

FACULDADE DE TECNOLOGIA DE SANTO ANDRÉ
CURSO SUPERIOR DE GRADUAÇÃO TECNOLÓGICA DE ELETRÔNICA
AUTOMOTIVA

CARLOS HENRIQUE BEZERRA GONÇALVES
GUSTAVO OLIVEIRA ORTEGA
VITOR BENÍCIO SILVA SANTOS

SISTEMA ON BOARD DE MONITORAMENTO DE FROTAS

SANTO ANDRÉ
2018

CARLOS HENRIQUE BEZERRA GONÇALVES
GUSTAVO OLIVEIRA ORTEGA
VITOR BENÍCIO SANTOS SILVA

SISTEMA ON BOARD DE MONITORAMENTO DE FROTAS

Monografia apresentada ao Curso de tecnologia em Eletrônica Automotiva da FATEC Santo André, na disciplina de Trabalho de Graduação II, como requisito parcial para conclusão do curso de Tecnologia em Eletrônica Automotiva.

Orientador: Orlando de Salvo Junior

SANTO ANDRÉ
2018

LISTA DE PRESENÇA

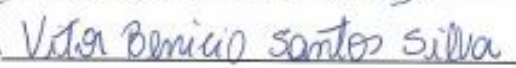
SANTO ANDRÉ, 06 DE DEZEMBRO DE 2018.

LISTA DE PRESENÇA REFERENTE À APRESENTAÇÃO DO
TRABALHO DE CONCLUSÃO DE CURSO COM O TEMA "Sistema
On Board de monitoramento de frotas" DOS ALUNOS DO 6º
SEMESTRE DESTA U.E.

BANCA

PRESIDENTE:
PROF. ORLANDO DE SALVO JUNIOR MEMBROS:
PROF. FERNANDO GARUP DALBO PROF. WAGNER MASSAROPE 

ALUNOS:

CARLOS HENRIQUE BEZERRA GONÇALVES GUSTAVO OLIVEIRA ORTEGA VITOR BENÍCIO SANTOS SILVA 

G635s

Gonçalves, Carlos Henrique Bezerra
Sistema On Board de monitoramento de frotas / Carlos Henrique
Bezerra Gonçalves, Gustavo Oliveira Ortega, Vitor Benício Santos
Silva. - Santo André, 2018. – 85f. il.

Trabalho de Conclusão de Curso – FATEC Santo André.
Curso de Tecnologia em Eletrônica Automotiva, 2018.

Orientador: Prof. Orlando de Salvo Junior

1. Eletrônica. 2. Sistema On Board. 3. Gerenciamento. 4. Frotas. 5.
Monitoramento. 6. Raspberry. 7. Tecnologia. 8. OBD. I. Ortega,
Gustavo Oliveira II. Silva, Vitor Benício Santos III. Sistema On Board
de monitoramento de frotas.

621.389

AGRADECIMENTOS

Gostaríamos de agradecer primeiramente a Deus por ter nos concedido força, determinação, foco, sabedoria e ter colocado em nossa caminhada até a finalização do trabalho pessoas que puderam esclarecer nossas dúvidas sempre com boa vontade de nos auxiliar da melhor maneira possível, entre elas estão o professor orientador Orlando Salvo Junior, que fez o acompanhamento do trabalho durante um ano sempre auxiliando e verificando alternativas que poderiam enriquecer e melhorar o projeto. Aos professores Murilo Zanini, Paulo Tetsu e Fernando Garup que nos ajudaram sempre quando precisávamos, o Flavson e o Álvaro sempre estando a disposição para o empréstimo de equipamentos e tendo a preocupação de nos deixarmos a vontade para trabalhar no TCC, nunca colocaram obstáculo nenhum. Ao professor Froes por ceder o espaço na praça técnica e o monitor aos sábados e à secretaria da Fatec no auxílio de fazer a liberação da entrada de alguns veículos para que pudéssemos realizar testes na praça técnica. Agradecemos todos aqueles que puderam nos ajudar de forma direta e indireta como familiares, amigos e professores.

RESUMO

O desenvolvimento da tecnologia veicular fez com que a mobilidade urbana mundial se tornasse o que ela é hoje, marcando presença nas mais diversas áreas de atuação de mercado. Pensando na massiva aplicação de veículos de transporte na prestação de serviços, foi desenvolvido um sistema para monitoramento de frotas, que consiste em um sistema que monitora diversos dados operacionais de um veículo integrante de uma frota em tempo real, obtendo informações que vão desde a localização espacial atual, até parâmetros como velocidade, rotação, temperatura do óleo, dentre outros, que possibilitam maior facilidade com o gerenciamento de manutenções e de acompanhamento em tempo em real dos veículos integrantes de frotas. A partir desses levantamentos de dados pode ser apontado os parâmetros a serem estudados para melhora a performance e reduzir os gastos, a partir de leitura de parâmetros disponíveis pela saída OBD e obtenção de localização via GPS. Por meio da aplicação de uma *Raspberry PI3*, o frotista tem em mãos, via internet, a localização atual dos veículos e a amostragem de diversos dados operacionais, além de também apresentar possíveis códigos de falha.

Palavras-chave: Frotas. OBD. Gerenciamento de Frotas.

ABSTRACT

The development of vehicular technology made what the worldwide urban mobility it is nowadays, being present in many actuation areas in the market. Thinking on the massive applications of transportations vehicles on services appliance, it was developed a system for fleets monitoring, that consists on a system which look into some operational data since actual localization to data's like speed, rotation, oil temperature, between others, that turns easier to handle maintenance management and real time monitoring of fleet vehicles. After this, a statistic data collection capable of pointing parameters can be done, to study better performance and less expenses, where, after the reading of available data on OBD output and obtainment of localization by GPS, together with application of a Raspberry PI3, the fleet manager have in hands, via internet, the actual localization of vehicles and sampling of several operational data, besides of presenting possible Diagnostic Trouble Codes.

Key-words: Fleets. OBD. Fleets management.

LISTA DE FIGURAS

Figura 1 - Evolução da Tecnologia nos Automóveis.....	19
Figura 2 - Conectores do tipo OBD I.....	22
Figura 3 - Conexão OBD II.....	22
Figura 4 - Esquemático da disposição dos satélites no sistema GPS	27
Figura 5 - Diagrama de Trilateração	27
Figura 6 - Multicaminhamento de Sinais GPS	28
Figura 7 - Situações distintas de PDOP.....	29
Figura 8 - Perturbação dos Sinais de Satélite pela Ionosfera e Troposfera	30
Figura 9 - Sistema Celular Básico	32
Figura 10 - Logo do Sistema Android	38
Figura 11 - Logo do Sistema IOS	39
Figura 12 - Logo do Sistema Bluetooth.....	40
Figura 13 - Programação no <i>App Inventor</i>	41
Figura 14 - Interface com o usuário no <i>App Inventor</i>	42
Figura 15 - Homepage do <i>Dweet.io</i>	43
Figura 16 - Exemplo de postagens no <i>Dweet.io</i>	43
Figura 17 - CI ELM327.....	44
Figura 18 - Diagrama de blocos do ELM327	45
Figura 19 - Conector ELM327.....	45
Figura 20 - Módulo GPS NEO 6M	46
Figura 21 - Logo da linguagem Python	48
Figura 22 - Exemplo de código acionamento de LED com PWM	49
Figura 23 - Exemplo de código em C.....	50
Figura 24 - Fluxograma do sistema SMOB.....	51
Figura 25 - Veículos de Teste: Audi A4 e Gol G5	53
Figura 26 - Veículo de Teste: Agile.....	53
Figura 27 - Conector Utilizado: ELM327 Bluetooth.....	55
Figura 28 - Fluxograma Multitarefa.....	57
Figura 29 - Placa do sistema de alimentação e bateria suplementar	59
Figura 30 - Página no <i>dweet.io</i> com todos os parâmetros lidos do OBD.....	63
Figura 31 - Logo SMOB.....	64

Figura 32 - Primeira <i>screen</i> do aplicativo e o ambiente de desenvolvimento	.65
Figura 33 - Segunda <i>screen</i> – digitação de usuário e senha.....	66
Figura 34 - Terceira <i>screen</i> – seleção de veículo	67
Figura 35 - Tela de acompanhamento operacional dos veículos da frota.....	68
Figura 36 - Função <i>Split</i> do <i>App Inventor</i>	69
Figura 37 - Tela de amostragem de localização dos veículos	70
Figura 38 - Projeto da caixa SMOB	71
Figura 39 - Projeto final da caixa SMOB.....	71

LISTA DE TABELAS

Tabela 1 - Principais definições de um sistema de comunicação sem fio.	33
Tabela 2 - Dados lidos pelo ELM327 e suas respectivas unidades	60
Tabela 3 - Dados lidos pela <i>Raspberry</i>	60
Tabela 4 - Dados lidos em cada veículo de teste	61

LISTA DE QUADROS

Quadro 1 - Dados obtidos pelo OBD I	21
Quadro 2 - Descrição dos pinos OBD.....	23
Quadro 3 - Funções Gerais do OBD II.....	24
Quadro 4 - <i>Hardware</i> do <i>Raspberry</i> PI 3	47
Quadro 5 - Estratégia para leitura de carga da bateria suplementar	56

LISTA DE ABREVIATURAS E SIGLAS

1G	Primeira Geração
2G	Segunda Geração
3G	Terceira Geração
3GPP	<i>3rd Generation Partnership Project</i>
3GPP2	<i>3rd Generation Partnership Project 2</i>
4G	Quarta Geração
ABS	<i>Anti Lock Brake System</i>
AMPS	<i>Advanced Mobile Phone System</i>
APP	Aplicativo
BATSU	Batéria Suplementar
CDMA	<i>Code Division Multiple Access</i>
DDC	<i>Display Data Channel</i>
DoD	Departamento de Defesa Americano
DOP	<i>Dilution of Precision</i>
DTC	<i>Diagnostic Trouble Codes</i>
DUN	<i>Dial-Up Networking Profile</i>
EEPROM	<i>Electrically-Eraseable Programmable Read-Only Memory</i>
EGR	<i>Exhaust Gas Recirculation</i>
ERB	Estação Rádio Base
ESN	<i>Electronic Serial Number</i>

EUA	Estados Unidos da América
EV-DO	<i>Evolution Data Optimized</i>
EV-DV	<i>Evolution Data and Voice</i>
FDD	<i>Feature Driven Development</i>
FTP	<i>File Transfer Profile</i>
GHz	<i>Giga Hertz</i>
GM	<i>General Motors</i>
GNSS	<i>Global Navigation Satellite System</i>
GPS	<i>Global Positioning System</i>
GSM	<i>Global System for Mobile Communication</i>
HDMI	<i>High Definition Multimedia Interface</i>
HID	<i>Human Interface Device Profile</i>
HSP	<i>Headset Profile</i>
INDE	Infraestrutura Nacional de Dados Espaciais
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
ISO	<i>International Organization for Standardization</i>
Kbps	Quilobit por segundo
Km	Quilômetros
LED	<i>Light Emitting Diode</i>
LTE	<i>Long Term Evolution</i>
Mbps	Megabit por segundo
MHz	<i>Mega Hertz</i>

MIMO	<i>Multiple Inputs Multiple Outputs</i>
MIN	<i>Mobile Identification Number</i>
MIT	Instituto de Tecnologia de Massachusetts
MMS	<i>Multimedia Messaging Service</i>
MS	<i>Mobile Station</i>
MSC	<i>Mobile Switching Center</i>
MTSO	<i>Mobile Telephone Switching Office</i>
NMEA	<i>National Marine Electronic Association</i>
OBD	<i>On-Board Diagnostics</i>
OFDMA	<i>Orthogonal Frequency Division Multiplexing/Multiple Access</i>
OHA	<i>Open Handset Alliance</i>
OPP	<i>Object Push Profile</i>
PC	<i>Personal Computer</i>
PDOP	<i>Positioning Dilution of Precision</i>
PID	<i>Parameter Identification</i>
PSTN	<i>Public Switched Telephone Network</i>
PWM	<i>Pulse Width Modulation</i>
RPM	Rotações por Minuto
RTCM	<i>Radio Technical Commission for Maritime Services</i>
SAE	<i>Society of Automotive Engineers</i>
SD	Secure Digital Card
SMOB	Sistema de Monitoramento <i>On Board</i>
SMS	<i>Short Message Service</i>

SPI	<i>Serial Peripheral Interface</i>
TDD	<i>Test Driven Development</i>
TDMA	<i>Time Division Multiple Access</i>
TM	Terminal Móvel
UART	<i>Universal Asynchronous Receiver Transmitter</i>
UMTS	<i>Universal Mobile Telecommunications System</i>
USB	<i>Universal Serial Bus</i>
WAN	<i>Wide Area Network</i>
WCDMA	<i>Wide-Band Code-Division Multiple Access</i>

SUMÁRIO

1 INTRODUÇÃO	19
2 REFERÊNCIA TEÓRICA	21
2.1 On-Board Diagnostic (OBD)	21
2.1.1 OBD I.....	21
2.1.2 OBD II.....	22
2.1.3 Protocolos de comunicação.....	24
2.1.4 Dados de diagnóstico	25
2.2 Rede GNSS: GPS	26
2.2.1 Método de Trilateração	27
2.2.2 Perturbações do sinal GPS.....	28
2.2.2.1 Obstrução	28
2.2.2.2 Refletido ou Multicaminhamento.....	28
2.2.2.3 DOP ou Diluição da Precisão	29
2.2.2.4 Atraso pela Ionosfera e pela Troposfera.....	30
2.3 Telefonia Móvel Celular.....	31
2.3.1 Sistemas de Rádio Móvel	31
2.3.2 Primeira Geração (AMPS - 1G)	34
2.3.3 Segunda Geração (GSM – 2G)	34
2.3.4 Segunda Geração (GPRS – 2,5G)	35
2.3.5 Segunda Geração (EDGE – 2,75G)	36
2.3.6 Terceira Geração (UMTS – 3G).....	36
2.3.7 Quarta Geração (LTE – 4G)	37
2.4 Sistemas Operacionais Móveis	37
2.4.1 <i>Android</i>	37
2.4.2 IOS	39

