pode-se concluir que um fator determinante para essa transformação é a crescente demanda por novos serviços e aplicações. As redes celulares 1G e 2G, foram desenvolvidas para transportar somente informação de voz, ou seja, eram usados apenas para prover o serviço de voz entre dois usuários.

Ainda conforme (SVERZUT, 2005), visando acompanhar as evoluções tecnológicas provocadas por novas demandas dos usuários, foram implementados novos serviços e aplicações nas redes GSM. Segundo (DI LUCIA SANTOS, 2008), a rede de segunda geração 2G preparou-se para oferecer telefonia digital, mas não para acessar a internet. A internet transporta dados por pacotes, através do protocolo IP (*Internet Protocol*) e, para que a rede móvel seja adaptada à internet, é preciso que os dados sejam organizados em pacotes. Foi criado então a tecnologia GPRS (*General Packet Radio Services*) cujo o objetivo é tráfego de dados por pacotes para que a telefonia celular seja integrada à internet.

2.5.2.1 Tarifação serviço GPRS

De acordo com Dillenburg (2004), nesse sistema de comutação de pacotes, um canal de rádio só é utilizado quando o usuário está efetivamente enviando ou recebendo dados. A tarifação do serviço é feita pelo volume de dados transportados. Aplicações de telemetria tipicamente são de baixo tráfego, resultando em uma tarifa mensal bastante reduzida. Marcos (2004), destaca que as operadoras de GSM no Brasil oferecem o serviço em todas as áreas em que sistema de voz está em operação. Outro ponto importante para aplicações em telemetria, é que o custo de um modem GPRS é elevado e pode ser limitante para aplicações de baixo valor.

2.5.3 Redes Wireless Wi-Fi

Redes *wireless* ou redes sem fio são um sistema de comunicação de dados extremamente flexível, que pode ser usado como uma extensão, ou uma alternativa a redes locais (LAN cabeadas). É uma tecnologia que combina conectividade de dados com mobilidade através de tecnologia de radiofrequência (RF) (MORAES, 2014).

Ainda conforme Moraes (2014), sistemas baseados em radiofrequência utilizam micro-ondas para transmitir o sinal através do espaço que cerca e, geralmente utilizam faixas de frequência ISM (*Industrial Scientific Medical*) que contempla as faixas de 900 MHz, 2,4 GHz

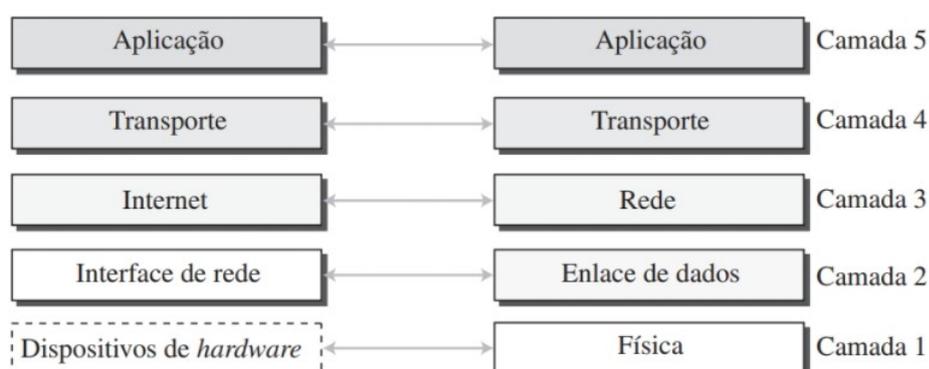
e 5 GHz. No Brasil, para o sistema *wireless* LAN na frequência de 2,4 GHz é permitido o uso de 11 canais de transmissão. Quanto ao alcance da transmissão, Moraes(2014) comenta que a distância com que as ondas RF podem se comunicar está relacionada basicamente com a potência de transmissão, sensibilidade do receptor e o caminho por onde a onda se propaga especialmente em ambientes *indoor*. O alcance em raio de uma rede *wireless* LAN em ambiente *indoor* varia de 35 a 100 metros.

O IEEE definiu as especificações para uma LAN sem fio, o padrão denominado IEEE 802.11, que abrange as camadas física e de enlace de dados. Em alguns países, incluindo os Estados Unidos e o Brasil, o público usa o termo *Wi-Fi* (abreviatura de *Wireless Fidelity*, ou Fidelidade Sem Fio) como sinônimo de LAN sem fio. O termo *Wi-Fi*, no entanto, refere-se a uma LAN sem fio que é certificada pela *Wi-Fi Alliance*, uma associação industrial global e sem fins lucrativos constituída por mais de 300 empresas dedicadas a promover o crescimento das LANs sem fio (FOROUZAN BEHROUZ, 2013).

2.5.3.1 TCP/IP

Conforme Forouzan (2013), TCP/IP (*Transmission Control Protocol/Internet Protocol*) ou Protocolo de Controle de Transmissão/Protocolo de Internet, consiste em uma pilha de protocolos usados na internet atual. É um protocolo hierárquico composto de módulos interativos, cada um dos quais provendo uma funcionalidade específica. A pilha de protocolos TCP/IP original foi definida como quatro camadas de *software* construídas sobre o *hardware*. Hoje, no entanto, o TCP/IP é visto como um modelo de cinco camadas. A Figura 9, demonstra ambas as configurações.

Figura 9 – Camadas TCP/IP.



Fonte: Adaptado de (FOROUZAN BEHROUZ, 2013).

- Camada de Aplicação: A comunicação na camada de aplicação se dá entre dois processos (dois programas em execução nessa camada). Para se comunicar, um processo envia um pedido para o outro processo e recebe uma resposta. A comunicação processo

a processo é a função da camada de aplicação. A camada de aplicação na Internet inclui muitos protocolos predefinidos, mas o usuário também pode criar um par de processos para serem executados nos dois *host*. O Protocolo de Transferência de Hipertexto (HTTP – *HyperText Transfer Protocol*) é um meio de acesso à *World Wide Web* (WWW) (FOROUZAN BEHROUZ, 2013).

- Camada de Transporte: A camada de transporte no *host* de origem recebe a mensagem da camada de aplicação, a encapsula em um pacote da camada de transporte (denominado um segmento ou um datagrama de usuário, dependendo do protocolo) e a envia através da conexão lógica (imaginária) para a camada de transporte no *host* de destino (FOROUZAN BEHROUZ, 2013).
- Camada de Rede: A comunicação na camada de rede é *host a host*. No entanto, uma vez que pode haver vários roteadores da origem ao destino, os roteadores no caminho são responsáveis por escolher a melhor rota para cada pacote. A camada de rede é responsável pela comunicação *host a host* e pelo roteamento de pacotes através de possíveis rotas (FOROUZAN BEHROUZ, 2013).
- Camada de Enlace de Dados: Os roteadores são responsáveis por escolher os melhores enlaces, no entanto, quando o próximo enlace a ser utilizado é determinado pelo roteador, a camada de enlace de dados é responsável por pegar o datagrama e movê-lo através do enlace. Este pode ser uma LAN cabeada usando um *tswitch* de camada de enlace, uma LAN sem fio, uma WAN cabeada ou uma WAN sem fio. Pode também haver diferentes protocolos usados com qualquer tipo de enlace. Em cada caso, a camada de enlace de dados é responsável por mover o pacote através do enlace (FOROUZAN BEHROUZ, 2013).
- Camada Física: A camada física é responsável por transportar os *bits* individuais de um quadro através do enlace. Embora a camada física seja o nível mais baixo na pilha de protocolos TCP/IP, a comunicação entre dois dispositivos na camada física é ainda uma comunicação lógica, pois há outra camada escondida sob a camada física, o meio de transmissão (FOROUZAN BEHROUZ, 2013).

2.5.3.2 Cliente-Servidor

Ainda de acordo com Forouzan (2013), o WWW atual é um serviço cliente-servidor distribuído, em que um cliente usando um navegador (*browser*) pode acessar um serviço por meio de um servidor. No entanto, o serviço fornecido é distribuído entre muitos locais chamados sites. Cada site possui um ou mais documentos, chamados de páginas *Web*. Cada página *Web*, no entanto, pode conter algumas ligações ou *links* para outras páginas do mesmo ou de outros sites. Cada uma delas é um arquivo com um nome e um endereço. A página *Web* é armazenada

no servidor. Cada vez que um pedido chega, o documento correspondente é enviado ao cliente.

Conforme Forouzan (2013), uma página *Web* precisa ter um identificador único para distingui-la das outras. Para definir, são necessários quatro identificadores.

- **Protocolo:** É a abreviação para o programa cliente-servidor que é utilizada para acessar a página *Web*. Embora na maior parte do tempo o protocolo seja o Protocolo de Transferência de Hipertexto (HTTP – *HyperText Transfer Protocol*), também pode-se usar outros protocolos, como o Protocolo de Transferência de Arquivos (FTP – *File Transfer Protocol*).
- **Host:** O identificador do *host* pode ser o endereço IP do servidor ou o nome único dado ao servidor. Os endereços IP podem ser definidos usando notações decimais pontuadas, por exemplo, 64.23.56.17. O nome costuma ser o de domínio que define univocamente o *host*.
- **Porta:** A porta, um número inteiro de 16 *bits*, normalmente é predefinida para a aplicação cliente-servidor. Se o protocolo HTTP é usado para acessar a página *Web*, o número de porta conhecido é 80. No entanto, se uma porta diferente for usada, o número pode ser fornecido explicitamente.
- **Caminho:** O caminho identifica o local e o nome do arquivo no sistema operacional subjacente.

O protocolo HTTP define o formato das mensagens de pedido e resposta. A primeira linha de uma mensagem de pedido é chamada de linha de solicitação. Existem três campos nessa linha separados por um espaço e terminados por dois caracteres - caractere de controle, que move o cursor para nova linha e nova linha. O campo método define os tipos de solicitação. Conforme a Tabela 3, são demonstrados os diversos tipos de métodos disponíveis.

Tabela 3 – Métodos HTTP.

Método	Ação
GET	Solicita um documento ao servidor.
HEAD	Solicita informações sobre um documento, mas não o documento em sí.
PUT	Envia um documento do cliente para o servidor.
POST	Envia alguma informação do cliente para o servidor.
TRACE	Ecoa a solicitação recebida
DELETE	Remove página <i>Web</i> .
CONNECT	Reservado
OPTIONS	Consulta opções disponíveis

Fonte: Adaptado de (FOROUZAN BEHROUZ, 2013).

2.6 TRABALHOS RECENTES NA ÁREA

Para uma compreensão do contexto ao qual o projeto está inserido e, realizar uma pesquisa sobre as diversas áreas as quais envolvem a coleta e análise de informações utilizando a rede CAN, o conector OBDII, GPS e acelerômetros, dividiu-se o tópico em três seções. A primeira seção está relacionada com alguns trabalhos focados em utilizar um método para a leitura dos dados disponíveis no barramento CAN e visualização dos dados, a segunda demonstra a simulação da rede CAN para possibilitar o desenvolvimento de aplicações na área e a terceira, aplicações que utilizam do mesmo método de leitura, porém, focado na aquisição para gerenciamento do motorista e frota.

2.6.1 Aquisição de dados e visualização em tempo real

Como ferramenta de coleta de informações e visualização em tempo real dos dados, a plataforma de aquisição de dados proposta por Andria (2015) consiste em um sistema de coleta de dados que pode ser adaptada e implementada em todos os veículos que seguem o padrão OBDII de comunicação. O sistema proposto utiliza uma placa de desenvolvimento Raspberry Pi, rodando um código em *Python* para a leitura dos sensores como acelerômetro MPU6050 e um receptor GPS. Os dados do acelerômetro, GPS e dados da CAN são salvos em um banco de dados MySQL e enviados via *Wi-Fi* para um supervisor.

Ainda conforme Andria (2015), durante os testes experimentais, verificou que a taxa de amostragem máxima dos dados coletados na porta OBD variam conforme a ECU do veículo e o *scanner* utilizado. A utilização de um *scanner* OBD comercial que conecta com a Raspberry Pi via *Bluetooth*, pode se tornar um gargalo no sistema devido a taxa de transmissão, podendo afetar a leitura correta de alguns parâmetros que requerem maior taxa de amostragem, como o RPM do motor.

Outro trabalho focado na análise de dados em tempo real é de Sawant (2018), onde o autor utiliza uma plataforma microcontrolada para conectar-se a porta OBDII do veículo, e utilizando os padrões de comunicação estabelecidos nas normas relacionadas, realiza a coleta de informações da ECU e de subsistemas. Para a visualização dos dados em tempo real, Sawant (2018) propõe conectar o protótipo de coleta ao *desktop* via interface *bluetooth*. A análise dos dados é realizada em uma interface gráfica programada utilizando linguagem C em ambiente *Windows*. Sawant (2018) cita que o sistema desenvolvido tem foco em baixo custo de implementação comparado com outros sistemas de aquisição de dados convencionais.

2.6.2 Simulação de Rede CAN via *software*

Atualmente, o desenvolvimento de aplicações envolvendo a rede CAN automotiva requerem ter acesso a um veículo real e estar conectado via porta OBDII durante todo o processo, o que

na maioria das vezes não é prático ou até mesmo viável por não ser um ambiente ideal para esse tipo de trabalho.

Pimple (2018), desenvolve seu sistema de telemetria utilizando ferramentas disponíveis no mercado para a simulação e coleta dos dados da rede CAN automotiva, utilizando o *software* Vector CANoe 9.0 que simula o barramento CAN de um veículo. Para realizar a coleta de dados, é utilizado o dispositivo de *hardware* Vector CANcaseXL. Para criar um modelo de um veículo real dentro do CANoe, o autor utilizou um arquivo de dados fornecido pela Motec USA, contendo dados da rede CAN de um veículo interagindo com o CANcaseXL. Para comunicar com o Arduíno Uno R3, utiliza uma placa com o circuito integrado MCP2515, e para visualizar os dados recebidos, usa uma interface desenvolvida no MATLAB exibindo os dados em tempo real.

2.6.3 Aplicação focada no gerenciamento da frota

No artigo de Naik (2017), o autor demonstra um sistema com foco em educar o motorista, coletando informações de comportamento na direção através de dados de consumo de combustível, nível de emissões de gases e detecção de motorista alcoolizado. Para servir de suporte ao projeto, o autor desenvolveu um dispositivo de *hardware* que simula um veículo, gerando os dados de velocidade, RPM, temperatura de motor, códigos de falha, sensor de combustível e sensor de gases no escapamento. Conectado a esse dispositivo de *hardware* via barramento CAN, está um microcontrolador Atmega328. Conectado ao Arduíno estão o módulo GPS e o sensor de nível de álcool.

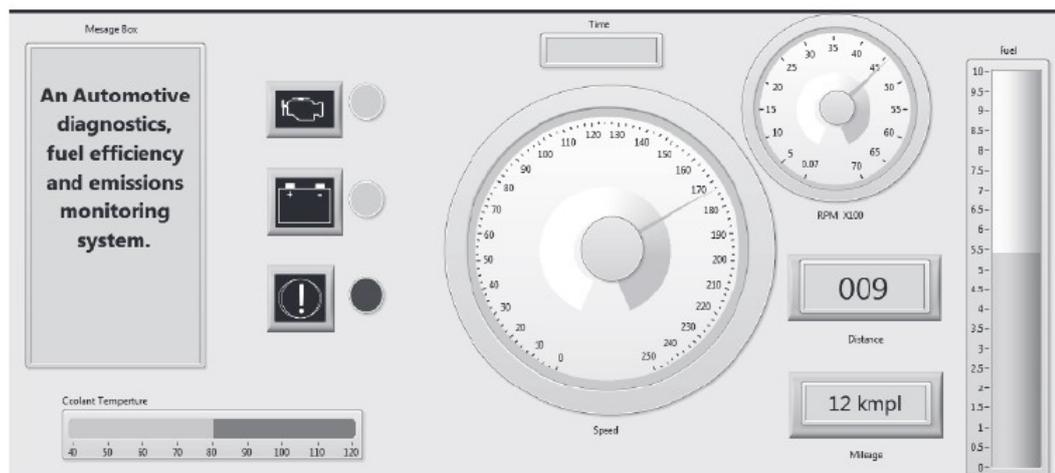
Conforme a Figura 10, para monitorar os dados, o autor desenvolveu um painel de instrumentos utilizando a plataforma *Labview* onde são exibidos os dados de velocidade, distância percorrida, consumo em quilômetros por litro, nível de combustível no tanque e temperatura de motor.

O ponto principal do trabalho de Naik (2017) é, através dos dados coletados, detectar certo tipo de comportamento ao volante e, caso ultrapasse limites preestabelecidos de alguns parâmetros, orientar o motorista ou emitir relatórios à empresas ou organizações governamentais de controle ambiental.