2.5 Ferramentas Computacionais	39
2.5.1 <i>Bluetooth</i>	40
2.5.2 <i>App Inventor</i>	41
2.5.3 <i>Dweet.io</i>	42
2.7 ELM 327.....	44
2.8 Módulo GPS NEO 6M	46
2.9 Raspberry PI 3	47
2.9.1 Linguagem de programação para <i>Raspberry</i>	48
2.9.2 Python	48
2.9.3 Linguagem C	49
3 METODOLOGIA.....	51
3.1 O projeto	51
4 PRÁTICA E RESULTADOS.....	54
4.1 Raspberry PI 3 e ELM327	54
4.2 Alimentação do sistema e bateria suplementar.....	57
4.3 Dados lidos e enviados pelo sistema	59
4.4 Envio de dados para a plataforma <i>Dweet.io</i>	62
4.5 Criação do aplicativo com o App Inventor	64
4.6 Caixa SMOB.....	70
5 CONCLUSÃO.....	72
5.1 Propostas Futuras	73
6 REFERÊNCIAS BIBLIOGRÁFICAS	74
APÊNDICE A – FLUXOGRAMA DO SOFTWARE DA RASPBERRY.....	81
APÊNDICE B – CIRCUITO BATERIA SUPLEMENTAR	82
APÊNDICE C – CIRCUITO MEDIDOR DE BATERIA.....	83
APÊNDICE D – CIRCUITO COMPLETO DE ALIMENTAÇÃO DO SISTEMA	84

ANEXO A – CIRCUITO DO CONECTOR ELM327	85
--	-----------

1 INTRODUÇÃO

A tecnologia no mercado automotivo foi surgindo e crescendo conforme o desenvolvimento dos sistemas eletroeletrônicos no decorrer da história, juntamente com a necessidade de seus usuários, vindo desde a implementação inicial nos primeiros automóveis em sistemas de iluminação até a implementação dos inúmeros sistemas micro controlados internos, controlando das mais simples até as mais complexas atividades envolvidas no veículo.

Estima-se que em 2000 a parcela da eletrônica embarcada no valor corrente dos veículos automotores era de aproximadamente 10% (MCALINDEN et al., 2000). De forma geral, hoje o automóvel pode ser descrito como uma plataforma que possui tecnologias de ponta e a indústria automotiva como uma das produtoras dessas tecnologias (MCALINDEN et al., 2000).

Um breve comparativo relacionado a evolução tecnológica dos automóveis pode ser conferida na Figura 1:

Figura 1 - Evolução da Tecnologia nos Automóveis



Fonte: Motorspot (2017)

A partir disso os automóveis pararam de ser considerados apenas um simples bem, e começaram a ser uma parte essencial na sociedade, seja para o simples ato de ir e vir, em transporte de cargas e de pessoas, ferramenta de trabalho, lazer ou simplesmente ostentação.

Vendo que os automóveis poderiam ser uma excelente ferramenta para prestação de serviços, como transporte de pessoas: taxi, UBER, 99Taxis, transporte público e transporte de cargas: matérias-primas, produtos para o mercado, empreendedores dos mais diversos tipos começaram a implementar as primeiras frotas na sociedade.

E desde então os números foram crescendo e crescendo, criando um extenso mercado de gerenciamento de veículos, visto que o transporte rodoviário nos dias atuais gira por volta de 7,5% do nosso Produto Interno Bruto (2,25 trilhão de dólares em 2012), ou seja, aproximadamente 170 bilhões de dólares por ano (VALENTE, NOVAES, et al., 2017).

A partir disso frotistas e afins são requeridos para realizar o direcionamento dos veículos para execução dos serviços, realização de manutenções preventivas, preditivas e corretivas, buscar formas para gastar o mínimo possível e realizar as atividades no menor tempo, além de diversas outras atribuições de um responsável por gestão de frotas.

Esclarecendo o sentido disso, temos que “O termo gestão de frotas representa a atividade de reger, administrar ou gerenciar um conjunto de veículos pertencentes a uma mesma empresa” (VALENTE, NOVAES, et al., 2017, p.1).

Quem está à frente de uma frota deve considerar todos os aspectos citados para buscar um gerenciamento eficiente, pois, quando se pratica um gerenciamento adequado da frota, a lucratividade é garantida (VALENTE, NOVAES, et al., 2017).

Em função do tempo e dinheiro consumido por falta de informação operacional precisa e no tempo certo, o objetivo deste estudo é desenvolver um sistema prático e de baixo custo que mostra em tempo real as informações embarcadas. Estes dados são importantes para o gestor de frota, visando otimizar todo o processo envolvido.

2 REFERÊNCIA TEÓRICA

2.1 On-Board Diagnostic (OBD)

Nesse capítulo serão abordados os principais aspectos relacionados a OBD.

2.1.1 OBD I

Está acontecendo um grande aumento da tecnologia embarcada nos veículos, ao decorrer do tempo, sendo assim é possível identificar se há avarias no sistema, verificar o funcionamento. Tudo isso pode ser obtido por meio da conexão OBD “*On-board Diagnostic*” que reúne todas as informações do sistema de gerenciamento do motor, como (rotação, ignição, tempo de injeção, falhas do sistema, ABS, *Airbag* e etc.) denominando um sistema de autodiagnostico, com esse sistema é possível realizar o monitoramento dos dados de emissões.

Os primeiros sistemas OBD I começaram a ser instalados em 1988 com a função de atender algumas legislações e efetuar monitorias no sistema como mostra o Quadro 1, no geral o que era obtido pelo OBD I, com algumas limitações pois não tinha uma padronização entre os fabricantes (Figura 2), havendo a falta de informações dos dados de cada sistema.

Quadro 1 - Dados obtidos pelo OBD I

OBD-I
Sensor de Oxigênio
Sistema de EGR
Sistema de combustível
Componentes elétricos
Sistemas Electrónicos
Informação de Diagnóstico
Códigos de erros

Fonte: Machado e Oliveira (2007)

Figura 2 - Conectores do tipo OBD I



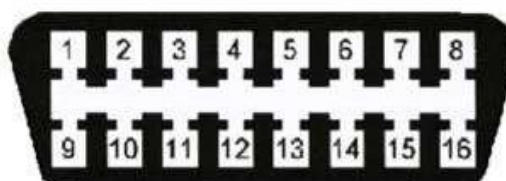
Fonte: Mercado Livre (2018)

2.1.2 OBD II

O OBD II é a evolução do OBD I com melhor padronização e capacidade, e essa conexão começou a ser obrigatória na Europa em 2000 e nos Estados Unidos a partir de 1994 e no Brasil a partir de 2010. São especificados o tipo de conector utilizado, pinos, sinais elétricos e formato de comunicação. Também é determinada a lista de parâmetros do veículo a serem monitorados e os dados que devem ser armazenados (SAE, 2002).

A padronização do conector de diagnóstico deve conter 16 pinos (2x8) conforme a Figura 3, de acordo com a norma SAE J1962 ou ISO 15031-3, tendo que estar localizado até 0,61 m do volante, ao alcance do motorista (SAE, 2002).

Figura 3 - Conexão OBD II



Fonte: OBD2-ELM327 [2018]

O Quadro 2 indica a função de cada pino para um conector padrão.

Quadro 2 - Descrição dos pinos OBD

Pinos	Função
1	Reservado ao fabricante
2	Sinal positivo do SAE-J1850 PWM e VPW
3	Reservado ao fabricante
4	Terra da Carroceria
5	Terra do Sinal
6	Rede Can-High (ISO 15765-4 e SAE J2234)
7	Rede K line da ISO 9141-2 e ISO 14230-4
8	Reservado ao fabricante
9	Reservado ao fabricante
10	Sinal negativo do SAE J1850 PWM
11	Reservado ao fabricante
12	Reservado ao fabricante
13	Reservado ao fabricante
14	Rede Can-Low (ISO 15765-4 e SAE J2234)
15	Rede L Line da ISO 9141-2 e ISO 14230-4
16	Voltagem de Bateria

Fonte: SAE International (2002)

O padrão OBD II estabelece uma lista padronizada de DTC's (*Diagnostic Trouble Codes*) que são códigos gerados pelo mau funcionamento de algum componente do veículo, ficando armazenado nas memórias de falhas do módulo eletrônico sendo possível identificar o que causou a falha por meio de uma única ferramenta genérica de *scanner* que poderá ler os dados que o sistema fornece como pode ser visto no Quadro 3 (MCCORD,2011).

Quadro 3 - Funções Gerais do OBD II

OBD II
Sensor de Oxigênio
Sistema de EGR
Sistema de combustível
Componentes elétricos
Sistemas Electrónicos
Informação de Diagnóstico
Códigos de falhas
Eficiência do catalisador
Aquecimento do catalisador
Combustão espontânea
Sistema de Evaporação
Sistema de ar secundário
Parâmetros do motor
Memorização de avarias
Estandardização das ligações

Fonte: Machado e Oliveira (2007)

2.1.3 Protocolos de comunicação

Nos dias atuais são utilizados cinco protocolos de sinais possíveis para serem utilizados em sistema OBD-II, tendo em vista que 4 modelos não são mais utilizados em veículos a partir de 2008, passando a utilizar somente o protocolo ISO 15765.

- SAE J1850 PWM: Utiliza-se dois fios para transmitir um sinal PWM possuindo uma taxa de transferência de 41.6 kbps com uma mensagem de 12 bytes esse sistema é um padrão utilizado pela Ford Motor (BASTOS, 2012; apud OLIVER,2012).
- SAE J1850 VPW: Um único fio que trabalha com pulsos variável VPW sendo considerado de baixo custo, tendo uma taxa de transferência de 10.4 kbps contendo o mesmo tamanho de mensagem do SAE J1850 PWM, utilizado pela *General Motors* (GM) e pela *Chrysler* (BASTOS, 2012; apud OLIVER,2012).
- ISO 9141-2: Tem a comunicação de forma serial assíncrona UART, com taxa de transmissão de 10.4 kbps, mensagens de 5 a 11 bytes esse

protocolo e utilizado pela *Chrysler*, por fabricantes europeus e asiáticos (BASTOS, 2012; apud ISO, 1994).

- ISO 14230: É apenas um link de diagnóstico e não pode ser usado para transmitir mensagens entre controladores (BASTOS, 2012; apud ISO, 1994).
- ISO 15765: Foi desenvolvido pela Bosch, tendo uma velocidade de transmissão de 500 kbps sendo o protocolo mais utilizado (BASTOS, 2012; apud ISO, 2004).

2.1.4 Dados de diagnóstico

As normas SAE J1979 e ISO 15031-5 definem como será feita a requisição de dados entre o veículo e o equipamento *scanner*, com o intuito de obter parâmetros do veículo como rotação, velocidade do veículo, temperatura do óleo, entre outros. Recebendo o nome de *Parameter Identification (PID)*, sendo alguns obrigatórios.

Existem, atualmente 10 tipos de serviço disponível no OBD-II como pode ser visto no Quadro 3.

Quadro 3 - Tipos de serviços do OBD-II

Serviço	Descrição	Informações disponíveis
\$01	Permite acesso aos dados <i>powertrain</i>	Temperatura do fluido de arrefecimento
		Velocidade do veículo
		Posição da borboleta do acelerador
\$02	Armazena as informações do veículo no momento que	Indica o que causou o <i>Freeze Frame</i>
\$03	Permite a ferramenta <i>scanner</i> listar todas as avarias de 5	Códigos de falha que impactam a emissões de poluentes
\$04	Utilizado para a ferramenta <i>scanner</i> limpar as avarias	-
\$05	Requisita resultados de teste do sensor de oxigênio.	-
\$06	Permite o acesso a resultados do monitoramento de componentes e sistemas específico	Falha de combustão
		Catalisador
\$07	Problemas relacionados a emissões durante o último ciclo de condução ou atual	-
\$08	Requisita o controle do sistema de bordo, teste ou componente	Permite que faça testes e opere componentes
\$09	Requisita informações do veículo	Informação de Calibração
\$0A	DTCs que impactam emissões de poluentes com status	-

Fonte: SAE (2007)

2.2 Rede GNSS: GPS

Uma das principais necessidades básicas do homem desde cedo foi saber onde ele está, em que parte do mundo ele se encontra (MARINO, 2018).

Partindo deste princípio, uma das primeiras ferramentas inventadas pelo homem foi a bússola, que não dava uma posição exata, porém auxiliava muito para encontrar a direção a se seguir.

Com o aumento da tecnologia em todo o globo terrestre, surgiu uma importante ferramenta de localização, muito popular nos dias de hoje, o GPS (*Global Positioning System*), ou simplesmente Sistema de Posicionamento Global.

O GPS é um dos sistemas vigentes de GNSS (*Global Navigation Satellites System*), ou Sistema Global de Navegação por Satélite, juntamente com o sistema GLONASS, que é o GNSS russo.

O GNSS é definido como uma constelação de satélites que permite determinar o posicionamento de um receptor em qualquer lugar do planeta, seja em terra, mar ou ar (SIRIUS, 2011).

Esses sistemas de posicionamento global surgiram como sistemas militares concorrentes, sendo estes implementados durante a guerra fria, onde o GPS foi implementado pelo Departamento de Defesa Americano (DoD) e o GLONASS pela antiga união soviética (FRANCO; DAMASCENO; PEREIRA; 2002).

O GPS se utiliza de uma tecnologia via satélite que permite determinar a atual posição na Terra, em termos de latitude, longitude e altitude (MARINO, 2018).

Segundo Franco, Damasceno e Pereira (2002), a constelação definitiva do GPS é composta por 24 satélites, sendo 21 deles operacionais e 3 reservas ativos, distribuídos em 6 planos orbitais. Cada um desses planos orbitais possuem 4 satélites a uma altura de 20200 km e uma inclinação de 55 graus em relação ao equador.

Marino (2018) afirma que essa distribuição garante que qualquer ponto da superfície da Terra esteja “visível” por pelo menos 4 satélites.

Um simples esquemático referente a disposição desses satélites pode ser visto na Figura 4.

Figura 4 - Esquemático da disposição dos satélites no sistema GPS



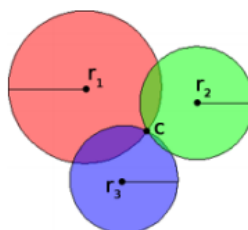
Fonte: Seos Project [2018].

2.2.1 Método de Trilateração

O método de Trilateração é o meio pelo qual a posição de um receptor GPS é obtida, tendo esse método como referência o sinal emitido pelos satélites.

Segundo Furlan (2015, p.14), “O método da trilateração baseia-se na medição de distâncias, usa-se a propriedade matemática da trigonometria, onde com a presença de três círculos é possível identificar um ponto único, na interseção destes”, como pode ser observado na Figura 5.

Figura 5 - Diagrama de Trilateração



Fonte: Furlan (2015)

2.2.2 Perturbações do sinal GPS

2.2.2.1 Obstrução

Partindo do princípio de que todo sistema possui modos de falha (saída não ideal) e fatores de ruído (fatores não controláveis), nesses capítulos serão abordadas as possíveis perturbações possíveis do sinal GPS.

Uma das possibilidades de perturbação dos sinais GPS está na obstrução do sinal, talvez a mais simples de ser compreendida.

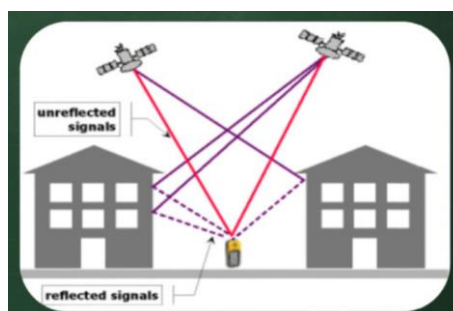
Obstrução do sinal GPS é quando tem-se um ou mais satélites fora do “campo de visão” do receptor GPS, diminuindo ou até não possibilitando uma medição precisa da posição atual, sendo que a solução ideal seria o receptor estar o mais alto possível e distante de quaisquer obstruções, para estar visível para o maior número possível de satélites (RIBAS, 2007).

Essa ocorrência de não estar nesse “campo de visão” normalmente ocorre quando o receptor GPS tem uma obstrução parcial, como prédios, ou total, quando se encontra no interior de um túnel, perdendo a capacidade de localização espacial exata.

2.2.2.2 Refletido ou Multicaminhamento

Dentre os sinais de satélite que chegam em um receptor, um ou mais deles podem não ser o sinal transmitido diretamente, sendo que esse pode ter sido desviado em algum objeto na superfície da Terra conforme visto na Figura 6 (MARINO, 2018).

Figura 6 - Multicaminhamento de Sinais GPS



Fonte: GIS Resources [2018]

Esse sinal refletido pode ser facilmente desconsiderado, visto que possui intensidade consideravelmente menor que o sinal original, além disso, existem antenas que possuem tecnologias para minimizar os efeitos do multicaminhamento (MARINO, 2018).

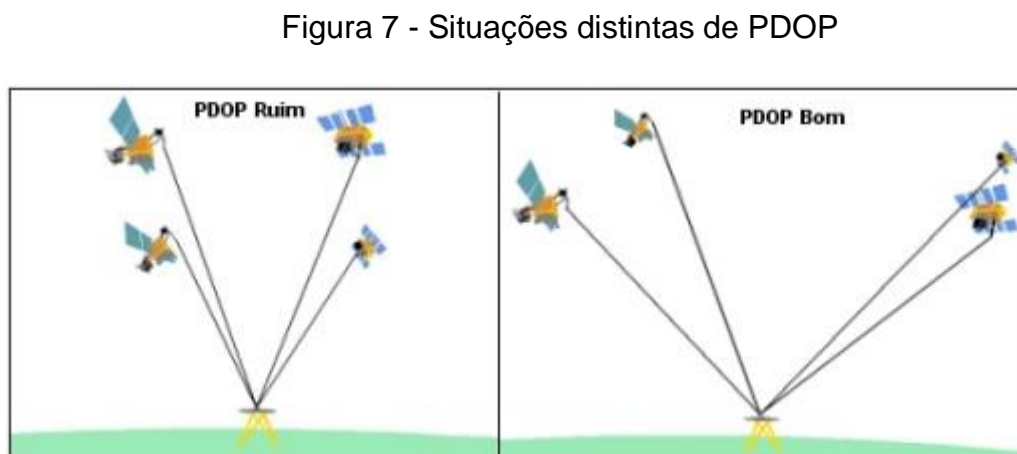
2.2.2.3 DOP ou Diluição da Precisão

A qualidade do levantamento da atual posição de um receptor GPS também está relacionada com a posição geográfica dos satélites na hora do rastreamento, sendo o DOP (*Dilution of Precision*), ou simplesmente Diluição da Precisão, um indicativo dessa geometria, conseqüentemente da qualidade dos dados a serem obtidos (MARINO, 2018).

Segundo a Infraestrutura Nacional de Dados Espaciais, existem diversos tipos de DOP, porém o mais significativo para o posicionamento por ponto é o PDOP (*Positioning Dilution of Precision*), que está diretamente relacionado com o posicionamento tridimensional (INDE, 2008).

Para fins de precisão, quanto menor o valor de PDOP, melhor a precisão esperada. Em termos práticos o valor de PDOP está diretamente relacionado com o inverso do poliedro formado entre os satélites e o receptor, onde volumes maiores proporcionam PDOP menores (INDE, 2008).

A Figura 7 apresenta duas situações distintas de PDOP:



Fonte: INDE (2008)

2.2.2.4 Atraso pela Ionosfera e pela Troposfera

No caminho entre satélite e receptor, o sinal emitido passa pela Ionosfera e pela Troposfera, de forma que essa passa a ter um determinado atraso, podendo influenciar no cálculo de levantamento da posição do receptor.

Em relação a Ionosfera, seu comportamento normal é alterado por fenômenos que ocorrem na superfície solar, normalmente explosões solares, provocando perturbações das camadas ionosféricas ionizando-as na região dos polos (MARINO, 2018).

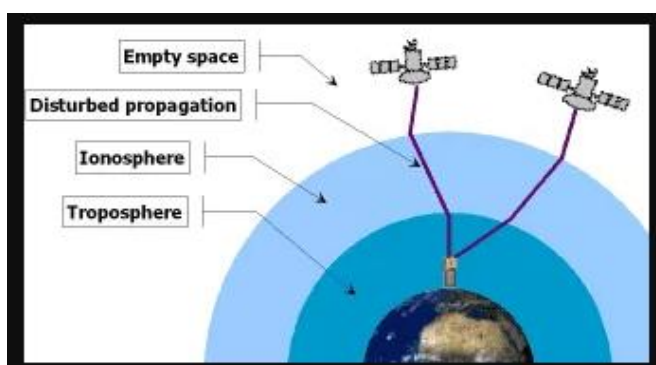
Durante o período em que a Terra está exposta a estas anomalias as características das diversas camadas são alteradas, influenciando diretamente nos sistemas de comunicação (MARINO,2018).

Entretanto, todas as variações que acontecem na Ionosfera são parcialmente previsíveis e dependem diretamente da atividade solar e do grau de ionização que as radiações solares provocam na ionosfera. Dessa forma é possível aplicar os conhecimentos locais para prever as condições de propagação dentro de certos limites (MARINO, 2018).

Em relação a Troposfera, camada grossa da atmosfera que possui em relação a superfície 50 km de altura, esta possui um atraso relativamente pequeno em comparação com a Ionosfera, atraso esse dependente de temperatura, umidade e pressão que variam conforme a altitude local (MARINO, 2018).

Na Figura 8 é possível ver a influência da Ionosfera e da Troposfera na transmissão do sinal dos satélites:

Figura 8 - Perturbação dos Sinais de Satélite pela Ionosfera e Troposfera



Fonte: GIS Resources [2018]

2.3 Telefonia Móvel Celular

Nesse capítulo do trabalho serão levantadas as principais características relacionadas ao funcionamento e composição das telefônicas móveis celulares, tendo como principal objetivo o esclarecimento de questões futuras da metodologia e prática.

Segundo Vaz (2015), Telefonia Móvel é uma rede de telecomunicação sem fio de grande cobertura (WAN – *Wide Area Network*) projetada para o provisionamento dos serviços de comunicação de dados.

A necessidade de comunicação móvel foi se tornando cada vez mais crítica na sociedade conforme a tecnologia foi se desenvolvendo, resultando no surgimento de inúmeras ferramentas de comunicação que permitiam de certa forma “encurtar” a distância entre as pessoas e instituições, trazendo e levando informações das mais diversas de um lado para o outro.

O que é conhecido hoje como celular teve em seus primórdios a simples capacidade de comunicação por voz entre um aparelho e outro, nos dias atuais essa característica é apenas uma das diversas ferramentas aplicadas nesses aparelhos, indo desde o envio de mensagens de texto, até a capacidade do uso de sistemas como GPS, canais de notícia, redes sociais, câmera fotográfica, dentre outros.

2.3.1 Sistemas de Rádio Móvel

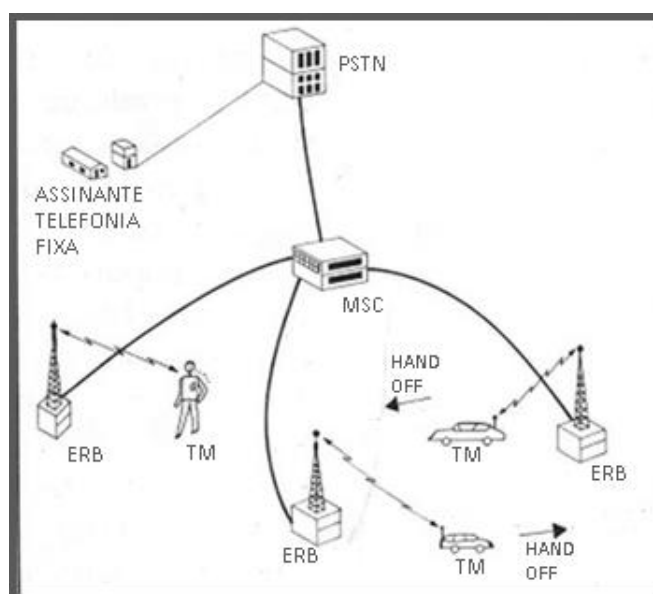
Até a telefonia chegar na tecnologia dos dias de hoje, segundo Rappaport (2009), muitos padrões de rádio móvel para sistemas sem fio foram desenvolvidos.

Em relação a um sistema de telefonia celular, este oferece uma conexão sem fio à PSTN (*Public Switched Telephone Network*) para usuários de qualquer local dentro do alcance de rádio do sistema, abrangendo um grande número de usuários dentro de um espectro de frequência limitado. A alta capacidade de um transmissor/receptor é alcançada de forma limitada a cobertura de cada transmissor de estação base e uma pequena área geográfica, sendo essa área geográfica chamada de célula (RAPPAPORT, 2009).

A Figura 9 mostra um sistema celular básico, que consiste em estações móveis ou TM (Terminal Móvel), estações base ou ERB e uma Central de Comutação Móvel,

ou simplesmente MSC (*Mobile Switching Center*), podendo ser chamada também de Escritório de Comutação de Telefonia Móvel [*Mobile Telephone Switching Office (MTSO)*], devido ao fato de ser responsável por conectar todas as estações móveis à PSTN (RAPPAPORT, 2009).

Figura 9 - Sistema Celular Básico



Fonte: Ebah (2016) (adaptado)

O *hand off* ocorre quando uma estação móvel transita de uma célula para outra, fazendo com que a comunicação passe a ser realizada pela estação base da outra célula.

Esse *hand off* pode ser realizado de duas formas, sendo essas: o *soft hand off* e o *hard hand off*.

O *soft hand off* ocorre quando a estação base da próxima célula começa a receber o sinal da estação móvel antes mesmo de se desconectar da estação base de origem, dessa forma, não existe perda de continuidade no sinal. O *hard hand off* ocorre quando a estação base de origem desconecta antes do sinal ser recebido por outra estação, dessa forma, se não houver a conexão entre a estação móvel e outra estação base, a ligação será finalizada (COSTA, 2005).

Existe também um tipo de *hand off* que é conhecido por *roaming*, onde a estação móvel transita de uma célula para outra que está conectada em outra MSC.

Para ajudar a entender os principais aspectos desse sistema, a Tabela 1 define o que é necessário para aprofundar o estudo em comunicação sem fio.

Tabela 1 - Principais definições de um sistema de comunicação sem fio.