Ainda Naik (2017) utiliza o parâmetro de consumo de combustível de forma que, conforme um valor esperado para determinado veículo, valores que ultrapassem esse *target* podem indicar alguns pontos de problema no veículo como os citados abaixo, sendo os dois primeiros inerentes ao projeto do veículo e os dois últimos, ao comportamento do condutor.

- Uso de lubrificantes eficientes
- Pneus com baixa resistência de rodagem;
- Manutenções sempre em dia;

Figura 10 – Supervisório em *Labview* para visualização de parâmetros.



Fonte: Adaptado de Naik (2017).

- Modificar os hábitos de direção;

O sistema proposto registra alertas nas seguintes situações:

1) Se um veículo apresentar os níveis de emissão de gases acima do limite predefinido, um alerta ao usuário é emitido indicando a necessidade de manutenção no veículo.

2) Como cada veículo apresenta um modo de condução ideal para atingir o melhor consumo de combustível, identificando comportamentos de direção agressiva como arrancadas bruscas, freadas bruscas e alta velocidade e cruzando esses dados com o cálculo do consumo de combustível, é possível indicar ao motorista que corrija seu modo de condução para atingir um melhor consumo de combustível.

Ainda dentro da área de aplicação de gerenciamento de motorista, Andria (2017) contribui com um trabalho utilizando o sistema de telemetria baseado em *On Board Diagnostic*, citado no item 2.7.1, voltado para a área de comportamento de condução. O objetivo do trabalho proposto, é avaliar os diferentes comportamentos de motoristas com base em mudanças de marcha e aceleração longitudinal, combinando dados coletados na porta OBD e dados de medições inerciais. O autor propõe uma análise focada no estilo de direção, diferentemente dos demais estudos utilizando a análise da troca de marcha que propõe o momento certo de troca.

Para o desenvolvimento do trabalho, Andria(2017) analisa as variações da velocidade, aceleração e rotação do motor, que acontecem no curto intervalo de tempo entre a seleção de duas engrenagens do câmbio. Para isso, um algoritmo de segmentação de marcha, definição de recursos e classificação de mudança de marcha é proposto. Inicialmente é analisada a relação entre a velocidade do veículo e a rotação do motor.

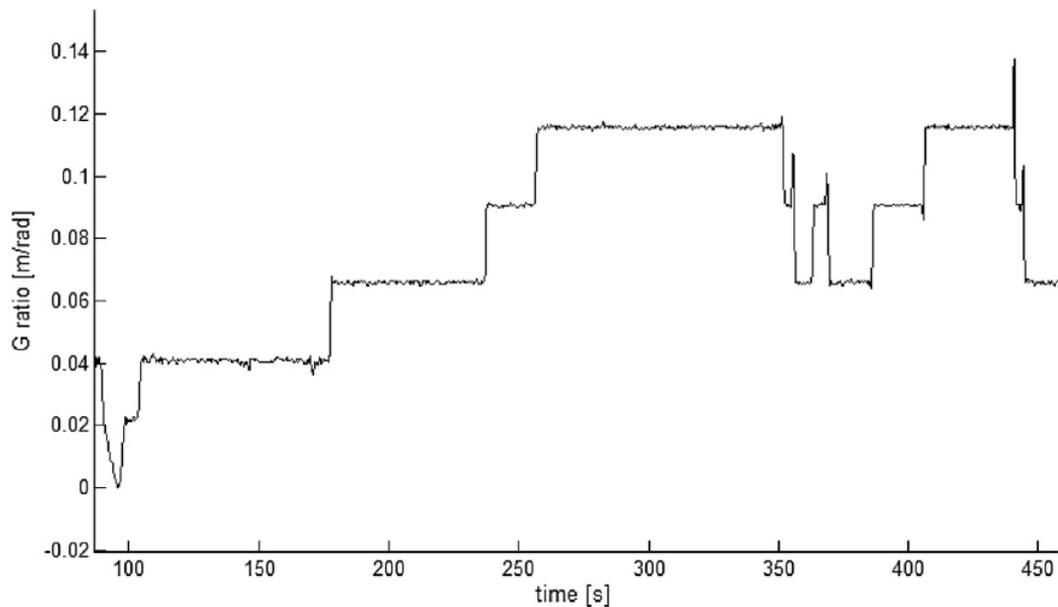
Ainda conforme Andria (2017), na equação (2.1) é possível estabelecer uma relação entre a velocidade do veículo $v(t)$ e a rotação do motor $r(t)$, sendo esse um parâmetro importante para

classificar o estilo de direção.

$$G(t) = \frac{v(t)}{r(t)} \quad (2.1)$$

Na Figura 11 as linhas horizontais formadas no gráfico correspondem ao carro em ordem de marcha com determinada marcha engatada. Os pontos que se encontram fora da linha horizontal, são os eventos de troca de marcha, onde o RPM desce, mas a velocidade continua a mesma.

Figura 11 – Gráfico de relação entre Velocidade e Rotação do motor.



Fonte: Adaptado de Andria (2017).

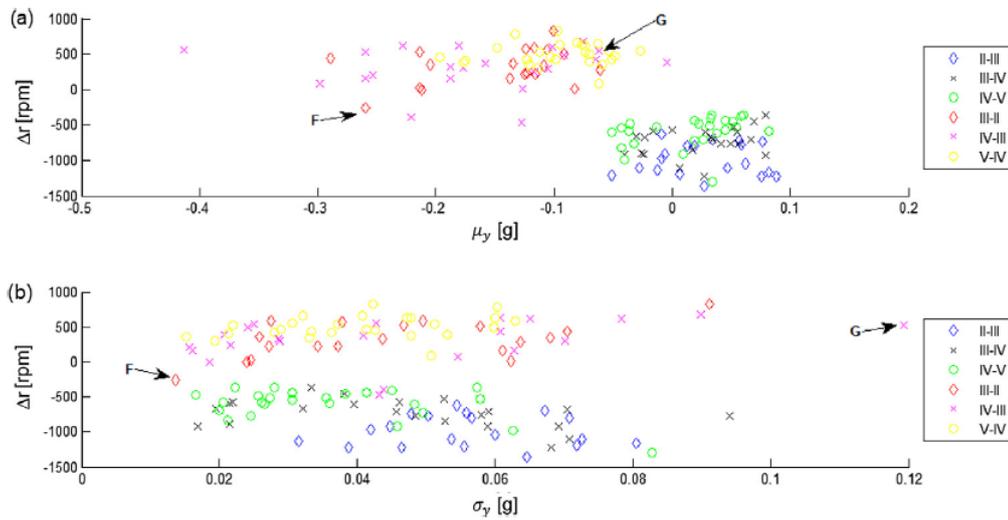
No trabalho Andria (2017) ressalta que deve-se tomar cuidado na análise de transição de fase do gráfico, para isso são aplicados filtros para suavizar o *overshooting* na curva $G(t)$ principalmente nas reduções de marcha.

Analisando a variação da rotação do motor versus a aceleração longitudinal na figura 12, é possível estimar o comportamento do condutor no momento do aumento de marcha. No gráfico (a) da Figura 12, tem-se a variação do RPM do motor versus o desvio médio da aceleração, e em (b) o RPM versus o desvio padrão da aceleração medida. Na legenda são indicadas as situações de troca de marcha que foram coletados os dados de aceleração.

Conforme a Tabela 4, Andria (2017) classifica dois comportamentos durante esse evento, classificados em *Soft/cruising* (SC) ou *Hard/accelerating* (HA).

Conforme análise feita por Andria (2017), *Soft/Cruising* (SC) no aumento de marcha é a situação em que o motorista deseja manter a rotação do motor baixa uma velocidade constante, para obter uma economia de combustível. *Hard/accelerating* (HA) é notada na fase de aceleração do veículo, nesse comportamento nota-se que a velocidade aumenta e há trocas de marcha enquanto o pedal do acelerador está acionado. Para analisar o comportamento durante as frenagens, utiliza os dados de variação da aceleração e da rotação do motor, estimando assim se foi

Figura 12 – Gráfico da relação entre a variação do RPM e desvio da aceleração longitudinal.



Fonte: Adaptado de Andria (2017).

Tabela 4 – Tabela de classificação de motorista conforme troca de marcha.

Classificação	Delta V	Desvio médio aceleração	Desvio padrão aceleração
<i>SC</i>	menor ou igual a zero	menor que 0,05g	menor ou igual a 0,05g
<i>HA</i>	maior que zero	maior que 0,05g	maior que 0,05g

Fonte: Adaptado de (Andria et al., 2017).

utilizado a própria rotação do motor ou somente o freio hidráulico para reduzir a velocidade.

Por fim, Andria (2017) realiza o cruzamento dos dados entre aumento de marcha e de reduções para classificar o motorista entre moderado e agressivo. O classificado como moderado, tem aumento de marcha classificado como (SC) e faz uso do freio hidráulico do veículo e, do contrário, o agressivo, faz aumento de marcha como (HA) e reduções utilizando mais o freio a motor.

A Tabela 5, apresenta um resumo de trabalhos com diferentes ferramentas de análise e visualização dos dados, os quais, cada um buscou analisar dados da ECU automotiva para detecção de falhas ou estimar comportamento ao volante.

Tabela 5 – Comparativo entre trabalhos na área.

Autor	Proposta	Ferramenta	Parâmetro e variáveis monitorados
Naik, Pa- vankumar (2017)	Utilizar a rede CAN para aquisição de dados ECU propondo uma nova abordagem de controle de emissões de gases, e emitir alertas em caso de problemas ou violação de limites pré selecionados para as variáveis da ECU.	Módulo GPS, Microcontrolador Atmega328 conectado ao veículo pela interface CAN OBDII. <i>Labview</i> para visualização dos dados.	Velocidade, RPM, Temperatura do Motor, Nível de combustível, Sensor de gases, Códigos DTC.
Pimple, Prajakta (2018)	<i>Data logger</i> em tempo real para visualização e diagnóstico da rede CAN automotiva.	Ferramenta de simulação de rede CAN. Microcontrolador da plataforma Arduíno Uno R3 conectado pela interface CAN OBDII. Interface MATLAB para visualização dos dados.	Velocidade, RPM, Temperatura do Motor, Nível de combustível, Códigos DTC.
Andria, Gregorio (2015)	Plataforma de aquisição de dados para telemetria automotiva focada em estilo de direção, gerenciamento de frotas e detecção de falhas.	Computador embarcado <i>Raspberry Pi</i> conectado via <i>bluetooth</i> a um <i>scanner</i> ligado ao conector OBDII do veículo. Para visualização dos dados interface baseada em <i>Labview</i> e um <i>dashboard</i> via internet.	Velocidade, RPM, Temperatura do Motor, Nível de combustível, aceleração inercial.
Sawant (2018)	<i>Data logger</i> em tempo real para visualização de e diagnóstico via rede CAN.	Microcontrolador com interface CAN conectado ao conector OBDII do veículo. Transferência de dados entre protótipo e PC usando <i>Bluetooth</i> . Para visualização interface para PC programada utilizando o QT <i>Creator</i> .	Velocidade, RPM, Temperatura do Motor, Nível de combustível, Sensor de gases, Códigos DTC.

Fonte: O Autor.

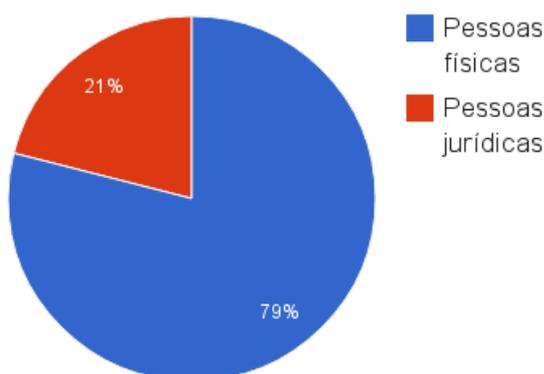
2.7 SOLUÇÕES COMERCIAIS RELACIONADAS COM O TEMA

De acordo com Mourad (2018) em artigo, a economia de recursos, manutenções e combustível estão entre os objetivos dos gestores de frotas. Por essa razão, o mercado de monitoramento e o gerenciamento de frotas, devem focar em tornar as empresas prestadoras de serviços de transportes, mais produtivas e reduzir o risco de operação.

Nesse cenário, conforme relatório da Associação Brasileira das Empresas de Gerenciamento de Riscos e de Tecnologia de Rastreamento, Monitoramento e Telemetria (GRISTEC), em 2016, aproximadamente 300 empresas atuavam no Brasil nos segmentos de tecnologia de informação veicular e gerenciamento de riscos.

Ainda conforme relatório setorial da GRISTEC (2016), as empresas do setor atendiam na época cerca de 740 mil clientes em todo o Brasil. Na Figura 13 é possível observar a distribuição dos clientes desse tipo de serviço.

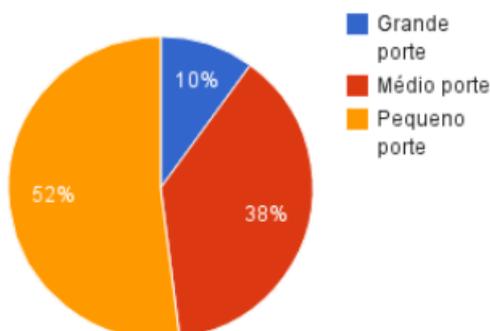
Figura 13 – Distribuição de clientes de serviços de gerenciamento de frotas.



Fonte: Adaptado de (GRISTEC, 2016).

Entre as pessoas jurídicas, conforme visto na Figura 14, as empresas de pequeno porte formam a maior parte do total.

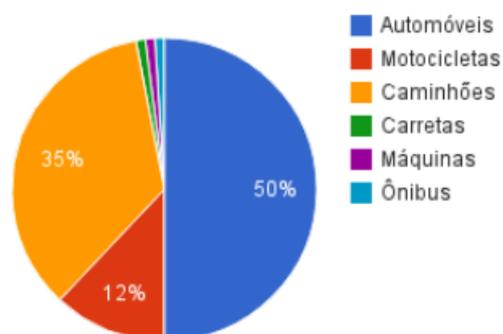
Figura 14 – Tamanho dos clientes.



Fonte: Adaptado de (GRISTEC, 2016).

Entre os milhões de veículos circulantes no território nacional, apenas 2,3 milhões possuem sistemas de monitoramento. Conforme visto na Figura 15, os veículos rastreados se dividem da seguinte maneira

Figura 15 – Tipo de veículos com serviços de monitoramento.



Fonte: Adaptado de (GRISTEC, 2016).

Dentro desse contexto, a Tabela 6 compara alguns dos serviços prestados por empresas de destaque no setor de monitoramento veicular. Dentre os principais serviços em comum que cada empresa disponibiliza ao cliente, os serviços de telemetria veicular e rastreamento em tempo real são os mais presentes no portfólio dos prestadores.

Tabela 6 – Comparativo dos serviços prestados por empresas nacionais do setor.

Empresa	Serviços	Método de leitura dos dados	Método de exibição	Custo
Carrorama®	Relatórios e <i>DashBoard</i> , alertas de problemas mecânicos, localização em tempo real, Sensores de velocidade, relatório de consumo.	<i>Scanner</i> conectado à porta OBDII CAN, receptor GPS, GPRS	<i>dashboard online</i> , aplicativo para dispositivo móvel	Plano mensal de acesso ao serviço, conforme número de veículos
Creare Sistemas®	Tempo e distância percorrida, arrancadas bruscas, registro de eventos de rotação ou velocidades excessivas, localização em tempo real, alertas de problemas mecânicos.	<i>Scanner</i> conectado à porta OBDII CAN, receptor GPS, GPRS	<i>dashboard online</i> , Aplicativo para dispositivo móvel	Plano mensal de acesso ao serviço, conforme número de veículos
Infleet®	Registro de eventos de rotação ou velocidades excessivas, Quilometragem por veículo, Identificação de frenagens, curvas e acelerações, Cultura de condução econômica, Tempo médio de conserto e falhas.	<i>Scanner</i> conectado à porta OBDII CAN, receptor GPS, GPRS.	<i>dashboard online</i> , Aplicativo para dispositivo móvel	Plano mensal de acesso ao serviço, conforme número de veículos.

Fonte: O Autor.

3 METODOLOGIA APLICADA

Este capítulo aborda a estratégia aplicada no desenvolvimento de um protótipo de coleta de informações da rede CAN automotiva através da porta OBDII, para monitorar eventos relacionados ao veículo e disponibilizar ao usuário os dados para consulta. Sendo assim, será descrita a metodologia de desenvolvimento do projeto do *hardware* e *firmware* onde esta será separado em etapas. O detalhamento do desenvolvimento das etapas aqui descritas será descrito no capítulo 4.