Estação-base (BTS)	Uma estação fixa em um sistema de rádio móvel usada para comunicação por rádio com estações móveis. As estações-base estão localizadas no centro ou na borda de uma região de cobertura e consistem em canais de rádio e antenas transmissoras e receptoras montadas em uma torre.
Estação móvel (MS)	Uma estação no serviço de rádio-celular destinada para uso enquanto estiver em movimento em locais não especificados. Estações móveis podem ser unidades pessoais portáteis ou instaladas em veículos.
Canal de controle	Canal de rádio usado para transmissão de configuração, solicitação, início da chamada, e outras finalidades de orientação ou controle.
Canal direto	Canal de rádio usado para transmissão de informações da estação-base para a móvel.
Canal reverso	Canal de rádio usado para a transmissão de informações da estação-móvel para a estação-base.
Sistemas <i>simplex</i>	Sistemas de comunicação que oferecem apenas comunicação unidirecional.
Sistemas <i>duplex</i>	Sistemas de comunicação que permitem a comunicação bidirecional simultânea. Transmissão e recepção normalmente são feitas em 2 canais diferentes (FDD), embora os novos sistemas sem fio estejam usando TDD.
Sistemas <i>semiduplex</i>	Sistemas de comunicação que permitem a comunicação bidirecional usando o mesmo canal de rádio para transmissão e recepção. Em determinado momento, o usuário só pode transmitir ou receber informações.
Transferência	O processo de transferir uma estação móvel de um canal ou estação-base para outra.
Página	Uma pequena mensagem que é transmitida em broadcast para toda a área de serviço, normalmente em um padrão <i>simulcast</i> por muitas estações-base ao mesmo tempo.
Central de Comutação móvel (MSC)	Central de comutação que coordena o roteamento de chamadas em uma área grande de serviço. Em um sistema de rádio-celular, a MSC conecta as estações base de celular e os dispositivos móveis à PSTN.
Assinante	Um usuário que paga taxas de assinatura para usar um sistema de comunicação móvel.
Visitante	Uma estação-móvel que opera em uma área de serviço (mercado) diferente daquela onde o serviço que foi assinado opera.
Transceptor	Um dispositivo capaz de transmitir e receber sinais de rádio simultaneamente.

Fonte: SJ IFSC [2018] (adaptado)

2.3.2 Primeira Geração (AMPS - 1G)

Como principal marco da primeira geração tem-se o sistema analógico AMPS (*Advanced Mobile Phone System*) inventado pelo *Bell Labs*, instalado primeiramente nos EUA em 1982, onde sistemas mais avançados herdaram muitas características desse sistema. (COSTA, 2005)

Esse sistema não oferecia muitos serviços, sendo muito semelhante a telefonia fixa, exceto pelo fato do meio físico de comunicação. (VAZ, 2015)

A primeira Geração funcionava a partir de modulação em frequência (FM - *Frequency Modulation*), com múltiplo acesso por divisão de frequência, operando na faixa de 800MHz (TELECO, 2018).

Segundo Stadler (2013), haviam muitos problemas de segurança relacionados a essa tecnologia, pois não haviam sistemas de criptografia, dessa forma, qualquer pessoa poderia interceptar dados de comunicação e decodificá-los.

Outro problema presente nesse sistema era a possibilidade de captura do ESN (*Electronic Serial Number*) e do MIN (*Mobile Identification Number*) que autenticavam o usuário ao sistema. Se aproveitando disso, usuários mal-intencionados poderiam facilmente se utilizar desses dados, praticando clonagem de telefones (STADLER, 2013).

Um outro grande problema apresentado nesse sistema foi a falta de padronização, onde cada fabricante buscava sua forma de implantar a sua própria rede, gerando muitos inconvenientes relacionados a aparelhos incompatíveis com redes incompatíveis. Esse sistema alcançava velocidades de *download* sempre abaixo de 10kbps (ROMER, 2013).

2.3.3 Segunda Geração (GSM – 2G)

A segunda geração trouxe consigo a implementação do sistema GSM (*Global System for Mobile Communication*) ou sistema Global para Comunicação Móvel, que iniciou suas atividades em 1991 (ROMER, 2013).

Este protocolo foi o responsável por eliminar os problemas relacionados a falta de padronização da implementação das redes de telefonia móvel, tendo como uma das principais vantagens o uso de criptografia digital, eliminando conversas por meios

analógicos, fato que colaborou muito para a expansão do uso de celulares, além da possibilidade de compatibilidade de qualquer aparelho em redes pelo mundo inteiro (ROMER, 2013).

Esse sistema funciona a partir de acesso múltiplo por divisão de tempo (TDMA) e acesso múltiplo por divisão de código (CDMA-One), operando entre 850 a 1900MHz. O TDMA, segundo Teleco (2018), “propicia a transmissão de voz, com uma divisão do canal de frequência em seis intervalos distintos de tempo”, já o CDMA, segundo Teleco (2018), “permite que todos os usuários assinantes transmitam e recebam informações por um mesmo canal, simultaneamente”.

Além disso tudo a geração trouxe consigo uma melhora considerável na qualidade dos serviços prestados, tais como: melhor qualidade de voz, identificador de chamadas, serviços de mensagens curtas (SMS), serviços de mensagens multimídia (MMS), dentre outros (TELECO, 2018).

Nos primeiros testes com essa rede, taxas de transferência de 97 kbps puderam ser utilizadas.

2.3.4 Segunda Geração (GPRS – 2,5G)

Considerada como rede “2,5G” o *General Packet Radio Service*, ou GPRS (Serviço de Rádio de Pacote Geral) trouxe, segundo Romer (2013), “uma melhora significativa para as transmissões móveis, aumentando as taxas de transferência de dados em redes GSM”.

O GPRS é um padrão de rádio por pacote, que realiza a comunicação através de um sistema sem a necessidade de conexão direta, onde o uso é tarifado por utilização e não mais por tempo de uso (TELECO, 2018).

Tendo entre 32 kbps a 80 kbps ao usuário final, novas funcionalidades foram incluídas, como a utilização simultânea de dados, voz e o acesso imediato e permanente à rede de dados, possibilidades não encontradas no sistema GSM (ROMER, 2013).

2.3.5 Segunda Geração (EDGE – 2,75G)

Vindo como mais uma evolução das redes móveis, o EDGE (*Enhanced Data Rates For GSM*), ou simplesmente Taxas de Dados Ampliadas para a Evolução do GSM, trouxe uma capacidade de banda de até 236 kbps (ROMER, 2013).

Essa tecnologia também propiciou o desenvolvimento da telefonia celular rural, devido a excelente qualidade de propagação do sinal (TELECO, 2018).

Essa rede funciona basicamente como uma “área reserva”, quando a rede 3G das operadoras está em tráfego intenso.

2.3.6 Terceira Geração (UMTS – 3G)

O 3G com seu sistema UMTS (*Universal Mobile Telecommunications System*), ou Sistema Móvel de Telecomunicações Universal, trouxe uma maneira mais eficiente de navegar na internet com *smartphones*, auxiliando e muito nas tarefas do dia a dia (ROMER, 2013).

A Terceira Geração, que passou a operar entre 1900 e 2100MHz, inclui novas tecnologias como o WCDMA, que é a banda larga CDMA, sendo um aperfeiçoamento do sistema GSM. Este sistema é fundamentado a partir do Protocolo de Internet (IP), onde o acesso a essa tecnologia é feita a partir de códigos, com elevadas taxas de transmissão por volta de 2 Mbps (TELECO, 2018).

Outra tecnologia implementada nesse sistema é o CDMA 2000 1XEV-DV, para pacote HSDPA, em outras palavras, acesso em pacote com enlace de descida em alta velocidade, que aumenta a capacidade do WCDMA. Tendo também a tecnologia CDMA 2000 1XEV-DO, que é um aperfeiçoamento de sua antecessora, que permitia a conectividade sem fio em altíssima velocidade, sendo comparável a banda larga com fio (TELECO, 2018).

2.3.7 Quarta Geração (LTE – 4G)

A quarta geração trouxe consigo o LTE (*Long Term Evolution*), que, segundo Teleco (2017), “é o termo adotado para designar o padrão de 4ª Geração estabelecido para a rede das operadoras de celular como evolução para operadoras de GSM”.

O LTE foi padronizado pelo 3GPP, e traz consigo novas técnicas de modulação, sendo elas: o sistema OFDMA (*Orthogonal Frequency Division Multiple Access*) e o MIMO (*Multiple Input Multiple Output*). Essas tecnologias também são utilizadas pelo WiMAX, padronizado pelo 3GPP2 (TELECO, 2017).

O sistema 4G trouxe consigo em 2012 no Brasil, uma frequência de operação de 2,5GHz com velocidades atingindo picos de até 120 Mbps (VAZ, 2015).

2.4 Sistemas Operacionais Móveis

Nos dias atuais o uso do celular se tornou indispensável e deixou de ser uma tecnologia de status para tornar-se uma ferramenta de trabalho, onde quase 3 bilhões de pessoas possuem um telefone celular, onde, com ele é possível de se obter comunicação rápida com várias pessoas do mundo todo, pedir um táxi, utilizar o GPS para se localizar, dentre outros, tudo isso é possível por meio de aplicativos desenvolvidos para auxiliar o cotidiano dos usuários.

Os sistemas operacionais mais comuns que podem ser encontrados em *smartphones* e *tablets* são, o *Android* e o *IOS*.

2.4.1 *Android*

O sistema *Android* consiste em uma plataforma de desenvolvimento para aplicativos móveis gratuita onde várias empresas líderes de mercado de telefonia como a Motorola, LG, Samsung, dentre outras se utilizam dessa plataforma (LEVHETA, 2010).

Essas empresas formam um grupo chamado de “*Open Handset Alliance*” OHA foi criado com a intenção de padronizar uma plataforma de código aberto livre para celulares e atender as necessidades do usuário.

O *Android* (Figura 10) é baseado em um sistema operacional *Linux* onde ele mesmo se encarrega de gerenciar a memória e os processos, permitindo que diversas aplicações possam ser executadas ao mesmo tempo, tendo um ambiente de desenvolvimento bastante poderoso, ousado e flexível, sendo possível programar aplicativos pela linguagem *Java*, uma das mais conhecidas pelos programadores, havendo também outros meios de programação (LEVHETA, 2010).

Figura 10 - Logo do Sistema Android



Fonte: Escola Android (2015)

Essa plataforma tem interação com o *SQLite* (um mecanismo para o banco de dados), um leve e poderoso banco de dados onde permite que uma aplicação possa utilizar normalmente essa função, podendo haver um ou mais banco de dados.

Outra funcionalidade muito útil do *Android* é a interação com o *Google Maps*, dando a possibilidade de desenvolver aplicativos de localização com o GPS (*Global Position System*), de uma forma muito simples e com poucas linhas de código.

O *Android* possui diversas funcionalidades possíveis, tornando ele uma das ferramentas mais usadas no mundo, tendo um crescimento muito grande no decorrer dos anos por ser *open source* (LEVHETA, 2010).

2.4.2 IOS

O *IOS* (Figura 11) é um sistema operacional desenvolvido originalmente para a linha de dispositivos móveis da *Apple*. Sendo uma plataforma única de *hardware* e *software* projetados para funcionar em conjunto. Por conta disso, os aplicativos conseguem aproveitar totalmente os recursos do *hardware*.

Figura 11 - Logo do Sistema IOS



Fonte: CBR (2018)

O desenvolvimento de aplicativos para *IOS* é baseado em uma linguagem de programação chamada *Objective-C*, sendo uma linguagem de programação orientada a objetos que adicionam o estilo de troca de mensagens de *Smalltalk* à linguagem de programação C (SIMÕES, PEREIRA, 2014).

Por ser um sistema utilizado apenas nos dispositivos da *Apple*, este contém uma maior performance na execução dos aplicativos, havendo funcionalidades bem parecidas com a do sistema *Android*.

2.5 Ferramentas Computacionais

Nesse capítulo serão explanadas as ferramentas computacionais aplicáveis ao projeto de graduação.

2.5.1 Bluetooth

O sistema *Bluetooth* (Figura 12) foi implementado para a comodidade de diminuir a quantidade de cabos para conexão entre dispositivos, considerando isso, foi desenvolvido um sistema com características de curto alcance e baixa potência, que se adequa as necessidades de sua aplicação (RAPPAPORT,2008).

Figura 12 - Logo do Sistema Bluetooth



Fonte: Bluetooth (2018)

O desenvolvimento do *Bluetooth* fez-se por um consórcio de empresas composto por *Ericsson, IBM, Nokia, Toshiba e Intel*. Publicada em julho de 1999. Se tornando um padrão, sendo incluído em diversos projetos para a transferência de dados com baixo alcance. (MORIMOTO,2008).

O *Bluetooth* possui uma velocidade de até 1 *megabit*, mas pelo protocolo de comunicação adotado, o sistema apresenta atenuações, nas velocidades reais de 721 kilobits em assíncrono ou 432 kilobits em modo síncrono. Essas velocidades podem sofrer mais alterações por perda de dados e reenvio de pacotes (MORIMOTO,2008).

Segundo Morimoto (2008) o protocolo de comunicação do *Bluetooth* é destinado à aplicação, portanto cada dispositivo que utiliza o *Bluetooth* possui características de protocolo diferentes. Os protocolos podem ser:

- HSP (*Headset Profile*): Utilizado por *Headsets Bluetooth*, permitindo o envio de *streaming* de áudio.
- HID (*Human Interface Device Profile*): Usados por teclados, mouses, *joysticks* e outros dispositivos de entrada.
- FTP (*File Transfer Profile*): Permite transferência de arquivos.