Anteriormente foram citados e comparados trabalhos que descreviam soluções comerciais para a aquisição de dados veiculares citados na seção 2.6. Nos trabalhos de Naik (2017), Pimple (2018) e Sawant (2018), se utiliza a ideia de coleta e exibição dos dados em tempo real com visualização em uma interface gráfica, mas a forma de implementação e uso da solução, se distancia das aplicações comerciais por serem pouco práticas de aplicar em larga escala. Nesses trabalhos, a conexão entre o protótipo de coleta e um computador com o uso de cabos, ou como no trabalho de Sawant (2018), via interface *bluetooth*. Por outro lado, as soluções comerciais apresentadas na Tabela 6, utilizam de ferramentas de comunicação e rastreamento em tempo real como a tecnologia GPRS, que tem a necessidade de uso da rede celular para enviar os dados para um servidor e, visualizar em uma interface no computador. Por se tratarem de aplicações em grandes frotas de veículos, esses serviços são pagos através de mensalidades pelo usuário.

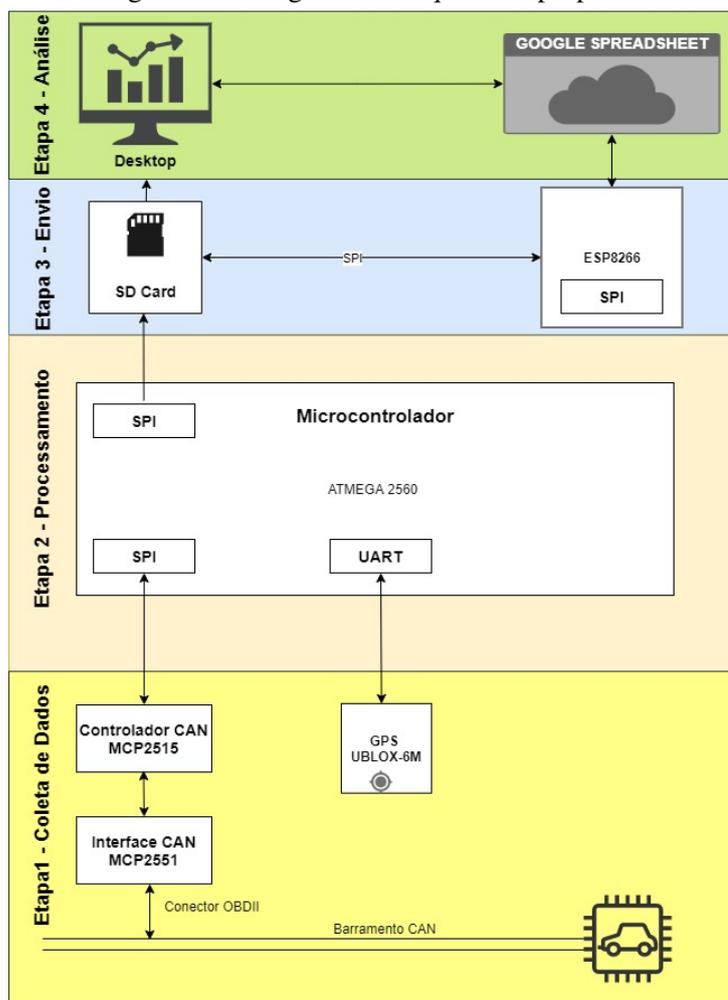
Para este trabalho em específico, como o foco é atender a pequenos gestores de frota, mantendo a solução simples e barata, será utilizado parte das duas abordagens mencionadas no parágrafo anterior. Com isso, será realizado a coleta dos dados tendo como premissa que não há assinatura de um plano de dados da rede celular no veículo, optando pelo armazenamento primário das informações no *SD Card* e, é proposto também, o envio de dados via internet para consulta através da exibição em um *dashboard online*.

3.1 METODOLOGIA DE PROJETO

Na Figura 16 é apresentada uma diagramação da implementação do protótipo, segmentando em etapas de desenvolvimento e as respectivas interfaces entre elas. A Figura 16 mostra uma visão geral do desenvolvimento, iniciando pela coleta de dados através do barramento do veículo, processamento por um microcontrolador, o armazenamento e envio dos dados, e por fim, a etapa de análise dos dados coletados. Dentro de cada etapa descrita, são demonstrados os respectivos testes funcionais. Em seguida, são demonstrados a metodologia dos testes de integração e testes de validação final do protótipo.

A seguir são descritas as etapas de desenvolvimento seguindo o diagrama da Figura 16.

Figura 16 – Diagrama da arquitetura proposta.



Fonte: O Autor.

3.1.1 Etapa 1 - Coleta de dados

A Etapa 1 consiste na coleta de dados. Conforme mostrado na Figura 16, em específico no que se refere à coleta de dados, trata-se da entrada de dados do protótipo, tendo como origem o barramento CAN do veículo e receptor GPS. Para implementar a coleta de dados, foi necessário desenvolver a comunicação com cada um dos sensores, utilizando como base as informações contidas no referencial teórico e trabalhos com tema semelhante à presente proposta.

Conforme as aplicações citadas anteriormente nesta monografia, o ponto comum entre os trabalhos de Andria (2015), Pimple (2018) e Naik (2017) descritos na seção 2.6, é o uso da porta OBDII para a conexão entre o protótipo e o veículo. Portanto neste trabalho, foi feito o uso da porta OBDII para a conexão entre o protótipo e o veículo. De acordo com o referencial teórico, para obter informações da rede CAN, deve-se enviar padrões de mensagens de requisição para o barramento a qual se está conectado.

Ainda na Etapa 1, foi implementada a leitura dos dados recebidos via GPS. Os dados recebidos via GPS, tem a finalidade de fornecer as coordenadas geográficas de cada evento registrado de forma semelhante ao trabalho de Naik (2017) e, com as soluções comerciais apresentadas na Tabela 6. O receptor GPS utilizado foi o módulo NEO-6M-0-001 do fabricante U-Blox. Conforme o referencial teórico, a aquisição das coordenadas geográficas foi implementada realizando a interpretação dos dados fornecidos pelo receptor de sinal GPS, no formato NMEA-0183. Dos dados recebidos do módulo GPS, são armazenadas as coordenadas de latitude, longitude, data e hora. Ao receber os dados de data e hora, uma correção ao fuso horário local faz-se necessária utilizando as bibliotecas disponíveis.

3.1.2 Etapa 2 - Processamento

Na Etapa 2, em específico no que se refere ao processamento, foi realizado o tratamento dos dados coletados através dos passos descritos na Etapa 1 e, realizado o processamento utilizando o microcontrolador. O microcontrolador utilizado foi o Atmega2560 do fabricante ATMEL, que possui 4 interfaces de comunicação UART (*Universal Asynchronous Receiver/Transmitter*), 5 interfaces SPI e 1 interface I2C atendendo assim as necessidades de interfaces de comunicação para o trabalho proposto. Como suporte ao desenvolvimento com o microcontrolador, será utilizado a plataforma Arduino.

Nesta etapa foram implementados os algoritmos de avaliação dos dados recebidos do veículo, seguindo a abordagem realizada no trabalho de Naik (2017), onde o autor foca em analisar a violação dos valores pré-selecionados para os parâmetros em análise. Analisando os estudos realizados por Andria (2017) e Naik (2017), uma condução agressiva ocasiona desgaste prematuro dos componentes mecânicos e consumo elevado de combustível.

3.1.3 Etapa 3 - Envio

Para o envio dos dados, as aplicações de proposta semelhante vistas no referencial teórico na Tabela 5, propõem formas de envio de dados para as interfaces de visualização via cabos. Já as aplicações comerciais apresentadas anteriormente fazem o uso de comunicação *wireless* para o envio dos dados até a interface de visualização. Com base na análise das duas abordagens e na pesquisa realizada no referencial teórico, foi implementado neste trabalho, o registro os dados no SD Card e o uso da rede *Wi-Fi* para publicação dos dados salvos no SD Card em uma planilha eletrônica *online*.

No publicação dos dados na plataforma *online*, foi utilizado os dados contidos na seção 2.5 e também o trabalho de Tong (2017), onde o autor utiliza o módulo ESP8266 para executar métodos HTTP, publicando dados no sistema de planilhas Google *SpreadSheet*.

3.1.4 Etapa 4 - Análise dos dados

Por fim, na etapa de Análise dos Dados, será realizada a visualização dos dados. Para a análise dos dados, é possível utilizar duas abordagens de leitura do arquivo. A primeira consiste em abrir o conteúdo do cartão de memória diretamente em um computador, utilizando qualquer editor de planilhas eletrônicas. O arquivo em formato .CSV está organizado de forma que cada linha é um evento distinto contendo as informações de data, hora, latitude, longitude, temperatura do líquido de arrefecimento, RPM e velocidade.

A segunda forma consiste em, a partir da publicação automática na planilha *online* do Google *SpreadSheet* que o protótipo realiza, visualizar o arquivo *online* a partir de qualquer dispositivo.

Para manter o objetivo de fornecer um meio gratuito de visualização dos dados *online*, optou-se por utilizar o Google *Data Studio*. A ferramenta da Google consiste em uma ferramenta gratuita que transforma dados em relatórios e painéis informativos.

3.1.5 Testes funcionais

Durante o processo de testes funcionais, realizou-se testes isolados em bancada de cada etapa do sistema, utilizando os algoritmos desenvolvidos para cada função. Nesta etapa, foram realizados testes práticos de funcionamento e de depuração buscando o melhor desempenho da funcionalidade em questão.

3.1.6 Testes de Integração

Os testes de integração compreendem avaliar o sistema como um todo, unindo as etapas validadas nos testes funcionais em uma única aplicação, avaliando como se comporta o sistema funcionando com todos os módulos integrados.

Na etapa de integração, buscou-se através de testes de campo, avaliar a detecção dos eventos, registro no arquivo .CSV e a publicação dos dados no Google *Spreadsheet*, contemplando assim a integração do sistema. Para que fosse possível validar o protótipo no veículo e monitorar as variáveis pelo terminal serial, foi configurado no código para que um evento fosse gerado somente se o valor de RPM ultrapassasse a faixa de 2000 RPM. Desta forma, com o veículo ligado e parado, foi possível avaliar o funcionamento de todas as etapas de forma segura.

Após detectado o evento, verifica-se o conteúdo do SD *Card*, analisando se houve o correto registro de data, hora, latitude, longitude, velocidade, temperatura do líquido de arrefecimento e RPM. Para que seja realizada publicação *online* dos dados, é configurada uma rede *Wi-Fi* conhecida fornecida pelo roteador interno do *smartphone* e, verificado através do terminal serial se ocorre a conexão. Por fim, caso os dados publicados na planilha *online* tenham correspondência com os dados do arquivo .CSV gerado, o processo a integração está validado.

3.1.7 Testes de Validação

Com o sistema funcional e integrado são desenvolvidos os testes de validação, os quais esses últimos, correspondem aos resultados do trabalho. Para esta etapa de validação, foram desenvolvidos testes em campo comparando o desempenho do sistema com ferramentas comerciais disponíveis para coleta de dados.

4 DESENVOLVIMENTO DO PROTÓTIPO DE COLETA DE INFORMAÇÕES ATRAVÉS DA PORTA OBDII

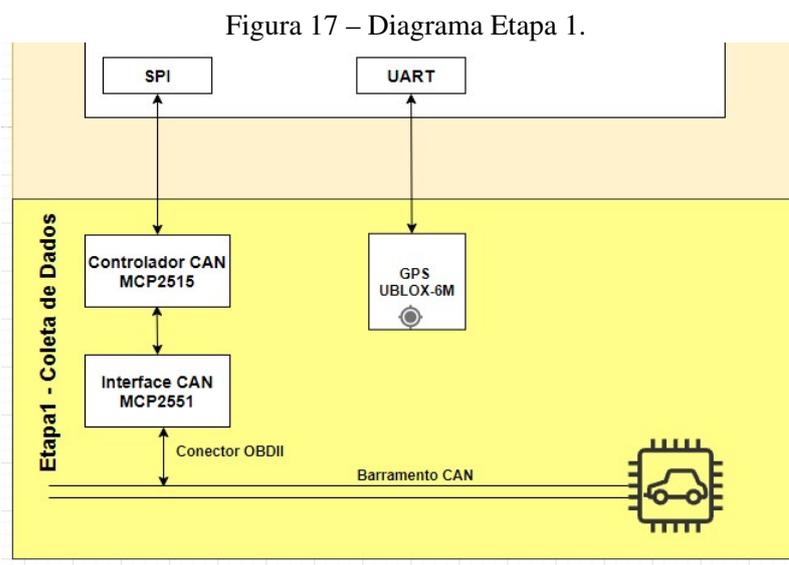
Neste capítulo, são apresentadas as implementações e resultados obtidos durante o desenvolvimento, seguindo a metodologia apresentada no capítulo anterior. Primeiramente será apresentado o desenvolvimento da etapa com os respectivos resultados dos testes funcionais. Na sequência, nos testes de integração, serão demonstrados os resultados dos testes de todas as etapas funcionando em conjunto. E por fim, nos testes de validação, são descritos os detalhes da preparação do ambiente de testes e os resultados obtidos.

4.1 ETAPA 1 - COLETA DE DADOS

A Etapa 1, consiste na aquisição dos dados disponibilizados através da porta OBDII do veículo e dados do receptor GPS.

4.1.1 Comunicação com o veículo

Para a implementação da comunicação com o barramento CAN do veículo, conforme a Figura 17, foi utilizado o controlador CAN MCP2515 e a interface CAN MCP2551 para implementar a camada física da comunicação entre o microcontrolador da aplicação e o barramento.



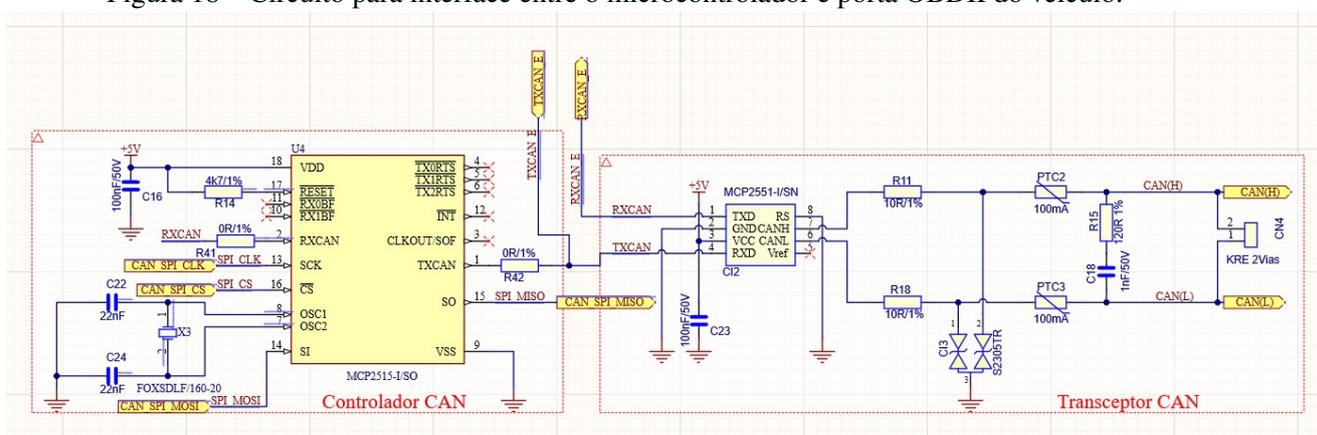
Fonte: O Autor.

Na Figura 18, pode-se visualizar parte do circuito utilizado para a conexão com o barramento CAN do veículo, contendo o controlador e transceptor CAN. No desenvolvimento desta parte do circuito, buscou-se seguir as recomendações do *datasheet* dos componentes quanto à função de cada pino, nível de tensão de alimentação e valores de cristal oscilador a ser utili-

zado. O controlador MCP2515 necessita de uma comunicação SPI com o microcontrolador Atmega2560. De acordo com o referencial bibliográfico, o circuito integrado MCP2515 é um *transceiver* CAN de alta velocidade, que opera com taxas de transmissão de até 1 Mb/s, e cumpre os requisitos da camada física descritos na norma ISO 11898.

Também, segundo boas práticas de circuito, foram previstas proteções na linha de conexão com a porta OBDII através do uso de resistores de 10R e PTC para atuarem como um fusível em caso de curto-circuito na linha. Para proteger quanto à pico de tensões fora da faixa do protocolo CAN, utilizou-se diodos do tipo TVS de dupla ação.

Figura 18 – Circuito para interface entre o microcontrolador e porta OBDII do veículo.



Fonte: O Autor.