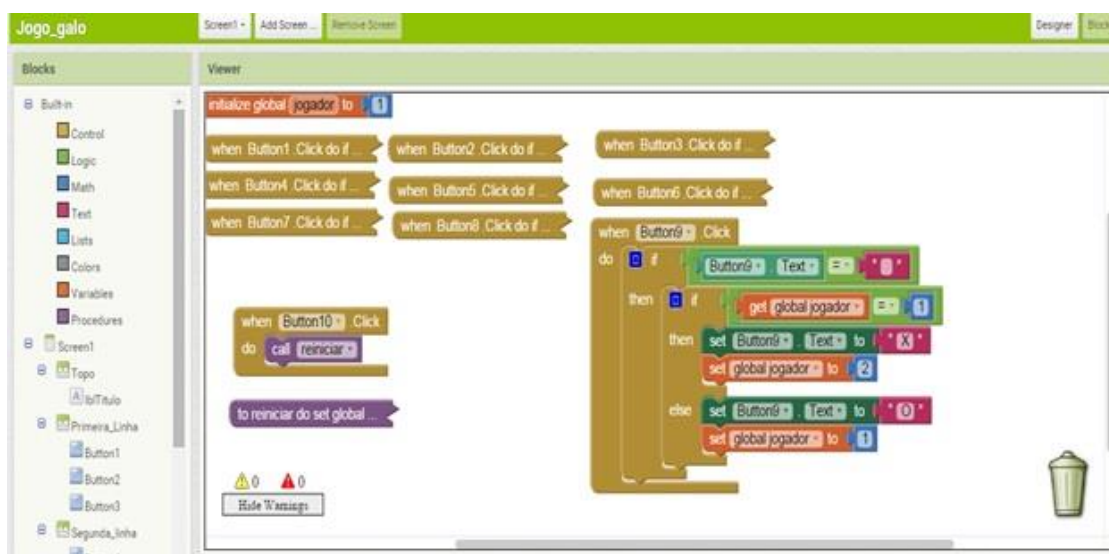
- OPP (*Object Push Profile*): Um protocolo de transferência de dados de uso geral, utilizado para transferir contatos, fotos, e outras informações.
- DUN (*Dial-up Networking Profile*): Usado por celulares para permitir o acesso à web através do computador.

2.5.2 App Inventor

É uma ferramenta desenvolvida pela *google* para a criação de aplicativos móveis, atualmente é mantida pelo Instituto de Tecnologia de Massachusetts (MIT), onde se permite que um usuário qualquer sem ter conhecimento de programação possa criar aplicativos para *smartphones*, *tablets* e outros que tenham um sistema operacional *Android*, sendo este um *software* gratuito.

A programação pode ser feita em navegadores como *Google Chrome*, *Mozilla Firefox*, *Safari*, dentre outros, sendo uma ferramenta de *drag-and-drop* (arrastar e soltar) visual podendo projetar a interface do usuário (aparência visual do aplicativo a ser criado) pelo construtor de interface gráfica podendo também especificar o comportamento que o aplicativo terá, reunindo as informações por blocos como se estivesse montando um quebra-cabeça como mostra a Figura 13, onde tem-se a demonstração do código para a execução de um jogo.

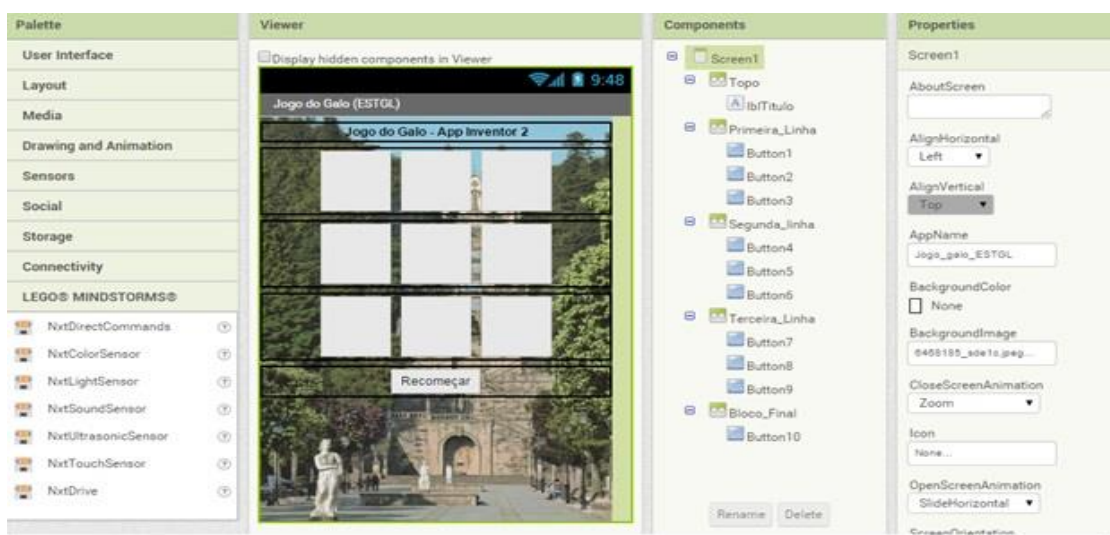
Figura 13 - Programação no App Inventor



Fonte: Clube Scratch (2015)

A Figura 14 mostra a interface com o usuário.

Figura 14 - Interface com o usuário no *App Inventor*

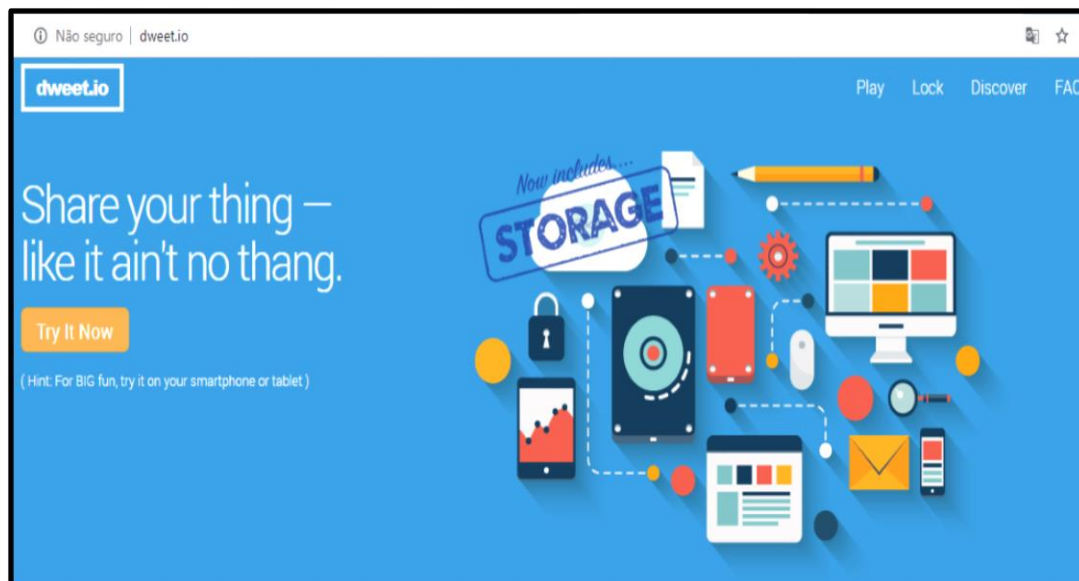


Fonte: Clube Scratch (2015)

O *App Inventor* permite a criação de vários tipos de aplicativos, tudo vai da imaginação, criatividade e da necessidade de quem está programando, pode ser para criar aplicativos educativos, de localização, jogos, que possam controlar robôs, dentre entre outros (SILVA, PAIVA, FORTES, 2017).

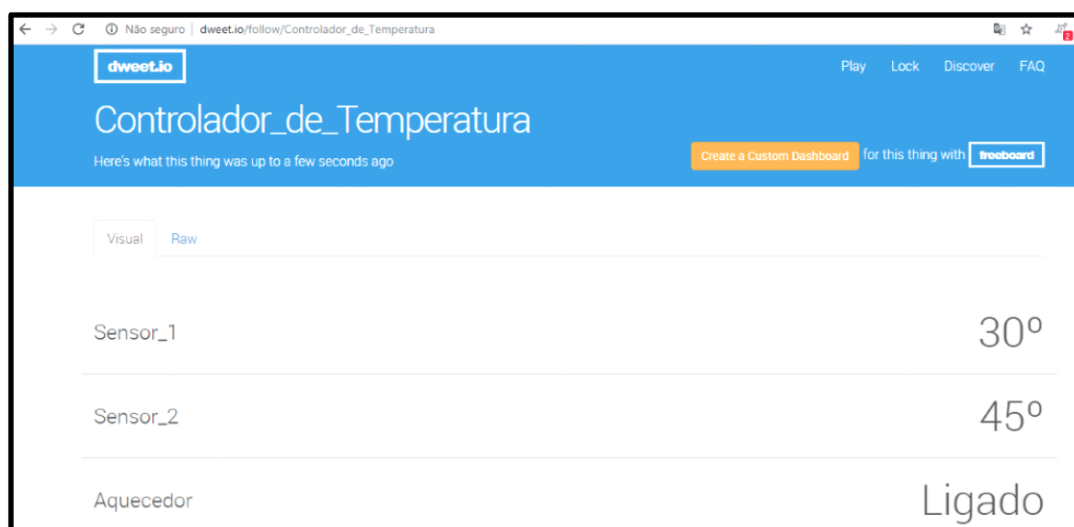
2.5.3 Dweet.io

Figura 15 é uma plataforma online para troca de mensagens e alertas para IoT (Internet das Coisas), sendo praticamente uma espécie de *Twitter* para máquinas e/ou sistemas que possuam uma conexão com a internet (DWEET, 2018).

Figura 15 - Homepage do *Dweet.io*

Fonte: Dweet.io (2018)

Nessa plataforma é possível que se crie um ou mais *links* onde sejam feitas postagens do interesse do usuário (Figura 16), seja por um sistema ou pelo usuário propriamente dito, e essas informações ficam à disposição para qualquer outro sistema conectado à internet por um tempo determinado.

Figura 16 - Exemplo de postagens no *Dweet.io*

Fonte: Dweet.io (2018)

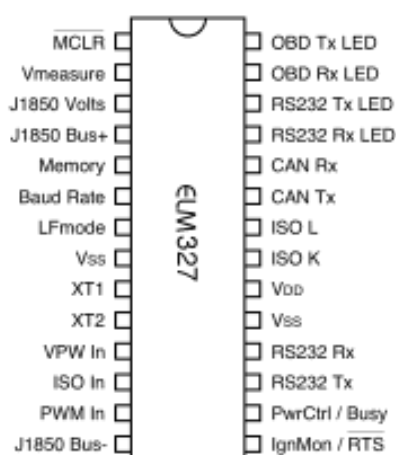
Apesar da facilidade existente no manuseio dessa ferramenta, os *links* criados são abertos a quem quiser acessar, a não ser que o usuário queira “trancar” um *link*, de forma a restringir o acesso por sistemas não autorizados, porém esse recurso está disponível apenas para *DweetPRO*, que é uma derivação paga do *dweet.io*.

2.7 ELM 327

Todos os veículos fabricados atualmente possuem um padrão de diagnósticos conhecido mundialmente por OBD-II, onde a transferência de dados nessas interfaces segue diversos padrões de comunicação, não podendo ser acessado diretamente por computadores, celulares, *tablets*, entre outros. O ELM327 é utilizado como um receptor que recebe os dados e transmite para algum dispositivo inteligente, além de não utilizar uma memória interna (ELM, 2018).

O ELM327 fornece suporte para comunicações de alta velocidade, sendo capaz de identificar automaticamente nove protocolos OBD, de acordo com o *datasheet do ELM*. Pode ser considerado uma ferramenta didática onde o usuário tem total liberdade para a criação de vários projetos destinados a leitura de dados. O CI do ELM pode ser visto na Figura 17 demarcado com suas devidas entradas (ELM, 2018).

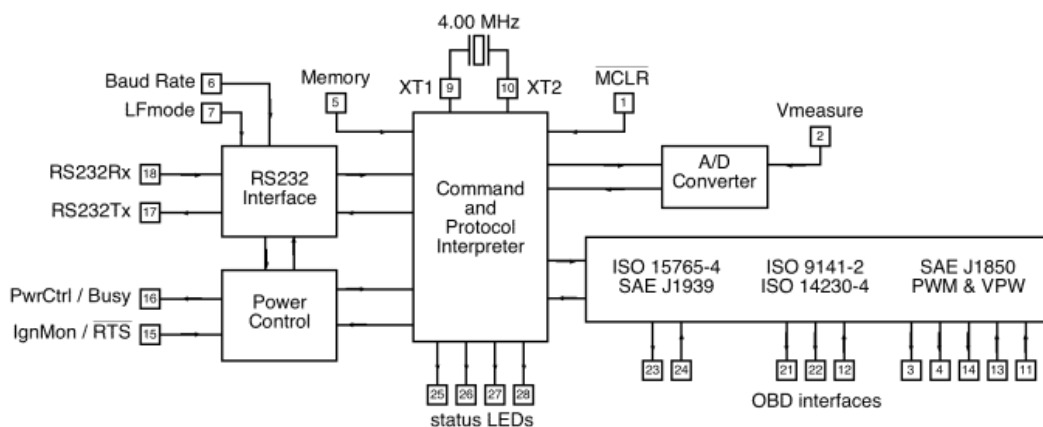
Figura 17 - CI ELM327



Fonte: ELM Electronics (2018)

Na Figura 18 é possível ver de uma forma mais ampla, como o CI ELM327 recebe os dados e manipula eles de uma forma organizada.