Para a implementação da etapa de comunicação com o veículo, foi desenvolvido inicialmente um código que solicita ao veículo o valor do PID desejado. O início da comunicação com o barramento é feito utilizando funções para detectar a qual formato de identificador de mensagem CAN (11 ou 29) responde o barramento do veículo conectado, bem como a taxa de transmissão requerida. Após, utilizando a função `CAN.begin()`, passando como parâmetro a taxa de transmissão desejada, que no protótipo foi de 500 kbps. Então, são configurados os filtros para recebimento somente do endereço `0x7E8` que, conforme referencial bibliográfico, é o endereço de resposta da ECU às requisições de equipamentos externos. Caso não seja recebido uma resposta válida, ocorre erro de conexão com o barramento.

Após verificado o êxito na conexão, são percorridos os 96 códigos disponíveis para requisição, conforme a norma ISO 15031. Caso o valor retornado não seja nulo, é exibido no terminal o nome do PID. Utilizando o algoritmo da função `pidRead()`, é solicitado o modo de serviço, código PID, endereço para armazenar o valor e tamanho do valor.

- `pidRead(0x01, pid, value, sizeof(value))`

4.1.2 Leitura e interpretação dos dados GPS

Para o desenvolvimento da etapa de leitura do dados GPS, utilizou-se como referência o documento *Receiver Description Including Protocol Specification* (U-BLOX, 2013), onde são descritos detalhes sobre a interpretação do formato de dados NMEA, e também, detalhes da configuração do módulo NEO-6M-0-001. A comunicação com o módulo é feita através do periférico UART do microcontrolador configurando um *baud rate* de 9600 bps. Realizando a comunicação serial com o módulo, é recebida a mensagem padrão RMC (*Recommended Minimum Data*) do protocolo NMEA e, utilizando bibliotecas para interpretação dos dados, são extraídos os dados de latitude, longitude, data e hora. As demais informações contidas na *string* recebida não são utilizadas.

Na Tabela 7 são descritos detalhes sobre a *string* recebida considerando o formato do protocolo NMEA. Na Tabela são descritos somente os campos de interesse, os campos não descritos não são utilizados no presente trabalho.

- GPRMC,083559.00,A,4717.11437,N,00833.91522,E,0.004,77.52,091202,,A*57

Tabela 7 – Descrição dos campos contidos na mensagem NMEA-0183 RMC

Campo	Ex.	Formato	Nome	Unidade	Descrição
0	GPRMC	string	-	-	Identificador da mensagem
1	083559.00	hhmmss.sss	Hora	-	Hora - UTC h/min/s
3	4717.11437	ddmm.mmmm	Latitude	-	Latitude em graus + minutos
5	00833.91522	ddmm.mmmm	Longitude	-	Longitude em graus + minutos
9	091202	ddmmyy	data	-	Data em dia, mês e ano

Fonte: O Autor.

4.1.3 Testes Funcionais Etapa 1

A partir dos primeiros resultados da implementação do algoritmo de requisição dos dados do veículo, foram realizados testes práticos em 3 modelos de veículos diferentes, utilizando um algoritmo dedicado a solicitar os códigos PID disponíveis no veículo e exibindo no terminal serial a resposta vinda da ECU do veículo. Para a validação do algoritmo implementado, foi utilizado o *scanner* OBDII X60 da empresa AUTOOL¹. Após a instalação da ferramenta utilizando o conector OBDII, é possível percorrer através da tecla de seleção, as informações disponíveis no veículo, onde caso seja selecionado uma informação que não está acessível para o modelo de veículo, é indicado no *display* como indisponível. No Apêndice A, é apresentada uma imagem de referência da ferramenta utilizada para validação da etapa.

¹Empresa especializada em ferramentas *aftermarket* automotivas, informações disponíveis em <https://www.autooltech.com>

A Tabela 8 demonstra os resultados obtidos através da utilização do algoritmo desenvolvido durante o teste em 3 modelos distintos de automóveis. Observa-se também que, conforme a Tabela 8, cada fabricante de ECU disponibiliza informações distintas, onde dependendo da configuração do veículo, uma maior quantidade de informações é disponibilizada para requisição.

Tabela 8 – Códigos PID disponíveis de acordo com modelo e fabricante do veículo testado.

PID	Descrição	MODELO VEÍCULO		
		Ford Ka Sport	Hyundai HB20s	Renault Sandero RS
0	<i>PIDs supported [01 - 20]</i>	x	x	x
1	<i>Monitor status since DTCs cleared</i>	x	x	x
3	<i>Fuel system status</i>	x	x	x
4	<i>Calculated engine load</i>	x	x	x
5	<i>Engine coolant temperature</i>	x	x	x
6	<i>Short term fuel trim—Bank 1</i>	x	x	x
7	<i>Long term fuel trim—Bank 1</i>	x	x	x
11	<i>Intake manifold absolute pressure</i>	x	x	x
12	<i>Engine RPM</i>	x	x	x
13	<i>Vehicle speed</i>	x	x	x
14	<i>Timing advance</i>	x	x	x
15	<i>Intake air temperature</i>	x	x	x
17	<i>Throttle position</i>	x	x	x
19	<i>Oxygen sensors present (in 2 banks)</i>		x	x
20	<i>Oxygen Sensor 1</i>		x	x
21	<i>Oxygen Sensor 2</i>		x	x
28	<i>OBD standards</i>	x	x	x
31	<i>Run time since engine start</i>		x	x
32	<i>PIDs supported [21 - 40]</i>	x	x	x
33	<i>Distance traveled with (MIL) on</i>	x	x	x
36	<i>Oxygen Sensor 1</i>			x
46	<i>Commanded evaporative purge</i>		x	x
47	<i>Fuel Tank Level Input</i>		x	x
48	<i>Warm-ups since codes cleared</i>		x	x
49	<i>Distance traveled since codes cleared</i>		x	x
51	<i>Absolute Barometric Pressure</i>		x	x
51	<i>Catalyst Temperature</i>		x	
64	<i>PIDs supported [41 - 60]</i>		x	x
65	<i>Monitor status this drive cycle</i>			x
66	<i>Control module voltage</i>		x	x
67	<i>Absolute load value</i>		x	x
68	<i>Air-Fuel Ratio (lambda)</i>		x	x
69	<i>Relative throttle position</i>		x	x
70	<i>Ambient air temperature</i>		x	x

PID	Descrição	MODELO VEÍCULO		
		Ford Ka Sport	Hyundai HB20s	Renault Sandero RS
71	<i>Absolute throttle position B</i>		x	x
72	<i>Absolute throttle position C</i>			
73	<i>Absolute throttle position D</i>		x	x
74	<i>Absolute throttle position E</i>		x	x
76	<i>Commanded throttle actuator</i>		x	x
79	<i>Fuel–Air equivalence ratio</i>		x	x
92	<i>Engine oil temperature</i>			x

Fonte: O Autor.

Dentre os 3 modelos comparados, evidencia-se que as informações de interesse do presente trabalho, como temperatura do motor (PID05), RPM do motor (PID0C), velocidade do veículo (PID0D), estão presentes nos 3 veículos utilizados para estudo.

Ainda dentro dos testes da Etapa 1, no que diz respeito ao recebimento dos dados de geolocalização, foi desenvolvido um algoritmo dedicado a comunicar com o módulo GPS U-BLOX e, utilizando uma biblioteca dedicada, foram exibidos no terminal os resultados de latitude, longitude, data e hora.

Como forma de validar os resultados obtidos na leitura dos dados GPS, foi utilizado um *smartphone*² que possui o periférico GPS integrado e, utilizando um aplicativo para exibir os valores de latitude e longitude, comparada a leitura de ambos no mesmo ambiente, estando o protótipo e o *smatphone* lado a lado.

Conforme os resultados dos testes realizados e demonstrados na Tabela 9, a diferença verificada no valor de longitude entre o *smartphone* e o protótipo através da ferramenta *Google Maps*, corresponde à 6 metros entre os dois pontos, validando assim o desempenho do protótipo para o tipo de aplicação.

Tabela 9 – Tabela de comparação entre dados GPS do *Smartphone* e protótipo

Dispositivo de teste	Latitude	Longitude
Smartphone	-29.17575	-51.18468
Protótipo	-29.17575	-51.18479

Fonte: O Autor

4.2 ETAPA 2 - PROCESSAMENTO

A partir da implementação das funções de comunicação com o veículo e validação das mesmas, parte-se para a implementação da lógica de análise dos dados. Conforme o referencial bibliográfico, de forma semelhante ao trabalho de Naik (2017), foi implementada a lógica de

²Modelo: Motorola G8 Plus, Sistema operacional: Android 10

comparar os valores recebidos do veículo para as variáveis de interesse com valores predefinidos. Após os testes de validação da Etapa 1, foram definidos os PID de temperatura do motor (PID05), RPM do motor (PID0C), velocidade do veículo (PID0D) como sendo os parâmetros de interesse.

4.2.1 Definição dos parâmetros máximos dos PID

A faixa de rotação máxima de cada motor, varia conforme sua construção e potência específica, tendo na grande maioria dos automóveis populares valores máximos entre 6000 RPM e 7000 RPM. Conforme indicação do manual de usuário e boas práticas na direção, deve-se evitar rodar em altas rotações, para evitar um consumo excessivo de combustível e desgaste prematuro de componentes mecânicos. Para o presente trabalho, será utilizado para o (PID0C) como limite, o valor de 5000 RPM, por já ser considerada uma faixa elevada de rotação e que é geralmente atingida em condições fora do uso moderado do veículo.

A temperatura do motor (líquido de arrefecimento), serve para identificar o aquecimento do motor (período de enriquecimento) e os sobreaquecimentos (BRUNETTI, 2018). De acordo com consultas a manuais de operação veiculares, a temperatura operacional normal do motor de combustão interna está na faixa 87 °C a 105 °C. Com base nos referenciais, foi estipulado um valor limite de 100 °C para o PID05.

O limite para o PID0D, foram consideradas as informações contidas no Artigo 61 do CTB (Código de Trânsito Brasileiro) que diz que, onde não existir sinalização regulamentadora, a velocidade máxima em vias urbanas será de:

- Oitenta quilômetros por hora, nas vias de trânsito rápido;
- Sessenta quilômetros por hora, nas vias arteriais;
- Quarenta quilômetros por hora, nas vias coletoras;
- Trinta quilômetros por hora, nas vias locais.

Ainda conforme o Artigo 60 do CTB, Via Arterial é aquela caracterizada por interseções em nível, geralmente controlada por semáforo, com acessibilidade aos lotes lindeiros e às vias secundárias e locais, possibilitando o trânsito entre as regiões da cidade. Com base nos dados levantados, para o presente trabalho foi considerado o valor de 60 km/h como limite máximo para o PID0D.

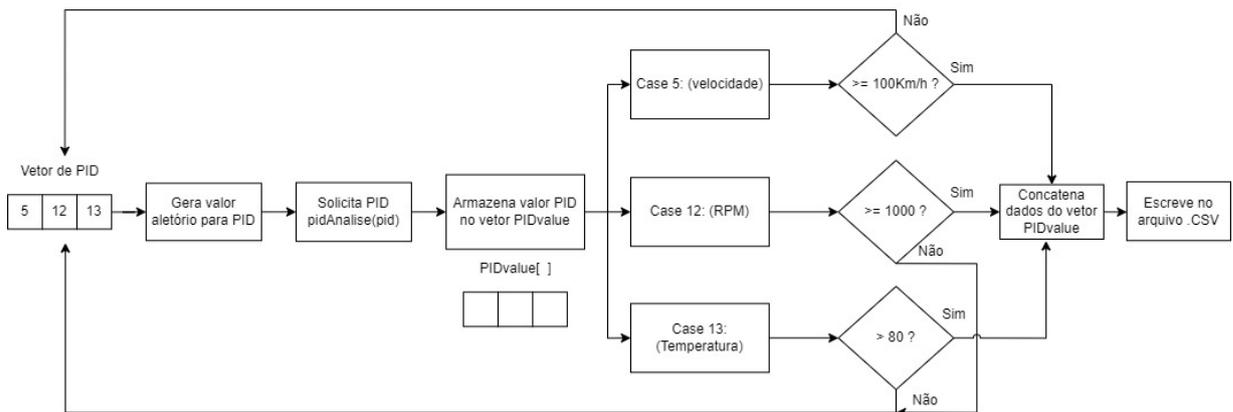
4.2.2 Algoritmo de comparação

O algoritmo implementado para a comparação dos valores, utiliza um laço condicional para varrer o vetor de PID, e armazenar os valores retornados da função de requisição de PID em um vetor secundário. A cada iteração, de acordo com o PID lido no momento, é realizada a

comparação com o valor limite definido. Caso o valor recebido seja maior, é feita a chamada da função de registro dos dados no *SD Card* detalhada na próxima seção do trabalho.

Para o que o processo de implementação e testes fosse possível de ser feito em bancada sem a necessidade de estar conectado ao veículo, foi utilizado uma função que gera valores pseudoaleatórios para a variável que armazena os valores de PID. Na Figura 19, é possível verificar o fluxo utilizado para testes e implementação da etapa em detalhes.

Figura 19 – Fluxograma do algoritmo de validação da detecção de evento e escrita no *SD Card*.



Fonte: O Autor.

4.2.3 Testes Funcionais Etapa 2

Para a validação da Etapa 2 e teste do algoritmo desenvolvido, foi realizado o teste em bancada simulando os valores da variável de controle e verificando através do terminal serial, se as condições de chamada da função de escrita no *SD Card* eram atendidas.

Utilizando um algoritmo pseudoaleatório para gerar os valores da variável de controle, foi possível observar que a lógica desenvolvida é funcional, uma vez que a chamada da função de registro é realizada somente ao ultrapassar os valores definidos para cada código PID.

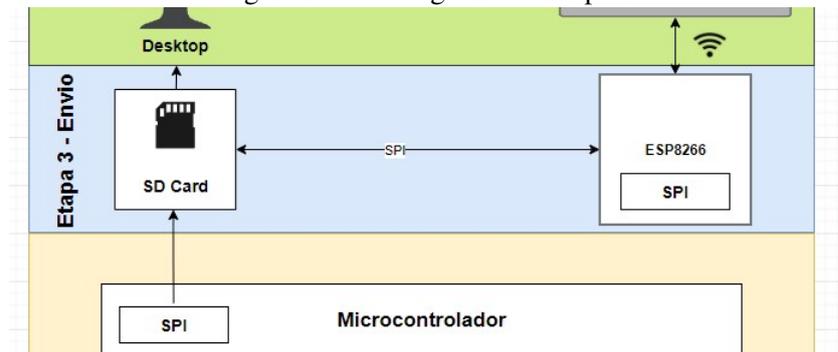
4.3 ETAPA 3 - ENVIO

Na Etapa 3, trata-se do desenvolvimento do registro dos dados no *SD Card* e publicação dos mesmos na plataforma *online*, seguindo a lógica retomada no fluxograma da Figura 20.

4.3.1 Registro dos dados no *SD Card*

No desenvolvimento do registro dos dados no *SD Card*, foi desenvolvida a lógica de forma que, caso o PID em questão ultrapasse o valor predefinido, ocorra o registro do vetor inteiro no arquivo *.CSV*. Desta forma, independentemente do valor que gere o evento, tem-se as demais variáveis para que seja feito um cruzamento de dados durante a análise.

Figura 20 – Fluxograma da Etapa 3.



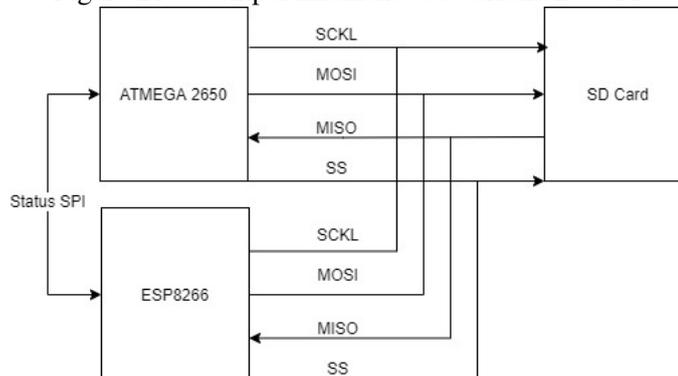
Fonte: O Autor.

A cada chamada da função de registro dos dados no arquivo .CSV, o vetor contendo os valores dos PIDs, é concatenado e salvo no *SD Card* em um arquivo no formato .CSV (*Comma Separated Value*) juntamente com os dados de data, hora, latitude, longitude utilizando funções de escrita em arquivos.

A comunicação com o *SD Card* foi feita utilizando o periférico SPI disponível no microcontrolador Atmega2560. Conforme visto na Figura 21, a comunicação SPI não permite que dois dispositivos do tipo *MASTER* acessem o mesmo dispositivo *SLAVE* simultaneamente, portanto foi necessário desenvolver uma lógica de compartilhamento do barramento com o módulo ESP8266 no *firmware* e no *hardware*.