Figura 18 - Diagrama de blocos do ELM327



Fonte: ELM Electronics (2018)

O conector ELM327 possui conexão *Bluetooth*, ou seja, a transferência de dados entre o veículo e um celular por exemplo não precisa de cabos para-se fazer a monitoria do sistema, tendo como leitura o RPM do motor, a temperatura, a pressão do coletor de admissão, código de falhas do sistema, entre outros. Esse conector pode ser visto na Figura 19, e no Anexo A é possível ver o circuito que compõe esse sistema de diagnóstico.

Figura 19 - Conector ELM327



Fonte: Nergiza (2014)

2.8 Módulo GPS NEO 6M

O módulo NEO 6M (Figura 20) faz parte de uma família de receptores *stand-alone* de GPS da u-blox, sendo esse um receptor flexível e de baixo custo que oferece inúmeras possibilidades de conectividade em seu pequeno chip, sendo este ideal para aplicações em dispositivos móveis alimentados por bateria onde o custo e espaço são muito restritos. (UBLOX, 2011)

Figura 20 - Módulo GPS NEO 6M



Fonte: U-blox (2011)

Esse receptor possui uma estratégia de localização de 50 canais em 6 posições, possuindo um tempo abaixo de um segundo para primeiro "fix", termo esse normalmente utilizado quando o GPS se encontra no espaço, conhecido também como *Time-To-First-Fix*. (UBLOX, 2011)

O seu sistema dedicado de aquisição, com 2 milhões de correlacionadores, é capaz de massivas buscas paralelas em espaço de tempo e frequência, possibilitando encontrar satélites instantaneamente. (UBLOX, 2011)

Requer alimentação de 2,7V até 3,6V, possuindo como tecnologias de interface: UART, USB, SPI e DDC, além de trabalhar nos protocolos: NMEA, UBX e RTCM. (UBLOX, 2011)

2.9 Raspberry PI 3

O *Raspberry* é um hardware de pequenas proporções que foi concebido no ano de 2006, na universidade de Cambridge, no Reino Unido.

Quando os professores observaram a queda de notas dos alunos, a fim de ajudar os mesmos, foi desenvolvido um hardware que funciona como mini PC tendo saídas de HDMI, saída de áudio, interface serial (USB), comunicação *Bluetooth* e acesso a *Ethernet* (LEITE,2015).

O *Raspberry* veio como uma importante ferramenta no desenvolvimento de novas tecnologias abrindo um range de infinitas possibilidades.

Este sistema possui especificações técnicas robustas (Quadro 4), para um hardware de pequenas proporções, consistindo em:

Quadro 4 - *Hardware do Raspberry PI 3*

Placa	<i>Raspberry Pi 3 Model B</i>
Processador	<i>Broadcom BCM2837</i>
CPU	Quadcore ARM Cortex-A53, 64Bit
Clock	1.2GHz (cerca de 50% mais rápido que a Pi 2)
RAM	1 GB
GPU	400 MHz <i>VideoCore IV®</i>
Conectividade de rede	1 x 10 / 100 <i>Ethernet</i> (RJ45 Port)
Conectividade sem fio	802.11n <i>Wireless LAN</i> (Wi-Fi) e <i>Bluetooth</i> 4.1, BLE
Portas USB	4 x USB 2.0
GPIOs	40 pinos
Interface com câmera	15-pin MIPI
Interface com display	DSI 15 Pin / HDMI Out / Composite RCA
Fonte (Corrente)	2.5 A

Fonte: Souza (2016)

O *Raspberry* é uma plataforma, que utiliza diferentes tipos de sistemas operacionais podendo ser:

- *Raspbian*, *software* padrão da plataforma, que usa como base o Linux sendo ideal para iniciantes em *Raspberry*.
- PIXEL, *software* baseado no Linux, com ênfase na interface com o usuário, possibilitando a aplicação de projetos com facilidade.
- Windows 10 IoT, *software* baseado no Windows 10, sendo uma versão de menor tamanho, com aplicações visando a praticidade do sistema com a internet.

2.9.1 Linguagem de programação para *Raspberry*

A *Raspberry* é uma plataforma que se adequa ao perfil do usuário, assim possibilitando que o mesmo, programe na linguagem que se sinta mais confortável.

As linguagens que podem ser utilizadas na *Raspberry* são: Python, Lua, LaTeX, C, C++, Java.

Dentre essas as que mais se destacam são: Python e C.

2.9.2 Python

O *Python* foi criado em 1989 por Guido Van Rossum, no centro de pesquisas, na Holanda.

A linguagem de programação do *Python* Figura 21 é de alto nível, possibilitando o usuário do mesmo utilize totalmente os seus recursos, com particularidades que ajudam os programadores na otimização do código.

Figura 21 - Logo da linguagem Python



Fonte: *Python* (2018)

O *Python* pode ser utilizado em diversos tipos de projetos (Figura 22), de pequenos a grandes, e tem sido o substituto da linguagem *java* em projetos com aplicações na internet. Foi adotado como linguagem inicial para ingressantes nas universidades (DALL'AGNOL,2014).

Figura 22 - Exemplo de código acionamento de LED com PWM

```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BOARD)
5 GPIO.setup(23,GPIO.OUT)
6 pwm=GPIO.PWM(23,100) #Configura para 100 Hz
7 pwm.start(0) #Inicia com DC=0%
8
9 while(1):
10     for dc in range(0,100,1):
11         pwm.ChangeDutyCycle(dc)
12         time.sleep(0.1)
```

Fonte: Souza (2016)

2.9.3 Linguagem C

A linguagem C teve sua origem em 1972, sendo uma evolução de uma linguagem mais antiga BCPL. Desenvolvida por Dennis Ritchie (PEREIRA,2001).

A principal funcionalidade que popularizou a linguagem C foi a sua flexibilidade na maneira de aplicar o código (Figura 23), sendo difundida em diversas áreas de atuação (PEREIRA,2001).

Figura 23 - Exemplo de código em C

```
238 void main(void)
239 {
240     InitHardware();
241     /* Initialize tasks */
242     InitTasks();
243     RecuperaOdometro();
244     //main loop
245     while(1)
246     {
247         EstadoChaveIgnicao = LeituraChaveIgnicao();
248         //Verification: check if there's a task to be executed
249         if ((Timer0IntGeneraed == YES) && (NUMBER_OF_TASKS))
250         {
251             Timer0IntGeneraed = NO;
252             ExecuteTask();
253         }
254     }
255 }
```

Fonte: O Autor

A linguagem C, tem maior complexidade em relação a aprendizado, por estes motivos tem sido substituída em algumas localidades por *Python* como linguagem padrão em universidades e escolas técnicas, visando um aumento de produtividade e entendimento dos estudantes. Porém C ainda está na maior parte das universidades e escolas técnicas como o padrão no estudo de programação.

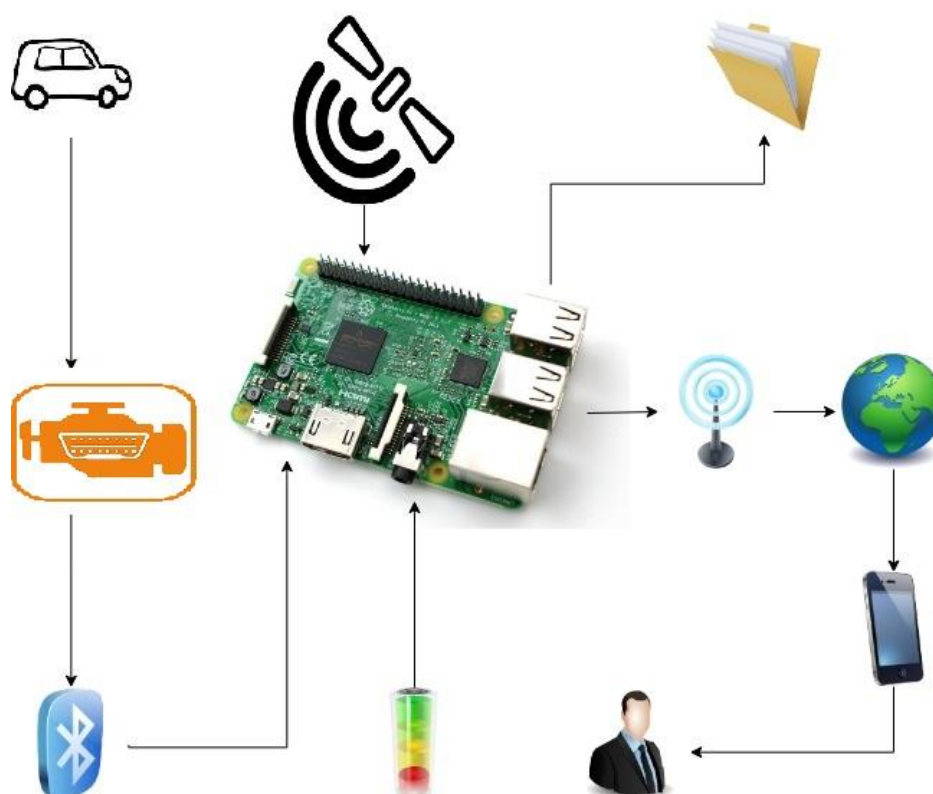
3 METODOLOGIA

Nesse capítulo serão abordadas todas as questões referentes ao método em fase prática, comentando detalhadamente cada fase de implementação do protótipo, desde a sua idealização até o produto final.

3.1 O projeto

O projeto idealizado funcionará basicamente conforme a Figura 24 abaixo, sendo esse projeto consistido em um sistema que monitora diversos dados operacionais de um veículo integrante de uma frota em tempo real, obtendo dados que vão desde a localização espacial atual, até dados como velocidade, rotação, temperatura do óleo, temperatura do líquido de arrefecimento, quantidade de combustível, dentre outros.

Figura 24 - Fluxograma do sistema SMOB



Fonte: O Autor

Além desses dados, códigos de falha também poderão ser lidos por esse sistema, sugerindo a quem estiver responsável pelo gerenciamento da frota tomar as devidas providências.

O responsável pelo gerenciamento de uma frota poderá acompanhar todas essas informações em uma central de monitoramento, ou de qualquer outro lugar de seu interesse por meio de um *smartphone* com sistema operacional *Android*, que esteja conectado à internet, por meio de um aplicativo.

A implementação desse aplicativo será feita através do *App Inventor*, que fará uma interface entre o usuário e os dados enviados pelo sistema embarcado.

O aplicativo receberá os dados de cada veículo de forma remota a partir do dispositivo conectado ao veículo, que deverá possuir uma conexão com a internet, realizando o envio dos dados previamente lidos pelo sistema para a plataforma *dweet.io*, que posteriormente fará interface com o aplicativo *Android*.

Como forma de obtenção dos dados operacionais, um *ELM327 Bluetooth* fica no conector OBD II do veículo, realizando a leitura dos dados necessários e enviando estes à *Raspberry*, que é responsável pelo processamento dos dados que circulam pelo sistema, buscando a melhor eficiência possível em questão de velocidade, tamanho de dados para transmissão, precisão de leitura e gerenciamento das informações.

Intrínseco a esse hardware um módulo GPS, mais especificamente o *NEO 6M u-Blox*, é aplicado para obtenção de informações referentes a localização do veículo, sendo elas: latitude, longitude e altitude.

Por segurança, no caso de insuficiência de alimentação do automóvel, um sistema de bateria suplementar é empregado para garantir autonomia de funcionamento por um tempo suficiente de alertar a central da anomalia de tensão de alimentação e realizar mais leituras para reduzir ao máximo a perda de informações.

Dessa forma a central perderá muito pouco ou quase nada das informações de cada veículo da sua frota, garantindo alta precisão no levantamento de dados relacionados a performance geral e acompanhamento muito próximo de ser em tempo real dos veículos em atividade.

Ainda sobre essa questão, do acompanhamento em tempo real, o grupo decidiu se aproximar ao máximo da taxa de atualização de 1Hz ou simplesmente uma atualização por segundo (frequência aplicada nos sistemas de telemetria veiculares) na amostragem de todos os dados para o usuário no aplicativo.

A maioria dos testes são realizados no laboratório técnico automotivo da Fatec Santo André e como veículos de teste o grupo utiliza o Gol G5 2009 da faculdade (OBDBr-1), um Agile 2011 (OBDII) e uma Audi A4 2012 (OBDII) (Figura 25 e Figura 26).

Figura 25 - Veículos de Teste: Audi A4 e Gol G5



Fonte: O Autor

Figura 26 - Veículo de Teste: Agile



Fonte: O Autor

4 PRÁTICA E RESULTADOS

4.1 Raspberry PI 3 e ELM327

Com as ferramentas que seriam necessárias e o objetivo do projeto, iniciaram-se as buscas pelos componentes.

O primeiro deles foi a *Raspberry PI 3*, que possui boa parte dos recursos para o desenvolvimento do projeto, como a comunicação *Bluetooth*, *Wi-fi*, entradas USB (*Universal Serial Bus*), pinos de entradas e saídas (GPIO – *General Purpose Inputs Outputs*) e a interface de programação para que a lógica do projeto funcione.

Depois da obtenção da *Raspberry PI* foi iniciada a fase de aprendizado com essa ferramenta, que até então era algo desconhecido, tanto pelo hardware quanto pela linguagem de programação aplicada (*Python*).

Para começar a usar a *Raspberry* foi necessário que um sistema operacional fosse carregado nela por meio de um cartão de memória SD, e para isso foi feito o *download* do sistema *Raspbian*, em um cartão de memória de 16 GB.

Após a instalação do sistema, a *Raspberry* ficou pronta para começar a ser programada, para isso o grupo se utilizou de diversas bibliotecas, que serão melhores explanadas mais adiante.

Nesse projeto um *Smartphone* foi aplicado como módulo de internet para permitir a conexão do sistema com a internet, como forma de tornar possível a transmissão de dados entre o sistema e o aplicativo.

A utilização do *Smartphone* veio da complexidade e preço elevado da utilização de um módulo de 3G, de forma que o custo será apenas o valor dos pacotes de internet utilizados em testes e na apresentação final, já que o *Smartphone* utilizado é de propriedade dos integrantes do grupo, para o caso do uso por uma empresa seria por meio de um celular corporativo.

A *Raspberry* dispõe de WiFi, assim sendo necessário apenas se conectar em uma rede de internet para ser capaz de se conectar a nuvem, desta forma o *Smartphone* tem a tarefa de servir como módulo 3G, roteando a sua internet para a *Raspberry*.

Em relação ao módulo para leitura da saída OBD foi necessária a decisão de qual tipo usar, se por fio, *wi-fi* ou *Bluetooth*, levando em consideração que antes de

iniciar a parte prática o grupo já tinha em mãos 4 conectores ELM327 para leitura de OBD.

Destes, referente a forma de transmissão dos dados lidos do OBD, se utilizam: de *Bluetooth*, um de *wi-fi* e um via cabo, e para selecionar o conector que melhor se adequa ao projeto foi cogitado o de transmissão via cabo, que teoricamente facilitaria a comunicação, entretanto foi constatado que este conector em questão apresentou algum mau funcionamento, pois não foi possível extrair nenhum dado.

O próximo conector testado foi o que se utilizava de *Bluetooth* em sua comunicação (Figura 27), após a configuração da *Raspberry* para esse tipo de comunicação, foi possível começar a receber dados do veículo, portanto definiu-se que esse seria o conector a ser trabalhado no decorrer desse projeto.

Figura 27 - Conector Utilizado: ELM327 Bluetooth



Fonte: O Autor

No desenvolvimento do código em *Python* foram utilizadas as bibliotecas *PythonOBD* e *Pynmea2* sendo essas de terceiros, as demais bibliotecas utilizadas são padrão *Python* disponíveis na própria *Raspberry*, sendo necessário apenas incluí-las dentro do código.

Para a tornar possível a comunicação entre a *Raspberry* e o conector OBD, foi utilizado o *Bluetooth*, porém com os dados convertidos em uma porta serial, sendo necessário modificar o arquivo fonte do *Bluetooth* para que o sistema sempre reconheça o conector OBD com a mesma porta serial.

Com o *GPS* foi necessário realizar um processo similar ao da comunicação descrita anteriormente, porém o mesmo utiliza comunicação *USART*, com portas já disponíveis na *Raspberry* sendo necessário apenas habilitar dentro do código fonte.

A estratégia do software empregado na *Raspberry*, tem como base coletar os dados do *OBD*, *GPS*, bateria suplementar e posteriormente o envio dos mesmos para o *Dweet.io*.

A coleta dos dados *OBD*, passa por três etapas para o preenchimento da matriz de dados, a primeira etapa consiste em enviar o comando para o *OBD* (o comando está relacionado ao dado requerido, como por exemplo: *RPM*), na segunda etapa o sistema espera a resposta do *OBD* com os dados, na terceira etapa é guardado os dados na matriz. Este processo é realizado para cada dado adquirido via *OBD*.

Os dados do *GPS*, são recebidos na linguagem *NMEA* em blocos, sendo necessário selecionar o bloco e os dados de latitude, longitude e altitude.

Por esse sistema necessitar de múltiplas leituras para recepção dos dados foi empregado um laço com dez leituras onde o dado é guardado em cada leitura e comparado com um valor nulo, caso o valor seja nulo o dado anterior fica mantido e posteriormente guardado na matriz de dados.

A bateria suplementar possui estratégia na comparação dos sinais recebidos via portas digitais da *Raspberry*, o Quadro 5 que demonstra a lógica para a medição do nível da tensão da bateria suplementar.

Quadro 5 - Estratégia para leitura de carga da bateria suplementar

Pino 11	Pino 12	Saída
0	0	Cheio
0	1	Médio
1	1	Vazio

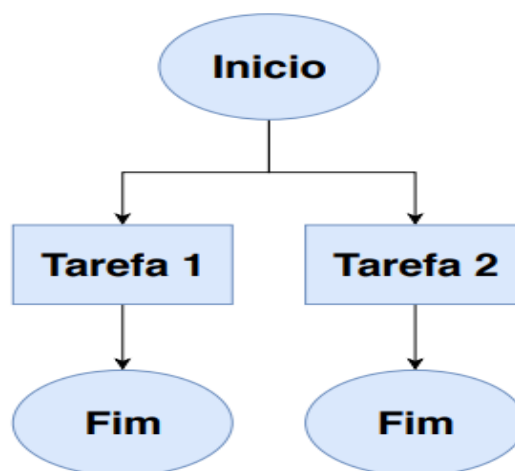
Fonte: O Autor

A função de envio de dados para o *Dweet* é consideravelmente simples, necessitando apenas da transformação da matriz de dados em uma *string*, onde a resultante é enviada junto ao link do servidor para o *Dweet*.

A estrutura do *software* empregado na *Raspberry* se baseia em um sistema multitarefa.

O sistema multitarefa (Figura 28) foi escolhido para um maior dinamismo do sistema assim diminuído os tempos de recepção dos dados e envio dos mesmos, trazendo um paralelismo ao sistema.

Figura 28 - Fluxograma Multitarefa



Fonte: O Autor

No caso o *software* da *Raspberry* foi desenvolvido com duas tarefas (Apêndice A), sendo a Tarefa 1 responsável pela a obtenção dos dados de OBD, GPS e bateria suplementar, a Tarefa 2 faz o envio dos dados para o servidor *Dweet* e também guarda os dados no arquivo "SMOB.txt" criado dentro da memória da *Raspberry* para possíveis auditorias ou consultas do frotista , funcionando como um *data logger*.

4.2 Alimentação do sistema e bateria suplementar

Para todo esse sistema funcionar *on board* é necessário que sua alimentação seja obtida do veículo, a solução pensada para o protótipo foi utilizar a saída 12V do painel de instrumentos, pois qualquer outra possibilidade seria muito invasiva no veículo e demandaria uma adaptação em cada veículo de teste.

Essa não seria a alternativa real pensando em um produto final, pois o motorista perderia quaisquer outras aplicações que possam se utilizar dessa alimentação.

Foi decidido então criar um sistema de bateria suplementar que vai gerenciar a alimentação do sistema.

O sistema de bateria suplementar, denominado também como BATSU, funciona como um "Nobreak", o sistema utiliza um transistor *Mosfet* tipo N, que chaveia o *Pack* de Pilhas caso a alimentação do veículo torne-se insuficiente para manter o sistema funcionando ou haja alguma tentativa de burlar o sistema.

Como ponto de saída do sistema de bateria suplementar, ela passa por um regulador de tensão *Lm2596* que fornece a tensão de 5 volts, com corrente máxima de até 3 amperes.

O circuito pós regulador fornece alimentação para a *Raspberry* e também alimenta a referência do circuito de nível de bateria suplementar. O esquema elétrico da bateria suplementar pode ser observado no Apêndice B.

A ideia desse sistema funcionar como um "Nobreak" veio após uma constatação nos primeiros testes de que, essa saída de 12V previamente mencionada apenas funciona com a chave do veículo na posição de ignição, além de também ter sua alimentação cortada durante a partida do veículo.

E pensando na possibilidade de perda de alimentação do sistema devido a esse problema citado, perda de carga da bateria ou simplesmente desconexão de um dos polos, foi justificada essa intenção de implementar essa bateria suplementar mencionada, para que esta atue quando a alimentação do veículo se tornar insuficiente para alimentar o sistema.

A importância da existência desse sistema no projeto é para garantir que quando não haja alimentação vinda do veículo o sistema ainda tenha capacidade de gerar uma mensagem notificando a avaria e enviar o máximo de dados possíveis, antes que a bateria suplementar se esgote.

O circuito que faz a medição do nível de bateria possui três divisores de tensão um para diminuir a tensão da bateria suplementar para a referência utilizada e outros dois que estão relacionados a cada nível de tensão, o sistema funciona de tal maneira que compara tensão do *pack* de pilhas com as referências de tensão utilizadas.

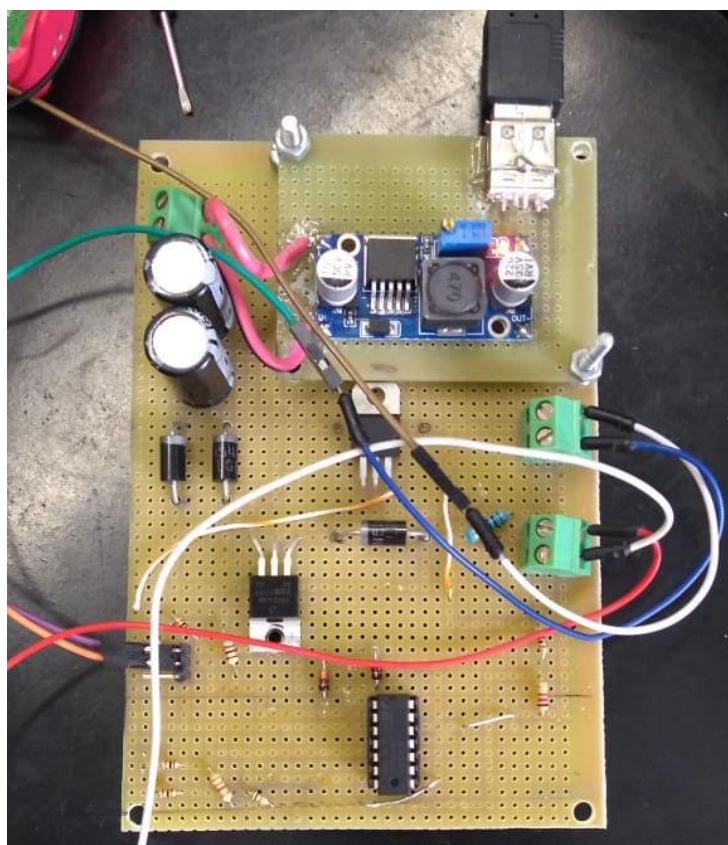
O objetivo é utilizar a leitura da bateria do veículo para controlar o acionamento da bateria suplementar, de forma que só seja utilizado esse sistema quando realmente

for necessário, conforme exemplos citados nos parágrafos anteriores, visando máxima durabilidade de carga.

Os sinais de saída são controlados pelo amplificador operacional em função de como está a comparação entre a tensão do *pack* de pilhas e a referência, gerando assim dois pinos para implementação de lógica na *Raspberry*.

O esquema elétrico desse circuito medidor de bateria pode ser encontrado no Apêndice C e sua montagem em placa na Figura 29 a seguir:

Figura 29 - Placa do sistema de alimentação e bateria suplementar



Fonte: O Autor

4.3 Dados lidos e enviados pelo sistema

Após desenvolver o código em *Python* na *Raspberry* para leitura dos dados, foi necessário selecionar quais das informações disponíveis na biblioteca OBD seriam implementadas para leitura e posterior envio ao usuário, e, para isso, foi decidida a leitura dos seguintes dados (Tabela 2):

Tabela 2 - Dados lidos pelo ELM327 e suas respectivas unidades

DADOS LIDOS POR OBD	UNIDADE
Rotação do motor	RPM
Velocidade do veículo	km/h
Abertura da válvula borboleta	%
Temperatura do ar na admissão	°C
Distância percorrida desde último DTC	Km
Temperatura do motor	°C
Vazão de ar na admissão	g/s
Pressão na admissão	kPa
Distância percorrida com a MIL ligada	km
Tempo de motor ligado	s
Nível de Combustível	%
Temperatura ambiente	°C
Porcentagem de etanol	%
Tensão da bateria	V
Número de DTC's	Número decimal
DTC's	Código de Falha + Descrição

Fonte: O Autor

Além desses dados lidos por comunicação OBD, informações como latitude, longitude, altitude e situação da bateria suplementar podem ser obtidos pelo hardware adicional aplicado na *Raspberry*, nos pinos de GPIO, TX, RX e de comunicação UART, sendo eles, respectivamente: BATSU (Bateria Suplementar) e GPS. (Tabela 3)

Tabela 3 - Dados lidos pela *Raspberry*

DADOS LIDOS POR GPIO (BATSU)	UNIDADE
Status da bateria suplementar	Ligado ou Desligado
Carga da bateria suplementar	Cheia, Média ou Vazia
DADOS LIDOS POR UART (GPS)	UNIDADE
Latitude	-90 até 90
Longitude	-180 até 180
Altitude	m

Fonte: O Autor

Apesar dos dados a serem lidos terem sido definidos, alguns dados não puderam ser obtidos, variando de carro para carro, sendo que o maior déficit de informações pode ser encontrado no Gol G5, principalmente devido a sua comunicação de OBDBr-1.

Logo abaixo na Tabela 4 é possível conferir um comparativo relacionado aos dados em que se obteve sucesso na leitura para cada carro de teste.