No *hardware*, foi necessário utilizar um pino como saída no microcontrolador Atmega2560 conectado à uma entrada do ESP8266. Para o *firmware* foi necessário desenvolver uma lógica onde, ao obter as condições para o início da leitura do *SD Card* pelo ESP8266, o periférico SPI é desabilitado no microcontrolador principal. Esta condição é informada ao módulo através do nível alto no pino de comunicação entre os dois dispositivos. Assim que a leitura é finalizada, o barramento SPI é liberado pelo ESP8266 desabilitando seu periférico e, através do pino de comunicação, é informado que a leitura terminou ao microcontrolador principal, onde o mesmo está apto a comunicar com o *SD Card* novamente.

Figura 21 – Compartilhamento do barramento SPI.



Fonte: O Autor.

4.3.2 Publicação dos dados no Google *Spreadsheet*

Para obter a conexão com a internet e realizar o envio dos dados, o componente escolhido foi o módulo *Wi-Fi* ESP8266. O ESP8266 é um SoC (*System-on-a-Chip*) com tecnologia *Wi-Fi* embarcada disponibilizando uma ampla variedade de modelos. Entretanto o escolhido para o projeto é o ESP8266 NodeMCU V2 por apresentar até 10 pinos de I/O, saídas PWM, I2C, UART e antena integrada ao próprio módulo.

Para que seja possível o acesso a internet do módulo ESP8266, foram utilizadas bibliotecas para manuseio do periférico *Wi-Fi*. O microcontrolador do módulo foi configurado para operar em modo de *station*, que é o modo onde o dispositivo se conecta a um *access point* com acesso a internet.

Para a transmissão dos dados, optou-se pela tecnologia *Wi-Fi*, visto que se trata de uma tecnologia de fácil acesso, e a maioria dos *smartphones* podem operar como um roteador, fornecendo assim conexão do dispositivo ao servidor desejado para a transmissão dos dados.

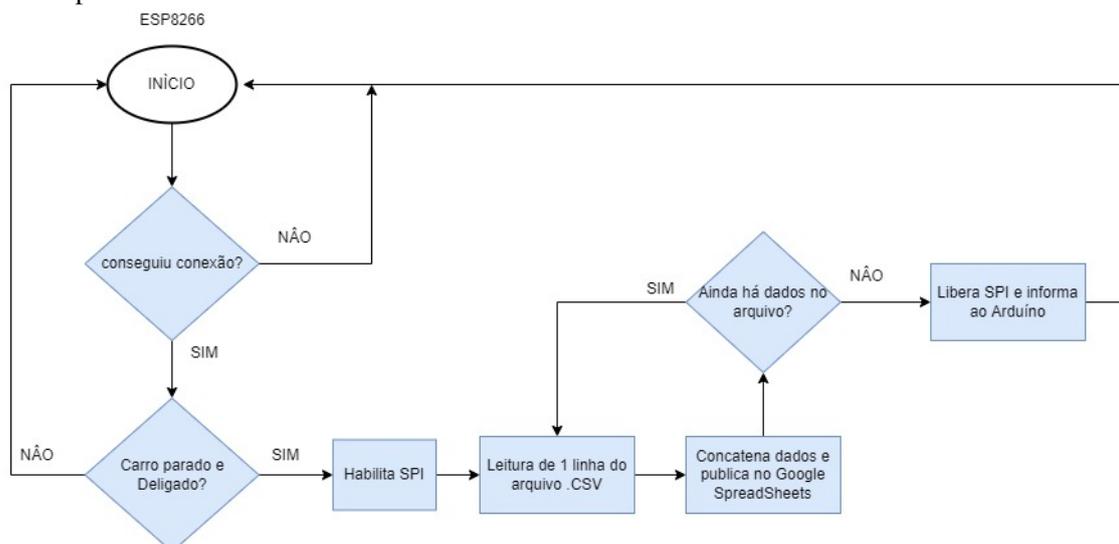
O algoritmo do microcontrolador principal, ao detectar que o carro está parado e desligado, ou seja, RPM = 0 e velocidade = 0, informa ao ESP8266 que a leitura e envio do arquivo pode iniciar. Caso o ESP8266 tenha sucesso na conexão com a rede *Wi-Fi* configurada, inicia o processo de leitura do arquivo *.CSV* armazenado no *SD Card*.

De forma semelhante ao trabalho de Tong (2017), foi criado uma planilha na ferramenta Google *Spreadsheet*, contendo os mesmos nomes de colunas contidos no arquivo *.CSV*. Após, foi armazenado a chave URL da planilha que fica contida entre os caracteres `"/d"` e `"/edit"`, e também, renomeado o arquivo. Na sequência, foi necessário criar um Google *app script* disponível dentro da própria planilha, que será responsável por receber os dados do ESP8266 e escrever nas linhas e colunas corretas. Para que ocorra o *link* correto entre o *script* e a planilha, é necessário informar a chave URL da planilha anteriormente copiada. Publicando o *script* como uma aplicação *Web*, a URL gerada será utilizada na requisição pelo ESP8266 durante o envio. No Apêndice B é possível observar mais detalhes do código para o *script* utilizado para publicar na planilha.

Após receber o sinal vindo do microcontrolador principal Atmega2560 pelo pino de *statusSPI*, de que o barramento está livre para acesso, o algoritmo implementado inicia a leitura de uma linha inteira do arquivo. Na sequência, concatena os dados em forma de uma única *string* juntamente com a URL de acesso ao *script* realizando a publicação na página gerada através do comando GET.

Na Figura 22 é possível visualizar o fluxograma do algoritmo implementado no ESP8266 para a leitura e envio dos dados salvos no cartão de memória. Através da legenda, é possível verificar que este processo está inserido na Etapa 3.

Figura 22 – Fluxograma de funcionamento do algoritmo implementado para leitura e publicação dos dados na planilha *online*.



Fonte: O Autor.

4.3.3 Testes Funcionais Etapa 3

Os testes de validação da Etapa 3, foram realizados em bancada utilizando como forma de acompanhamento e validação do código o monitor serial, onde são impressos os resultados de pontos chave do código afim de validar cada subetapa do mesmo. Além dos resultados obtidos pelo monitor serial, foi utilizada uma planilha de testes criada no Google *Spreadsheets*.

Para a validação do registro dos dados no *SD Card*, foi utilizado o monitor serial para imprimir a *string* utilizada pela função de escrita após o seu uso. Na validação do processo de envio dos dados, foi utilizado também o monitor serial para imprimir o resultado da concatenação da linha do arquivo lida, sendo possível validar o resultado da função de concatenação implementada.

Foi possível verificar que, a conexão com a página *Web* foi bem sucedida, através de uma mensagem de retorno recebida da página criada após receber a requisição. Após, foi possível verificar que a *string* foi publicada com sucesso na planilha, comparando os dados impressos no terminal com os visualizados na planilha *online*.

4.4 ETAPA 4 - ANÁLISE DOS DADOS

4.4.1 Análise dos dados utilizando Google *Data Studio*

Na Etapa 4 que trata da análise dos dados, conforme mencionado na metodologia, foi utilizado o Google *Data Studio* como ferramenta gratuita de análise de dados. Para montar o *dashboard*, foi utilizado como referência, as interfaces presentes nas soluções comerciais apre-

sentadas na Tabela 6.

O *dashboard* desenvolvido conta com uma visualização em forma de mapa, contendo o registro dos eventos, onde nos pontos marcados, é possível verificar o valor da variável desejada. Desta forma, o usuário saberá que, se há um ponto marcado no mapa, significa que naquela coordenada ao menos uma das três variáveis de interesse (velocidade, temperatura do motor ou RPM), ultrapassaram os valores definidos como limite.

O Google *Data Studio* aceita diversas fontes de dados como banco de informações para os *dashboards*. Para utilizar como banco de dados uma planilha do Google *SpreadSheet* é necessário que a planilha pertença a mesma conta que está acessando a ferramenta.

A visualização em forma de mapa, consiste em a partir de dados em forma de latitude e longitude, marcar pontos no mapa para análise das métricas relacionadas. Para que a interpretação das coordenadas fosse feita de forma correta, foi necessário criar um campo adicional concatenando as colunas de latitude e longitude separadas na planilha de banco de dados, em um único campo utilizando a fórmula: (CONCAT(latitude, CONCAT(', ', longitude))). O novo tipo de dado, derivado das duas coordenadas, foi chamado de "localização" e utilizado como métrica da visualização de mapa.

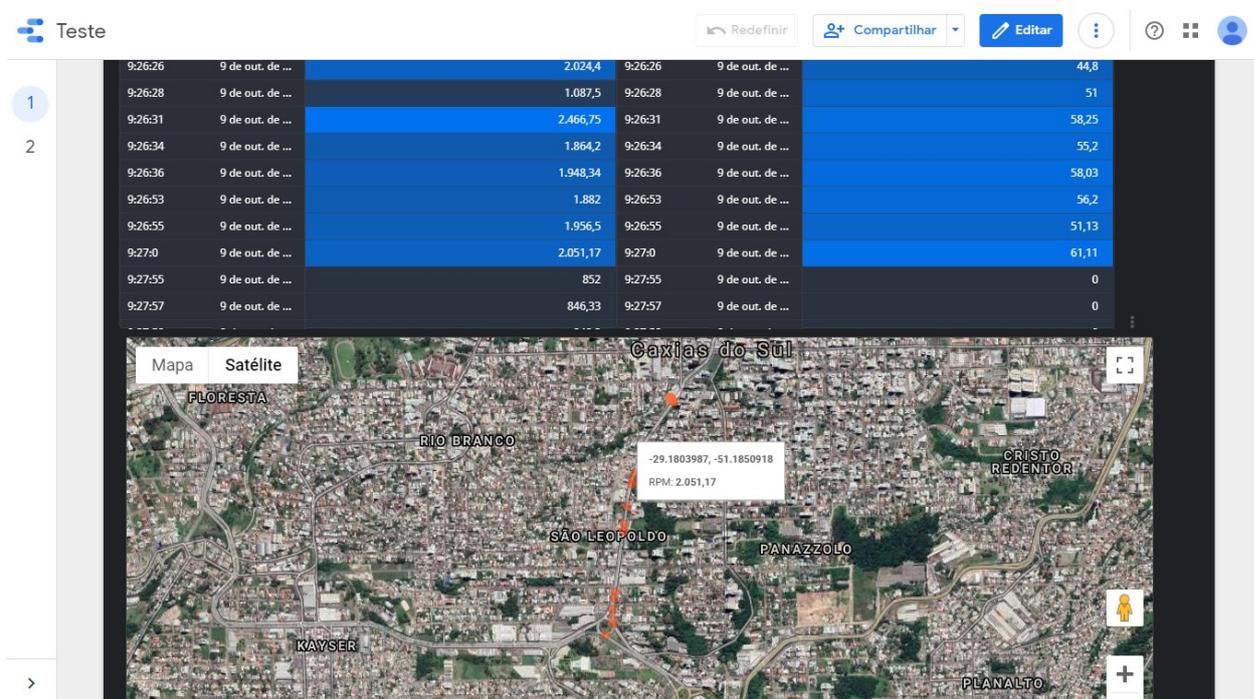
Outra ferramenta utilizada para exibir os dados contidos na planilha de forma gráfica, foi a ferramenta de tabela dinâmica com mapa de calor. Esta ferramenta possibilita adicionar níveis de cores às linhas da tabela de acordo com o valor da variável.

4.4.2 Testes Funcionais Etapa 4

Como forma de validação da Etapa 4, foi realizada a comparação dos dados contidos no arquivo .CSV salvo durante os testes anteriores no SD *Card*, a planilha de testes criada no Google *Spreadsheet* e o modelo de testes do *dashboard* desenvolvido.

Na Figura 23, é possível observar o protótipo do *dashboard* desenvolvido durante os testes, onde na tabela à direita, tem-se a hora, data e registro de RPM e à esquerda, a hora, data e velocidade. Na visualização em forma de mapa, tem-se a coordenada do registro do evento. Estes dados foram comparados com os presentes na planilha de consulta do *dashboard* e também com os presentes no arquivo .CSV aberto utilizando um leitor de planilhas.

Figura 23 – *dashboard* desenvolvido utilizando os dados publicados no Google *SpreadSheet* como banco de dados.



Fonte: O Autor.

4.5 DESENVOLVIMENTO DO *HARDWARE* DO PROTÓTIPO

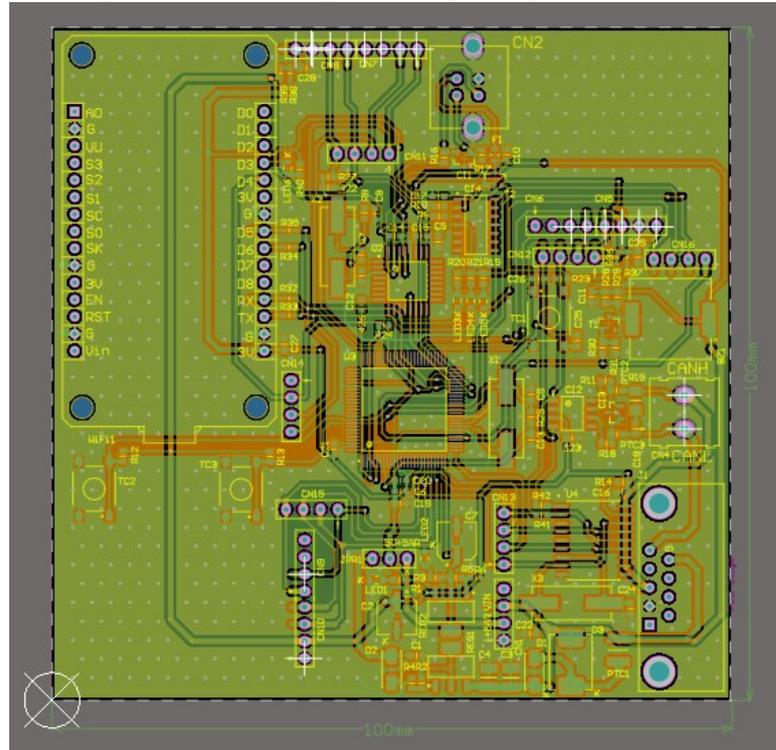
Para servir de suporte ao desenvolvimento do trabalho e integrar o *hardware* até então testado individualmente, fez-se necessário o desenvolvimento de uma placa de circuito impresso contendo os circuitos necessários. Para atender a demanda de energia dos módulos utilizado, a placa contempla barramento de tensão de 3,3 Vdc e 5 Vdc. O módulo GPS foi conectado ao periférico UART do microcontrolador ATMEL, conforme recomendado na documentação.

Para a conexão com o *SD Card* foi necessário utilizar um dos periféricos SPI disponíveis, conectando de forma paralela, o módulo ESP8266 devido à necessidade de compartilhar o acesso ao *SD Card*. Além de servir como suporte aos módulos, a placa desenvolvida conta com duas teclas, um *buzzer* para avisos sonoros, barras de pinos para prototipação e o conector necessário para a interface OBDII. Nas Figura 24 e Figura 25 é possível visualizar a placa de circuito impresso desenvolvida em detalhes.

4.5.1 Validação do *Hardware*

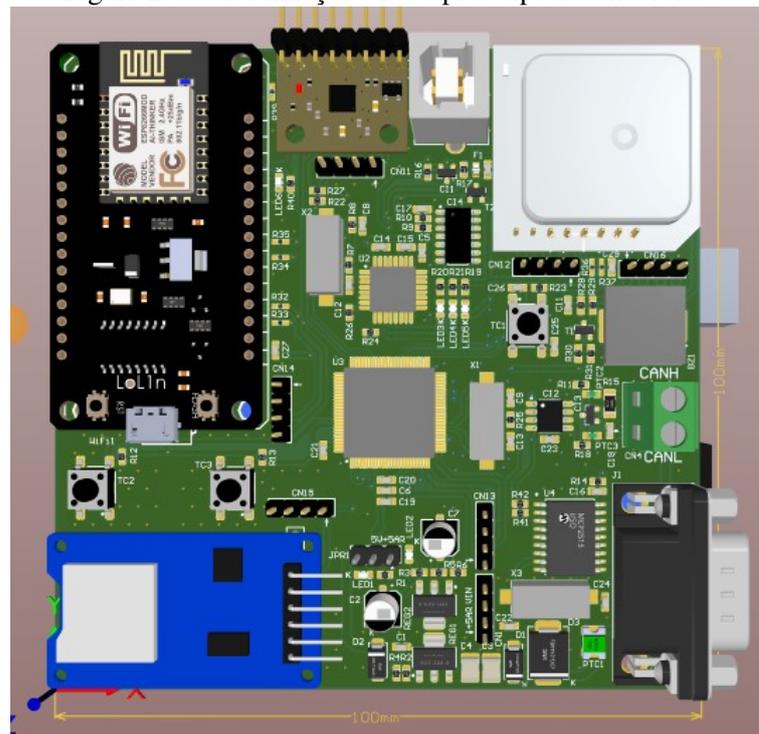
A validação da placa de circuito impresso desenvolvida, foi feita repetindo os testes funcionais executados nas Etapas 1,2,3 e 4, buscando obter os mesmos resultados quanto ao funcionamento do *hardware*. A validação foi feita comparando os resultados anteriormente obtidos com os desta etapa.

Figura 24 – Visualização 2D do protótipo desenvolvido.



Fonte: O Autor.

Figura 25 – Visualização 3D do protótipo desenvolvido.



Fonte: O Autor.

Após a execução dos testes de cada etapa isolada em bancada, foi verificado utilizando as mesmas formas de validação das etapas anteriores, que o *hardware* desenvolvido está funcionando conforme os resultados anteriormente obtidos.

4.6 TESTES DE INTEGRAÇÃO

Os testes de integração, consistem em realizar os ensaios com os subsistemas descritos nas etapas anteriores funcionando em conjunto, tendo assim a integração do *hardware* e *firmware* até então desenvolvidos.

Na integração do *firmware* até então desenvolvido e testado de forma isolada, buscou-se implementar o código de forma que cada macro ação fosse uma função dentro do código, como por exemplo, a etapa de registro de dados no *SD Card* anteriormente testada em um código separado, foi transformada em uma função que recebe o conteúdo a ser escrito no arquivo *.CSV*. Na Figura 27, é possível visualizar um fluxograma completo do algoritmo desenvolvido para o microcontrolador principal, contendo as referências de cor de cada etapa descrita na seção 3.1.

Conforme a metodologia de teste, para a validação do sistema, buscou-se realizar a detecção de um único evento realizando o teste com o protótipo conectado ao veículo. Após realizada a validação em bancada do *firmware* e *hardware* desenvolvido, o protótipo foi conectado ao veículo para o teste de campo. Na Figura 26 é possível verificar detalhes da alocação do protótipo durante os testes. No Apêndice C, são demonstrados mais detalhes do cenário de testes e localização da porta OBDII no veículo.

4.6.1 Resultados Testes de Integração

Conforme metodologia de testes descrita na seção 3.1.6, ao acelerar o veículo parado, de forma a elevar o RPM até o valor de 2000 conforme o painel de instrumentos, foi detectado através do monitor serial que ocorreu o registro de um evento no arquivo *.CSV*, conforme o esperado.

Após a etapa de coleta descrita acima, verificando o conteúdo do arquivo registrado no *SD Card* através de um leitor de planilhas, a etapa de registro é aprovada após constatar que o arquivo está completo, contendo todas as colunas separadas pelo carácter vírgula. Os campos relacionados aos dados coletados pelo módulo GPS, estavam zerados, pois conforme observado, o receptor GPS sofre grande interferência em garagens e ambientes fechados. Repetindo o mesmo teste com o veículo estacionado na rua, foram obtidos os dados do módulo GPS corretamente.

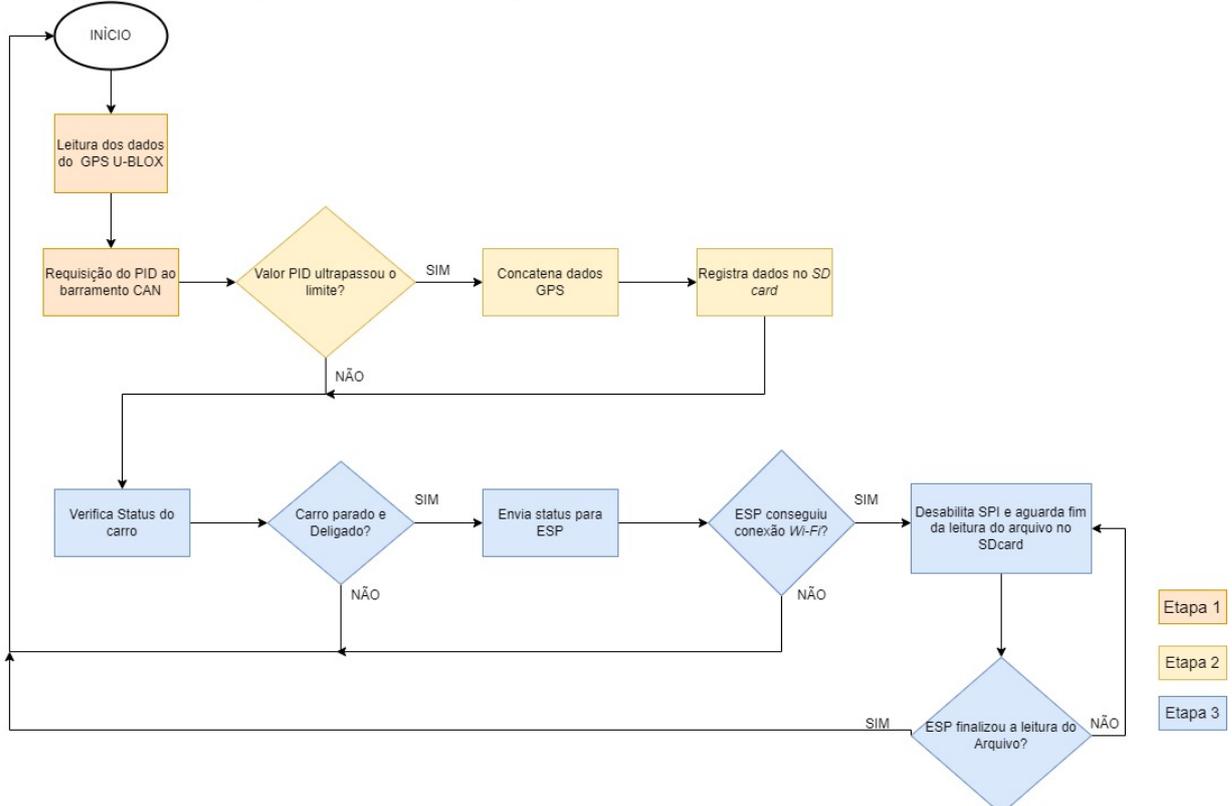
Ao desligar o veículo, mantendo o RPM = 0 e velocidade = 0, o algoritmo entra em modo de conexão, onde informa ao ESP8266 que pode iniciar a leitura de envio do conteúdo do arquivo para a planilha do Google *SpreadSheet*. Ao conseguir conexão com o roteador de *smartphone*,

Figura 26 – Protótipo montado no veículo Ford Ka Sport para teste de integração.



Fonte: O Autor.

Figura 27 – Fluxograma completo do algoritmo desenvolvido para o microcontrolador principal.



Fonte: O Autor.

foi utilizado o monitor serial para verificar o *status* da conexão e início da leitura dos arquivos conforme esperado. Acessando a planilha *online*, foi possível verificar que os dados estavam sendo publicados corretamente conforme o conteúdo presente no arquivo .CSV.

4.7 TESTES DE VALIDAÇÃO

Com o sistema funcional e integrado são desenvolvidos os testes de validação, os quais correspondem aos resultados do trabalho de acordo com os objetivos específicos sendo eles:

- Implementar um algoritmo de leitura dos parâmetros de temperatura do líquido de arrefecimento, rotação do motor e velocidade através do acesso à rede CAN pela porta de comunicação OBDII, gerando o registro quando um desses parâmetros ultrapasse o valor determinado;
- Ler informações de geolocalização através de um módulo GPS para integrar as informações de latitude, longitude, hora e data coletadas, aos registros de eventos;
- Realizar o registro dos dados em um cartão de memória, e quando tenha conexão de rede *Wi-Fi (Wireless Fidelity)*, publicar os dados salvos na memória para uma planilha eletrônica *online* integrada com um *dashboard* na plataforma *Google Data Studio*;
- Comparar o desempenho da solução com relação a aquisição dos dados, com ferramentas similares e de baixo custo encontradas no mercado.

De acordo com as soluções comerciais pesquisadas e demonstradas na Tabela 6, a interface de *hardware* utilizada por estas soluções seguem o mesmo conceito. Como forma de conexão com o veículo, são utilizados *scanners* conectados diretamente na porta OBDII do veículo. Estes dispositivos, são responsáveis por realizar as requisições de informação, e em modelos que utilizam a tecnologia GPRS, enviar para um servidor de dados. De forma semelhante, há também ferramentas de diagnóstico acessíveis para usuários sem treinamento específico ou pagamento de mensalidades que são os *scanners* com conexão *bluetooth*. Dentro desta proposta, existem diversos dispositivos disponíveis no mercado que, após conectados a porta de diagnóstico OBDII e pareados com um *smartphone*, exibem através de aplicativos os dados disponíveis para o veículo em questão.

Ainda dentro da proposta de serviços gratuitos para o monitoramento dos dados do veículo, existem também aplicativos para *smartphone* que através de dados inseridos pelo usuário, sinal de GPS e acelerômetro, são capazes de fornecer estatísticas sobre o uso do veículo, como velocidade e localização em tempo real. Estas soluções não são conectadas fisicamente ao veículo através da porta OBDII, sendo limitadas na quantidade de informações disponíveis ao usuário.

Como forma de validação do *hardware* e *firmware* do protótipo desenvolvido, foram realizados testes comparativos com a ferramenta de diagnóstico RaceChrono. Esta ferramenta, trata-se

de um aplicativo para *smartphone*, que serve como interface visual para os dados coletados através do *scanner* *OBDA Advanced* ligado à porta OBDII do veículo. Os dados de velocidade, RPM e temperatura, são coletados através do *scanner* e os dados de GPS, obtidos a partir receptor do próprio *smartphone* e tratados pelo aplicativo. O aplicativo é capaz de fornecer uma visualização em forma de mapa, contendo a velocidade, RPM e temperatura registrada em determinada latitude e longitude.

Então, como forma de validação, foram comparados os dados obtidos pelo protótipo, o aplicativo apresentado e o painel de instrumentos do veículo.

4.7.1 Cenário de testes

Através do aplicativo RaceChrono, é possível ter uma visualização em forma de mapa contendo a velocidade atingida em determinado ponto. A forma de marcação dos pontos no mapa, difere da desenvolvida neste trabalho, de forma que no aplicativo, deve-se marcar os pontos de controle previamente antes de iniciar o trajeto. Neste trabalho, os pontos são marcados no mapa a cada registro de evento, sendo eles velocidade, RPM ou temperatura acima dos parâmetros limite. Outra diferença observada é que no aplicativo, é exibido abaixo do mapa, as coordenadas de latitude e longitude, RPM e temperatura durante todo o trajeto registrado. Durante os testes, não foi possível conectar o protótipo e o *scanner* simultaneamente à porta OBDII do veículo, devido ao mesmo só possuir um conector para este fim. Desta forma, buscou-se avaliar se o protótipo é capaz de coletar e exibir o mesmo tipo de dados que o da ferramenta de comparação durante o mesmo trajeto predefinido, sendo testado uma aplicação por vez.

Na Figura 28, é possível visualizar o trajeto definido para os dois testes de rodagem. Foi percorrido um roteiro urbano de aproximadamente 1,7 km.



Fonte: O Autor.

- RaceChrono - Primeiramente, foi conectado o *scanner* *OBDA Advanced* à porta OBDII do veículo de testes, sendo este um Ford Ka Sport. Após realizar a conexão via *bluetooth* com o aplicativo RaceChrono, são exibidas as informações lidas do veículo através

da conexão OBDII. Então, após a demarcação dos pontos de controle no mapa onde serão registrados os valores de velocidade, inicia-se o trajeto. Os dados obtidos durante o trajeto, são salvos na memória do próprio *smartphone* e acessados através do aplicativo.

- Protótipo - Após realizada a coleta dos dados utilizando o aplicativo e *scanner* descritos anteriormente, parte-se para o teste utilizando o protótipo desenvolvido. Neste teste, com o protótipo conectado a porta OBDII do veículo, é percorrido o mesmo trajeto do teste anterior buscando manter um estilo de condução semelhante. Ao fim do percurso, com o veículo parado e conectado uma rede *Wi-Fi*, a leitura dos dados contidos no *SD Card* é iniciada pelo módulo ESP8266. Enquanto ocorre a leitura de cada linha do arquivo *.CSV*, é realizada a publicação na planilha do *Google Spreadsheet* que serve de banco de dados para o *dashboard* desenvolvido.

4.7.2 Resultados dos Testes de Validação

A partir da análise dos dados do *dashboard*, o qual tem como banco de dados esta planilha, busca-se comparar as informações em pontos de localização aproximados nos dois testes, com o objetivo de verificar se o *dashboard* desenvolvido contém o mesmo nível de informações disponível no aplicativo. Na Tabela 10, são demonstrados os resultados dos dois testes considerando os dados registrados em uma coordenada comum aos dois testes. A partir da tabela, é possível verificar que o *dashboard* desenvolvido é capaz de entregar o mesmo tipo de informação que o aplicativo utilizado. Outro ponto validado é a capacidade do protótipo coletar informações do veículo através da porta OBDII.

Tabela 10 – Resultado teste comparativo Aplicativo RaceChrono e Protótipo.

Variável	RaceChrono	Protótipo
Latitude	-29,17361	-29,17360
Longitude	-51,18882	-51,18879
Velocidade	46 km/h	41 km/h
RPM	2593	2023
Temperatura Motor	95 °C	100 °C

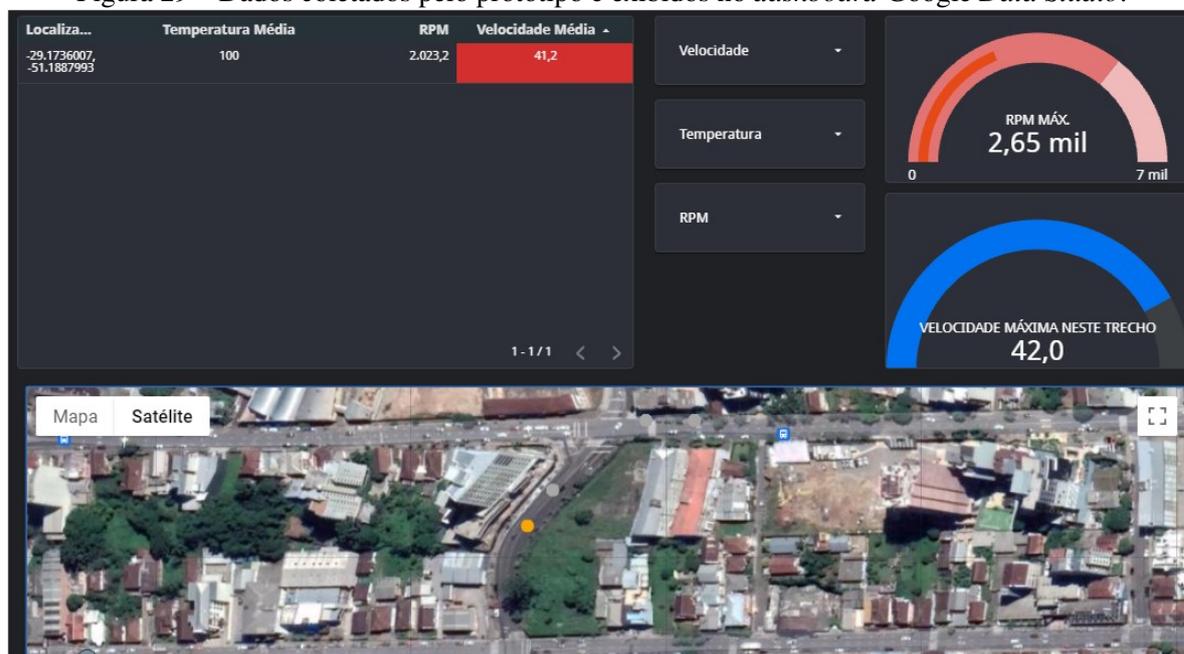
Fonte: O Autor.

Na Figura 29, é possível observar os dados disponíveis no *dashboard* desenvolvido no presente trabalho para uma coordenada aproximada ao da registrada no aplicativo. Na aplicação desenvolvida, tem-se a visualização de mapa contendo a marcação visual dos pontos de registro de eventos. Ao clicar em algum dos pontos em laranja registrados no mapa, é possível observar a velocidade, RPM e temperatura do motor média para aquele conjunto de registros.

O GPS realiza a atualização de dados a cada um segundo, desta forma são realizados mais do que um registro no *SD Card* com a mesma coordenada geográfica até ter uma nova atualização. Então, na visualização em forma de mapa no *dashboard*, é marcado somente um ponto para

cada conjunto de ocorrências contendo a mesma coordenada. Para filtrar dentro do registro com a mesma coordenada quais os máximos e mínimos daquele ponto, deve-se clicar na marcação de interesse e, a tabela geral do *dashboard* é filtrada para aquela ocorrência, exibindo os dados de temperatura média, RPM e velocidade daquele conjunto de registros. Nos mostradores a direita, são exibidos os valores máximos de RPM e velocidade do conjunto de registros. Outra forma de filtrar os dados, é através das caixas de seleção centrais, filtrar por determinados valores de velocidade, temperatura e RPM, sendo assim exibidos no mapa onde ocorreram e na tabela, somente os resultados do filtro aplicado.

Figura 29 – Dados coletados pelo protótipo e exibidos no *dashboard* Google Data Studio.



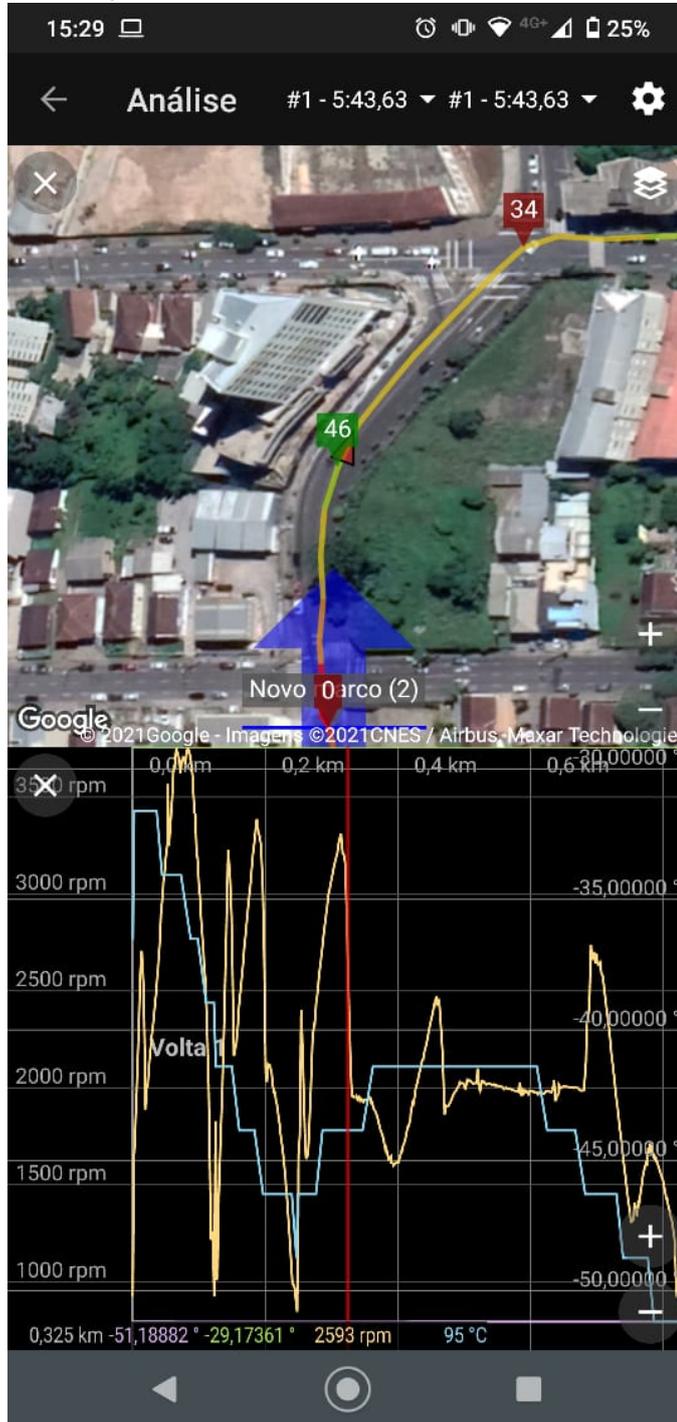
Fonte: O Autor.

Na Figura 30, é possível observar as informações disponíveis utilizando o aplicativo RaceChrono em conjunto com o *scanner* OBD *Advanced*. A forma de exibição das informações no aplicativo diferem da desenvolvida no protótipo, pois conforme mencionado, o aplicativo coleta dados em pontos demarcados manualmente no mapa. Os dados que formam a visualização de gráfico, são coletados de forma contínua durante todo o trajeto. Outro ponto é que este aplicativo em específico, trabalha com trajetos em forma de circuito, onde deve ser marcado pelo menos um ponto final e inicial no mapa.

Foi realizado também o comparativo visual com o *dashboard* da ferramenta Carrorama³, onde no Anexo B, é possível visualizar eventos registrados por esta solução que foram detectados durante o uso do veículo, como por exemplo velocidade excessiva, marcados no mapa. Também é possível, de forma semelhante ao protótipo, filtrar os registros por períodos específicos de data e hora.

³Empresa especializada em gestão de frotas, informações disponíveis em <https://carrorama.net/planos/>

Figura 30 – Dados coletados no RaceChorno.



Fonte: O Autor.

5 CONSIDERAÇÕES FINAIS

O presente trabalho, consistiu no desenvolvimento de um protótipo de coleta de informações via porta OBDII do veículo e, disponibilização dos dados coletados em um *dashboard online*. Para isso, foram utilizados circuitos de conexão com o barramento CAN veicular, extraindo os dados de interesse e através da tecnologia *Wi-Fi*, publicado os dados em uma planilha do *Google Spreadsheets* para posterior visualização e análise via *Google Data Studio*.

A partir do que foi apresentado no desenvolvimento do trabalho e nos resultados obtidos, os objetivos do trabalho foram alcançados, de forma que o protótipo propõe uma solução de baixo custo capaz de fornecer informações importantes sobre o uso do veículo à gestores de pequenas frotas, através de ferramentas gratuitas. Conforme mencionado no decorrer do desenvolvimento do trabalho, as soluções encontradas em artigos de propostas similares, apresentavam meios de análise dos dados pouco usuais e acessíveis a maioria dos usuários e público interessado na gestão de frotas. As ferramentas como o LabView e o MatLab utilizadas para a análise dos dados nestes trabalhos, são focadas para o âmbito acadêmico. Já as soluções comerciais com serviços já consolidados na gestão de frotas apresentadas, e utilizadas como referências, são soluções focadas em atender a frotas com número expressivo de veículos, fornecendo ferramentas e métricas avançadas, onde estas estão disponíveis por meio de planos de mensalidades pagas pelo usuário.

Com a abordagem utilizada no protótipo aqui elaborado, o gestor poderá visualizar através do *dashboard* como seu veículo está sendo utilizado. É possível verificar se houve excesso de velocidade, detectar uma condução agressiva através do RPM elevado do motor e, a manutenção básica do veículo por meio da temperatura do líquido de arrefecimento, onde caso ocorram registros de valores acima do considerado seguro e com alta frequência de ocorrências, pode indicar falta de fluido neste sistema.

Para criar uma ferramenta de diagnóstico genérica e ter dados que representem o comportamento do motorista na condução do veículo, foi necessário definir quais os códigos PID seriam utilizados. O teste funcional da etapa de leitura dos dados do veículo, foi fundamental para verificar que, apesar de o protocolo OBDII fornecer uma ampla gama de dados, nem todos os veículos disponibilizam a totalidade dos códigos PID existentes. Outro ponto é que, alguns códigos PID trazem informações específicas dos sistemas do veículo, não representando a ação do condutor para a variação daquele parâmetro, como por exemplo, a temperatura de admissão ou pressão de combustível. Por tanto a escolha da velocidade do veículo e RPM, remetem a ações diretas do condutor para sua variação de valor. Já o parâmetro de temperatura do motor, é um parâmetro importante relacionado à mecânica do veículo mas que também, trás informações sobre o cuidado do usuário quanto a manutenção do mesmo.

Com relação ao sistema de localização via GPS, utilizado para coletar informações de latitude e longitude, data e hora, de cada evento registrado, foi detectado que a posição da antena dentro do veículo é de grande influência para a aquisição do sinal de rádio. O sinal de GPS, so-

fre grande atenuação dentro de ambientes fechados, como garagens de edificações e túneis. Ao posicionar o protótipo dentro do veículo de forma a melhorar a visada da antena, foi percebido uma redução no tempo de aquisição inicial dos dados de GPS.

Durante a etapa de publicação dos dados, a escolha de publicar os dados salvos no *SD Card* em formato *.CSV* em uma planilha *online*, forneceu flexibilidade na forma de analisar os dados, sendo possível ter o mesmo conteúdo de forma *online* ou somente abrindo o arquivo diretamente do *SD Card*. Outro ponto para esta escolha de envio, foi que é possível ser realizado de forma gratuita, sem a necessidade de pagar mensalidades para acesso a plataformas de análise de dados. No entanto, a escolha desta forma de envio, apresenta uma baixa taxa de transferência das informações, o que dependendo do tamanho do arquivo salvo, pode impactar no tempo de envio.

Para que o protótipo desenvolvido possa ser elevado a condição de produto final, são necessárias alterações como: Definir componentes eletrônicos classificados como automotivos, em especial o microcontrolador. Para o uso da tecnologia de transmissão *Wi-Fi*, é necessário submeter o produto à análise de órgãos como ANATEL (Agência Nacional de Telecomunicações), para a certificação do uso de tecnologias de rádio frequência. O algoritmo deve ser genérico, afim de atender a maioria dos veículos dentro do mercado a quais irá atender.

Este trabalho permitiu contato com diversas áreas da Engenharia Eletrônica, onde é possível destacar o estudo dos protocolos de comunicação veiculares, o projeto das interfaces de *hardware* para implementação da camada física e também, sistemas de geolocalização, como sendo tecnologias amplamente utilizadas em produtos embarcados atualmente. O desenvolvimento do *firmware* para a escrita em arquivo utilizando protocolos de comunicação entre dispositivos, o uso de protocolos de rede como o HTTP e também, o contato com a linguagem de programação *JavaScript* para o desenvolvimento da página *web* de interface com a planilha, foram importantes para aplicar conceitos de manipulação de arquivos. A pesquisa sobre métodos de análise de dados, utilizando o Google *Data Studio*, trouxe uma visão da importância desta área no contexto da engenharia. O uso de ferramentas de desenvolvimento de *hardware* para projeto das placas de circuito impresso e o processo de logística envolvendo a fabricação de placas protótipo e componentes, trouxeram uma visão geral deste processo importante no desenvolvimento de um produto.

Por fim, declara-se que o processo de desenvolvimento deste trabalho elucidou para diferentes oportunidades de continuidade e possíveis melhorias a serem aplicadas ao protótipo. Neste sentido, poderiam ser pontos a tratar, por exemplo:

- Fazer a inclusão dos dados de acelerômetro e magnetômetro para uma melhor avaliação do comportamento de direção do motorista. Essa implementação traria a possibilidade de identificar colisões, acelerações e freadas bruscas.
- Desenvolver uma placa de circuito impresso mais compacta e já preparada para ser inserida diretamente no conector OBDII do veículo, sem a necessidade de cabos. Com isso,

melhoraria a usabilidade, tornando mais atrativo como produto.

- Utilizar um microcontrolador mais robusto e compatível com a aplicação automotiva, podendo ser escolhido um modelo com o periférico de *Wi-Fi* integrado, simplificando o circuito e auxiliando na compactação do *hardware*.
- Expandir o uso para outros tipo de veículo como, ônibus, caminhões e máquinas agrícolas, utilizando os protocolos adequados. Isto traria flexibilidade nas possíveis aplicações.

REFERÊNCIAS

Andria, G. et al. Design and implementation of automotive data acquisition platform. In: IEEE INTERNATIONAL INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE (I2MTC) PROCEEDINGS, 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p. 272–277.

Andria, G. et al. An automotive diagnostics, fuel efficiency and emission monitoring system using can. In: INTERNATIONAL CONFERENCE ON BIG DATA, IOT AND DATA SCIENCE (BIG DATA), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p. 14–17.

BRUNETTI, F. **Motores de combustão interna**. 2. ed. Rio de Janeiro: Editora Edgard Blucher, 2018. 544 p. ISBN 978-85-212-1294-2.

CAPELLI, A. **Eletroeletrônica automotiva - injeção eletrônica, arquitetura do motor e sistemas embarcados**. 1. ed. Sao Paulo: Editora Saraiva, 2010. 369 p. ISBN:978-85-365-1845-9.

Carrorama. **Carrorama/planos**. Disponível em: <<https://carrorama.net/planos/>>. Acesso em: 06 mar. 2021.

CARVALHO, E. A. d. **Leituras cartográficas e interpretações estatísticas ii**. 1. ed. Natal RN: Editora Saraiva, 2009. 244 p. ISBN:978-85-7273-525-4.

Chardsutthi, P.; Achariyasombat, K.; Adsavakulchai, S. E-training for private bus preventive maintenance. In: INTERNATIONAL CONFERENCE ON EDUCATION AND MANAGEMENT TECHNOLOGY, 2010., 2010. **Anais...** [S.l.: s.n.], 2010. p. 524–527.

COSTA, P. **O que é obd ii? parte i**. Disponível em:<<http://carroetecnica.com.br/2018/03/19/o-que-e-obd-ii-parte-i>>. Acesso em: 05 abril 2021.

Crear Sistemas. **Monitoramento e telemetria avançada para frotas leves**. Disponível em: <<https://www.crearesistemas.com.br/frota-leve/>>. Acesso em: 06 mar. 2021.

Cristiane Crelier. **Número de pessoas que trabalham em veículos cresce 29,2 por cento, maior alta da série**. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/26424-numero-de-pessoas-que-trabalham-em-veiculos-cresce-29-maior-alta-da-serie>>. Acesso em: 22 fev. 2021.

CUNHA, H. R. d. C. **Sistema de diagnóstico obd2/eobd para viaturas automóveis**. 2011. Dissertação (Mestrado) — Universidade de Aveiro, Aveiro, Portugal, 2011.

DI LUCIA SANTOS, R. **Redes gsm, gprs, edge e umts**. Disponível em:<https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf20082/ricardo/2.html> .

DIAS, Anderson. **Funcionamento e detalhes da unidade de comando eletrônico do sistema de injeção eletrônica do motor de combustão interna automotivo**. Disponível

em:<<http://carrosinfoco.com.br/2012/07/injecao-eletronica-uceecu-unidade-de-comando-eletronico>. Acesso em: 31 mar. 2021.

DILLENBURG, M. **Alternativas de aplicação do serviço gprs da rede celular gsm em telemetria pela internet**. Disponível em:<<https://www.novus.com.br/artigosnoticias/arquivos/artigotelemetriagprs.pdf>.

ELETRONICS, E. **Can a serial bus system - not just for vehicles**. Disponível em:<<https://www.esd-electronics-usa.com/Controller-Area-Network-CAN-Introduction.html>.

Equipe Munic.io. **Your car, your computer: ecus and the controller area network**. Disponível em:<https://store.munic.io/documentations/get_started. Acesso em : 31mar.2021.

Equipe Unidas. **Conheça 7 desafios da gestão de frotas e como resolvê-los**. Disponível em:<<https://frotas.unidas.com.br/blog/desafios-gestao-de-frotas/>. Acesso em: 22 fev. 2021.

FOROUZAN BEHROUZ, A. e. F. M. **Redes de computadores**. 1. ed. Porto Alegre: Editora Afiliada, 2013. 887 p. ISBN 978-85-805-5169-3.

FRENZEL, L. E. **Fundamentos de comunicação eletrônica**. 3. ed. Porto Alegre: Editora Bookman, 2013. 244 p. ISBN:978-85-7273-525-4.

GEO, M. **Padrão de formato de dados nmea**. Disponível em:<<https://mundogeo.com/2004/01/01/gps-21-11>>.

Gowda, V. R. C.; Gopalakrishna, K. Real time vehicle fleet management and security system. In: IEEE RECENT ADVANCES IN INTELLIGENT COMPUTATIONAL SYSTEMS (RAICS), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p. 417–421.

GRISTEC, E. **Relatório setorial**. Disponível em:<<https://gristec.com.br/novo/relatorio-setorial/>>.

GUIMARAES, A. **Eletrônica embarcada automotiva**. Sao Paulo: Editora Saraiva, 2007.

Lin, X. et al. Driver evaluation and identification based on driving behavior data. In: INTERNATIONAL CONFERENCE ON INFORMATION SCIENCE AND CONTROL ENGINEERING (ICISCE), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 718–722.

MACHADO, Antônio. **O sistema obd (on-board-diagnostic)**. Disponível em:<http://ave.dee.isep.ipp.pt/mjf/act_ect/SIAUT/Trabalhos. Acesso em : 27mar.2021.

MICROCHIP. **A can physical layer discussion**. 2002. Disponível em:<<http://ww1.microchip.com/downloads/en/Appnotes/00228a.pdf>>. Acesso em: 24 abr. 2021.

MICROCHIP. **Stand-alone can controller with spi interface**. 2019. Disponível em:<<https://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf>>. Acesso em: 24 abr. 2021.

MORAES, A. F. D. **Redes de computadores**. 1. ed. São Paulo: Editora Saraiva, 2014. 137 p. ISBN 978-85-365-2204-3.

MOURAD, R. **Por que o mercado de monitoramento e gestão de frotas está crescendo?**

Disponível em: <<https://revistamundologistica.com.br/artigos/por-que-o-mercado-de-monitoramento-e-gestao-de-frotas-esta-crescendo>>.

Naik, P. et al. An automotive diagnostics, fuel efficiency and emission monitoring system using can. In: INTERNATIONAL CONFERENCE ON BIG DATA, IOT AND DATA SCIENCE (BIG DATA), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p. 14–17.

OLIVEIRA, J. Pereira de. **Análise de desempenho de tcp sobre gprs em um ambiente fim a fim**. 2004. Dissertação (Mestrado) — Universidade Federal de Pernambuco, Pernambuco, Brasil, 2004.

Pimple, P. Sniffing the automotive can bus for real-time data-logging and real time diagnostics display. In: INTERNATIONAL CONFERENCE ON SMART ELECTRIC DRIVES AND POWER SYSTEM (ICSEDPS), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 167–170.

REDAÇÃO AB. **Aluguel para motoristas de aplicativos volta a crescer em julho**.

Disponível em: <<https://http://www.automotivebusiness.com.br/noticia/31432/aluguel-para-motoristas-de-aplicativos-volta-a-crescer-em-julho>. Acesso em: 21 fev. 2021.

SAWANT, P. R.; MANE, Y. B. Design and development of on-board diagnostic (obd) device for cars. In: FOURTH INTERNATIONAL CONFERENCE ON COMPUTING COMMUNICATION CONTROL AND AUTOMATION (ICCUBEA), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 1–4.

Stoltzfus, Justin. **Obd non obd protocols**. Disponível em: <<http://https://www.techopedia.com/your-car-your-computer-ecus-and-the-controller-area-network/2/32218>. Acesso em: 31 mar. 2021.

SVERZUT, J. U. **Redes gsm, gprs, edge e umts: evolucao a caminho da quarta geracao (4g)**. 4. ed. São Paulo: Saraiva, 2005. 456 p. ISBN 978-85-365-2206-7.

TONG, K.; NORIZAN, M. N.; MOHAMAD, I. S. Smart rash driver system via internet of things (iot). **MATEC Web of Conferences**, [S.l.], v. 140, p. 01026, 12 2017.

U-BLOX. **u-blox 6 receiver description including protocol specification**. 2013. Disponível em: <<https://www.u-blox.com/sites/default/files/products/documents/u-blox6ReceiverDescrProtSpec>. Acesso em: 02 jul. 2021.

APÊNDICE A - FERRAMENTA OBDII AUTOMOTIVA AUTOOL X60

A Figura 31 demonstra a ferramenta OBDII utilizada para a validação dos dados lidos da ECU. A ferramenta consiste em um *scanner* contendo um *display* mostrando os valores em tempo real das variáveis disponíveis no barramento CAN do veículo a qual está conectada através da porta OBDII. Para selecionar qual variável será lida, existe um botão que alterna o PID solicitado ao veículo. Caso o PID não esteja disponível para aquele modelo, é exibido um sinal indicativo no *display*.

Figura 31 – Ferramenta OBDII Autoool X60.



Fonte: O Autor.

APÊNDICE B - SCRIPT PARA PUBLICAR NA PLANILHA GOOGLE SPREADSHEET

O código apresentado nas Figura 32 e Figura 33, diz respeito ao *script* da aplicação *Web* elaborada para servir de *link* entre o dispositivo ESP8266 e a planilha do Google *Spreadsheet*. Na variável *sheet id*, é atribuído a URL da planilha onde serão publicados os dados. O *script* é configurado para publicar após a última linha de dados escrita. Dentro do laço *for*, são lidos os valores recebidos através dos parâmetros da *string* recebida. Conforme o parâmetro, o valor é escrito em sua respectiva coluna.

Figura 32 – Script desenvolvido para página *web* responsável por receber os dados do ESP8266 parte 1.

```

1  function doGet(e) {
2      Logger.log( JSON.stringify(e) );
3      var result = 'Ok';
4      if (e.parameter == 'undefined') {
5          result = 'No Parameters';
6      }
7      else {
8          var sheet_id = '1jjeuF_Th9MIMnA_hZp16PFV6TL0LY598Gq0hK7QDmY'; // Spreadsheet ID
9          var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet();
10         var newRow = sheet.getLastRow() + 1;
11         var rowData = [];
12
13         for (var param in e.parameter) { //varre a string separando os valores
14             //nas colunas corretas
15             Logger.log('In for loop, param=' + param);
16             var value = stripQuotes(e.parameter[param]);
17             Logger.log(param + ':' + e.parameter[param]);
18             switch (param) {
19                 case 'data':
20                     rowData[0] = value; // Coluna A
21                     break;
22                 case 'hora':
23                     rowData[1] = value; // Coluna B
24                     break;
25                 case 'latitude':
26                     rowData[2] = value; // Coluna C
27                     result = 'Temperature Written on column C';
28                     break;
29                 case 'longitude':
30                     rowData[3] = value; // Coluna D
31                     break;
32                 case 'temperaturaMotor':
33                     rowData[4] = value; // Coluna E
34                     break;
35                 case 'RPM':
36                     rowData[5] = value; // Coluna F
37                     break;
38                 case 'velocidade':
39                     rowData[6] = value; // Coluna G
40                     break;
41                 default:

```

Fonte: O Autor

Figura 33 – Script desenvolvido para página *web* responsável por receber os dados do ESP8266 parte 2

```
42     result = "unsupported parameter";
43 }
44 }
45     Logger.log(JSON.stringify(rowData));
46     var newRange = sheet.getRange(newRow, 1, 1, 7); //rowData.length
47     newRange.setValues([rowData]);
48 }
49     return ContentService.createTextOutput(result);
50 }
51 function stripQuotes( value ) {
52     return value.replace(/^[\'"]|[\']$/g, "");
53 }
54
55
```

Fonte: O Autor.

APÊNDICE C - CONEXÃO DO PROTÓTIPO COM O VEÍCULO

Conforme observado na Figura 34, o protótipo foi conectado ao veículo utilizando a porta OBDII, localizada logo abaixo do volante. A localização do conector de diagnóstico pode variar conforme o modelo do veículo.

Figura 34 – Conexão do protótipo com a porta OBDII.



Fonte: O Autor.

ANEXO A - TABELA PID

A Tabela 11 demonstra os códigos PID disponíveis e suas respectivas descrições.

Tabela 11 – Códigos PID OBDII para requisições de *status*.

PID (hex)	PID (dec)	bytes	Description
0	0	4	PIDs supported [01 - 20]
1	1	4	Monitor status since DTCs cleared.
2	2	2	Freeze DTC
3	3	2	Fuel system status
4	4	1	Calculated engine load
5	5	1	Engine coolant temperature
6	6	1	Short term fuel trim—Bank 1
7	7	1	Long term fuel trim—Bank 1
8	8	1	Short term fuel trim—Bank 2
9	9	1	Long term fuel trim—Bank 2
0A	10	1	Fuel pressure (gauge pressure)
0B	11	1	Intake manifold absolute pressure
0C	12	2	Engine speed
0D	13	1	Vehicle speed
0E	14	1	Timing advance
0F	15	1	Intake air temperature
10	16	2	Mass air flow sensor (MAF) air flow rate
11	17	1	Throttle position
12	18	1	Commanded secondary air status
13	19	1	Oxygen sensors present (in 2 banks)
14	20	2	Oxygen Sensor 1
			A: Voltage B: Short term fuel trim
15	21	2	Oxygen Sensor 2
			A: Voltage B: Short term fuel trim
16	22	2	Oxygen Sensor 3
			A: Voltage B: Short term fuel trim
17	23	2	Oxygen Sensor 4
			A: Voltage B: Short term fuel trim
18	24	2	Oxygen Sensor 5
			A: Voltage B: Short term fuel trim

			Oxygen Sensor 6
19	25	2	A: Voltage
			B: Short term fuel trim
1A	26	2	Oxygen Sensor 7
			A: Voltage
			B: Short term fuel trim
1B	27	2	Oxygen Sensor 8
			A: Voltage
			B: Short term fuel trim
1C	28	1	OBD standards this vehicle conforms to
1D	29	1	Oxygen sensors present (in 4 banks)
1E	30	1	Auxiliary input status
1F	31	2	Run time since engine start
20	32	4	PIDs supported [21 - 40]
21	33	2	Distance traveled with malfunction indicator lamp (MIL) on
22	34	2	Fuel Rail Pressure (relative to manifold vacuum)
23	35	2	Fuel Rail Gauge Pressure (diesel, or gasoline direct injection)
24	36	4	Oxygen Sensor 1
			AB: Air-Fuel Equivalence Ratio (lambda,)
			CD: Voltage
25	37	4	Oxygen Sensor 2
			AB: Air-Fuel Equivalence Ratio (lambda,)
			CD: Voltage
26	38	4	Oxygen Sensor 3
			AB: Air-Fuel Equivalence Ratio (lambda,)
			CD: Voltage
27	39	4	Oxygen Sensor 4
			AB: Air-Fuel Equivalence Ratio (lambda,)
			CD: Voltage
28	40	4	Oxygen Sensor 5
			AB: Air-Fuel Equivalence Ratio (lambda,)
			CD: Voltage
29	41	4	Oxygen Sensor 6
			AB: Air-Fuel Equivalence Ratio (lambda,)
			CD: Voltage
2A	42	4	Oxygen Sensor 7
			AB: Air-Fuel Equivalence Ratio (lambda)
			CD: Voltage
2B	43	4	Oxygen Sensor 8
			AB: Air-Fuel Equivalence Ratio (lambda)
			CD: Voltage
2C	44	1	Commanded EGR
2D	45	1	EGR Error
2E	46	1	Commanded evaporative purge
2F	47	1	Fuel Tank Level Input

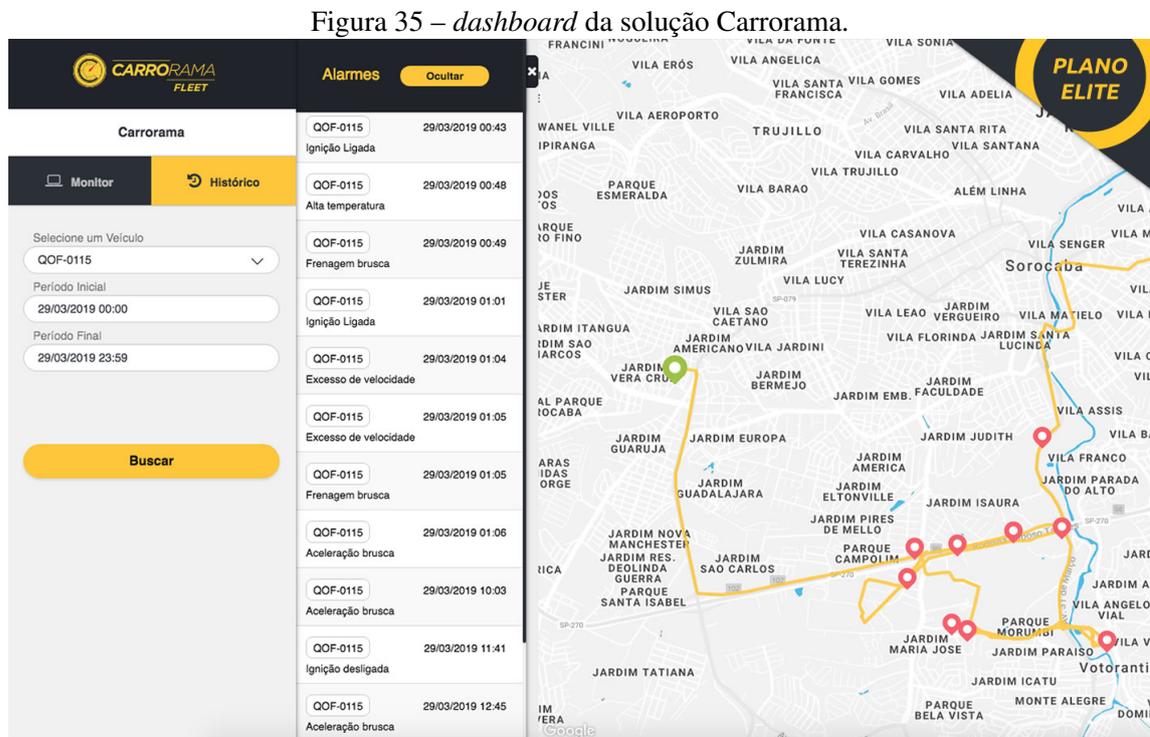
30	48	1	Warm-ups since codes cleared
31	49	2	Distance traveled since codes cleared
32	50	2	Evap. System Vapor Pressure
33	51	1	Absolute Barometric Pressure
34	52	4	Oxygen Sensor 1
			AB: Air-Fuel Equivalence Ratio (lambda)
			CD: Current
35	53	4	Oxygen Sensor 2
			AB: Air-Fuel Equivalence Ratio (lambda)
			CD: Current
36	54	4	Oxygen Sensor 3
			AB: Air-Fuel Equivalence Ratio (lambda)
			CD: Current
37	55	4	Oxygen Sensor 4
			AB: Air-Fuel Equivalence Ratio (lambda)
			CD: Current
38	56	4	Oxygen Sensor 5
			AB: Air-Fuel Equivalence Ratio (lambda)
			CD: Current
39	57	4	Oxygen Sensor 6
			AB: Air-Fuel Equivalence Ratio (lambda)
			CD: Current
3A	58	4	Oxygen Sensor 7
			AB: Air-Fuel Equivalence Ratio (lambda)
			CD: Current
3B	59	4	Oxygen Sensor 8
			AB: Air-Fuel Equivalence Ratio (lambda)
			CD: Current
3C	60	2	Catalyst Temperature: Bank 1, Sensor 1
3D	61	2	Catalyst Temperature: Bank 2, Sensor 1
3E	62	2	Catalyst Temperature: Bank 1, Sensor 2
3F	63	2	Catalyst Temperature: Bank 2, Sensor 2
40	64	4	PIDs supported [41 - 60]
41	65	4	Monitor status this drive cycle
42	66	2	Control module voltage
43	67	2	Absolute load value
44	68	2	Commanded Air-Fuel Equivalence Ratio (lambda)
45	69	1	Relative throttle position
46	70	1	Ambient air temperature
47	71	1	Absolute throttle position B
48	72	1	Absolute throttle position C
49	73	1	Accelerator pedal position D
4A	74	1	Accelerator pedal position E
4B	75	1	Accelerator pedal position F
4C	76	1	Commanded throttle actuator
4D	77	2	Time run with MIL on

4E	78	2	Time since trouble codes cleared
4F	79	4	Maximum value for Fuel–Air equivalence ratio,
50	80	4	Maximum value for air flow rate from MAF
51	81	1	Fuel Type
52	82	1	Ethanol fuel %
53	83	2	Absolute Evap system Vapor Pressure
54	84	2	Evap system vapor pressure
55	85	2	Short term secondary oxygen sensor trim, A: bank 1, B: bank 3
56	86	2	Long term secondary oxygen sensor trim, A: bank 1, B: bank 3
57	87	2	Short term secondary oxygen sensor trim, A: bank 2, B: bank 4
58	88	2	Long term secondary oxygen sensor trim, A: bank 2, B: bank 4
59	89	2	Fuel rail absolute pressure
5A	90	1	Relative accelerator pedal position
5B	91	1	Hybrid battery pack remaining life
5C	92	1	Engine oil temperature
5D	93	2	Fuel injection timing
5E	94	2	Engine fuel rate
5F	95	1	Emission requirements to which vehicle is de- signed

Fonte: O Autor.

ANEXO B - DASHBOARD DA FERRAMENTA DE GESTÃO VEICULAR CARRORAMA

A plataforma de gestão de frotas Carrorama foi utilizada como fonte de consulta e inspiração para o desenvolvimento do *dashboard* desenvolvido no presente trabalho. A plataforma disponibiliza de planos ao usuário, onde de acordo com o plano contratado, são disponibilizados diferentes recursos de visualização de dados. Os dados são coletados através de *scanner* conectado a porta OBDII dos veículos. Na Figura 35, são exibidas as informações disponíveis em um dos planos de serviços da empresa. Na imagem é possível visualizar as informações em no mapa dos pontos onde ocorreram os registros. No painel à esquerda, tem-se a descrição do registro com informações de data e hora.



Fonte: (Carrorama,2021).