Tabela 4 - Dados lidos em cada veículo de teste

DADOS LIDOS POR OBD	Agile (OBDBr-2)	Audi A4 (OBD II)	Gol G5 (OBDBr-1)
Rotação do motor	✓	✓	✓
Velocidade do veículo	✓	✓	✓
Abertura da válvula borboleta	✓	✓	✓
Temperatura do ar na admissão	✓	✓	✓
Distância percorrida desde último DTC	✓	✗	✗
Temperatura do motor	✓	✓	✓
Vazão de ar na admissão	✓	✓	✓
Pressão na admissão	✓	✓	✓
Distância percorrida com a MIL ligada	✓	✗	✗
Tempo de motor ligado	✓	✓	✗
Nível de Combustível	✓	✓	✗
Temperatura ambiente	✓	✓	✗
Porcentagem de etanol	✗	✗	✗
Tensão da bateria	✓	✗	✗
Número de DTC's	✓	✓	✓
DTC's	✓	✓	✓

Fonte: O Autor

É importante ressaltar que os dados que não conseguiram ser lidos na verdade apresentaram fundo de escala na apresentação de seus valores, portanto não apresentam dados reais/coerentes do instante de leitura.

A partir da análise dessa tabela, é notável a disposição de mais dados via OBD II (Agile e Audi A4) em relação ao OBDBr-1 (Gol G5) e que o carro que mais disponibilizou dados válidos foi o Agile.

Como observações importantes a serem citadas referente aos testes nesses veículos é que por alguma razão não foi possível obter tempo de injeção de combustível em nenhum deles, todos apresentaram o valor de -210 ms, o que subentende-se por ser fundo de escala deste parâmetro.

Além disso, na leitura de porcentagem de Etanol tem-se marcado insucesso nos três veículos, porém em um breve teste em uma Ford *Ecosport* este valor apresentou uma leitura válida, portanto foi decidido manter esse parâmetro.

4.4 Envio de dados para a plataforma *Dweet.io*

Em função da necessidade de uma ferramenta online para recebimento dos dados operacionais lidos foi pensada a implementação de um banco de dados específico para aplicação.

Após algumas pesquisas, foi identificado um sistema de fácil utilização para envio e leitura de dados online, a plataforma *dweet.io*, utilizada para sistemas com aplicação de IoT, que não é um sistema de banco de dados, porém se adequa a proposta do trabalho.

Em função desta ferramenta, foi feita a alteração no programa principal para que depois de ler cada dado operacional seja feito o envio destes para um *link* no *dweet.io* denominado Sistema de Monitoramento *On Board* (SMOB).

Para realizar a escrita em um determinado *link* no *dweet.io* é necessário usar o diretório de um navegador, como se fosse acessar um site qualquer, mas na realidade é um comando para escrita, sendo o comando: “*dweet.io/dweet/for/SMOB?* [dados_a_serem_enviados]”, após entrar nesse *link* uma tela informará que você criou uma nova página no *dweet* e que você postou o que estava em [dados_a_serem_enviados].

Em relação a leitura de um *link* no *dweet.io* segue a mesma lógica de comando pelo diretório de um navegador, sendo o comando: “*dweet.io/follow/SMOB*”, a partir disso uma página será aberta e os dados previamente escritos poderão ser lidos.

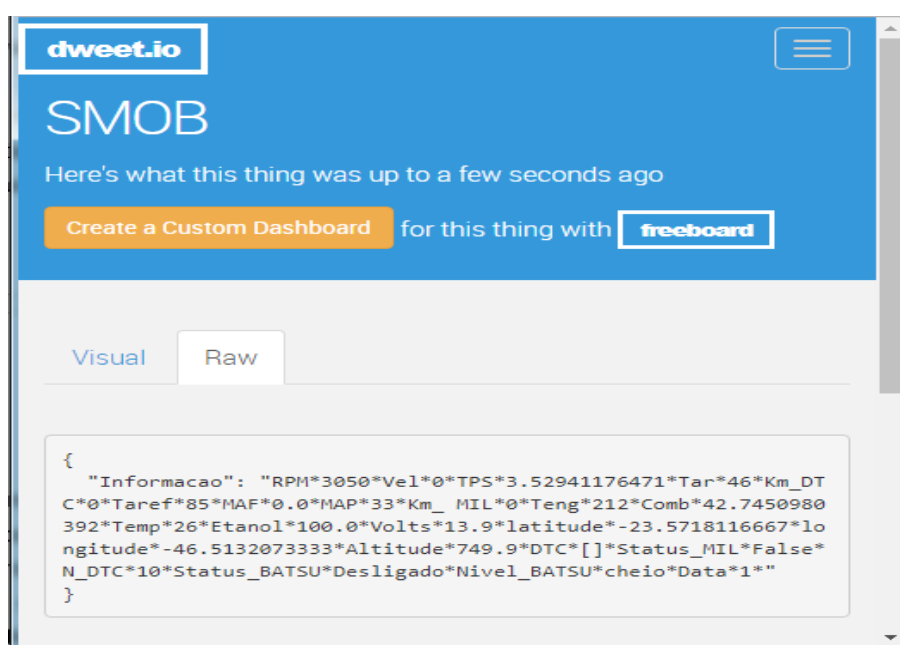
Entretanto foi identificado que essa plataforma possui um limite de tempo de um segundo entre cada postagem feita.

Esse problema acabou ocasionando um erro referente a frequência de atualização dos dados, visto que eram 23 informações a serem enviadas, dessa forma

um dado operacional demoraria por volta de 23 segundos para ser atualizado no *dweet.io*, o que não traria uma amostragem agradável ao usuário.

Como solução para esse problema de apenas uma informação a ser enviada por segundo, foi decidido enviar um pacote com todos os dados operacionais coletados, ou seja, primeiro fazer a coleta desses 23 dados e posteriormente enviá-los ao *dweet.io* conforme podemos observar na Figura 30.

Figura 30 - Página no *dweet.io* com todos os parâmetros lidos do OBD



Fonte: O Autor

Dessa forma foi possível ter uma amostragem mais rápida dos dados, partindo do tempo de 23 segundos para atualização de um único parâmetro para um tempo de aproximadamente 3 segundos para atualização de todos os parâmetros de leitura, entretanto ainda eram necessárias melhorias na lógica do sistema para que a taxa de amostragem dos dados se aproxime de 1 segundo, o que é o principal objetivo para o acompanhamento em tempo real.

Para mandar todos os parâmetros em apenas um *link* foi definida uma determinada forma de preenchimento, sendo que no campo de preenchimento dos dados, as informações vão sendo alocadas, por exemplo, da seguinte forma:

*RPM*2000*TPS*3,676* [...], ou seja, uma *string* com o que é o dado a ser preenchido e o valor do dado lido, separados por caracteres '*' (asteriscos).

Posteriormente o aplicativo fara a leitura desse *link* de forma a separar e alocar os dados de acordo com sua ordem de escrita.

4.5 Criação do aplicativo com o App Inventor

Como forma de tornar possível a amostragem dos dados lidos pelo sistema, foi implementado um aplicativo a partir do uso da ferramenta *App Inventor*, que se mostrou algo eficiente o suficiente para a aplicação.

Para criação do aplicativo as primeiras questões levantadas foram as questões de como se chamaria esse aplicativo e qual seria o seu logotipo.

Após algumas tentativas de logotipo para o aplicativo, foi fixado que a Figura 31 a seguir representaria o projeto.

Figura 31 - Logo SMOB



Fonte: O Autor

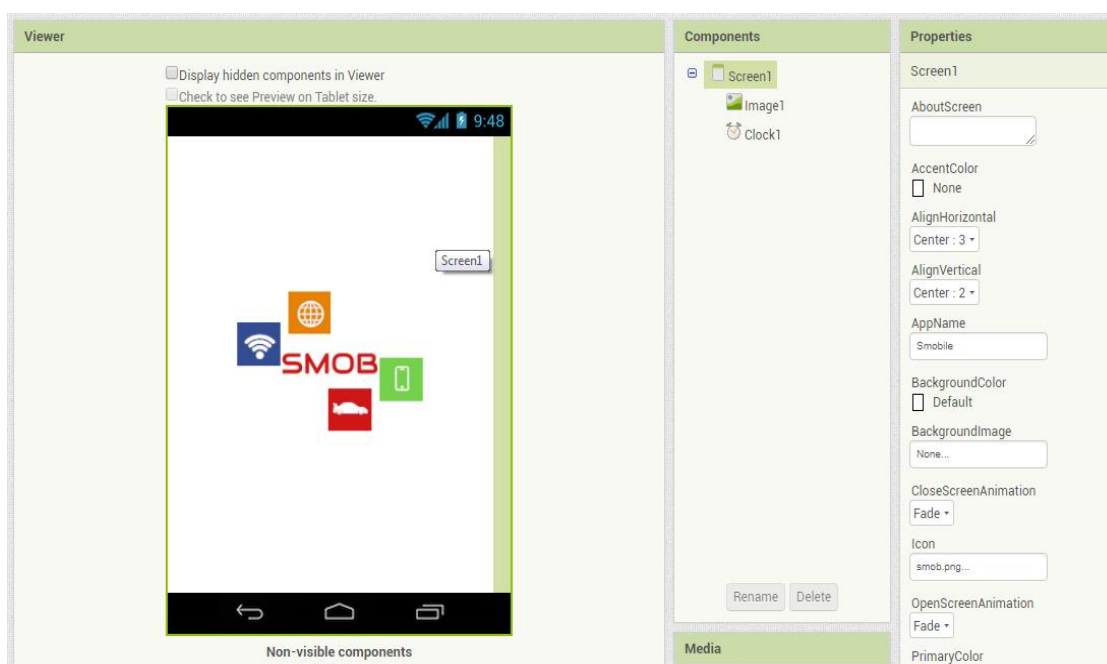
Em relação ao nome do aplicativo, foi decidido pelo grupo que seu nome seria *Smobile*.

A dinâmica de criação no *App Inventor* é feita por *screens*, ou simplesmente telas, que nada mais são que as telas que irão compor o aplicativo, e cabe ao

programador decidir o que vai estar nessa tela, como o que está nessa tela irá se comportar, o fluxo de dados para lógica e amostragem das *screens*.

Na primeira *screen* do aplicativo (Figura 32) é mostrado o logotipo do aplicativo, por um período de 3 segundos, após esse intervalo passa-se para a segunda *screen*.

Figura 32 - Primeira *screen* do aplicativo e o ambiente de desenvolvimento



Fonte: O Autor

Na *screen 2* (Figura 33), tem-se a interface para o frotista se identificar por meio de um usuário e uma senha, onde há toda uma lógica implementada que protege o acesso em caso de usuário e/ou senha errados, além de uma pergunta no caso do usuário pressionar o botão 'voltar', para confirmar o fechamento do aplicativo ou retornar para a tela normalmente devido a mudança de ideia ou aperto acidental do botão 'voltar'.

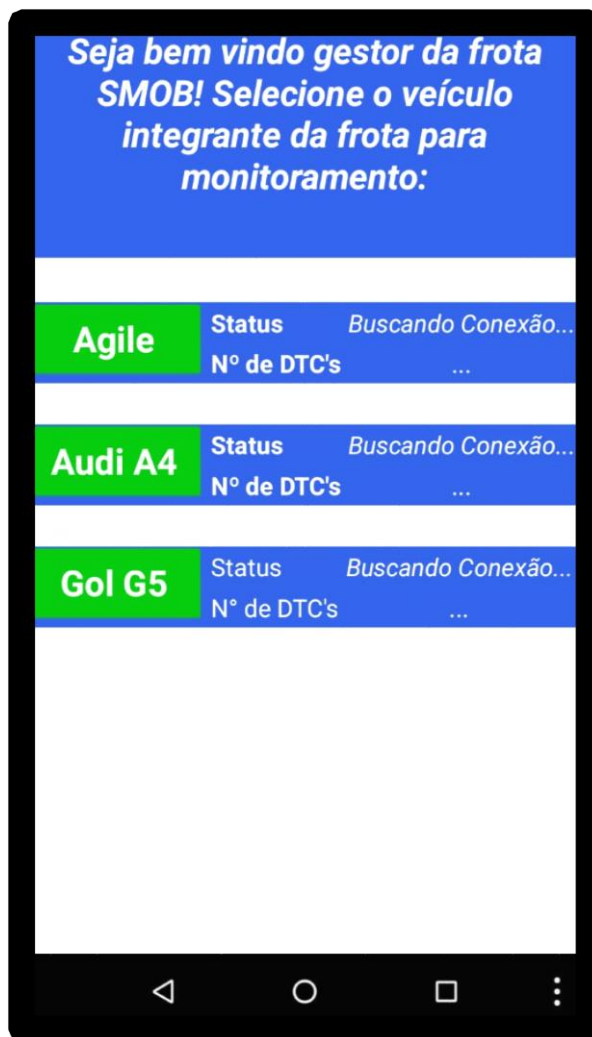
Figura 33 - Segunda *screen* – digitação de usuário e senha

Fonte: O Autor

Para fins de testes no aplicativo e padronização de um *login* foi criado como único usuário e senha válidos 'Smob' e '1234', respectivamente.

Após a digitação correta do usuário e senha é mostrada a *screen* 3 (Figura 34), que apresenta ao frotista os veículos pertencentes a frota, o seu *status* de conexão com cada veículo e o número de DTC's.

Figura 34 - Terceira screen – seleção de veículo



Fonte: O Autor

Essa tela permite que o frotista inicie o monitoramento ao apertar de um botão que está escrito o nome de cada veículo previamente inserido na frota, e para isso foram inseridos os três veículos de teste desse projeto, o *Agile*, a *Audi A4* e o *Gol* da Fatec.

Após a seleção do veículo desejado, uma tela com a disponibilização do dados lidos do veículo pode ser observada (Figura 35), sendo o *template* igual para todas as telas, apenas com a mudança do nome do veículo sendo acompanhado logo no primeiro texto da tela.

Figura 35 - Tela de acompanhamento operacional dos veículos da frota

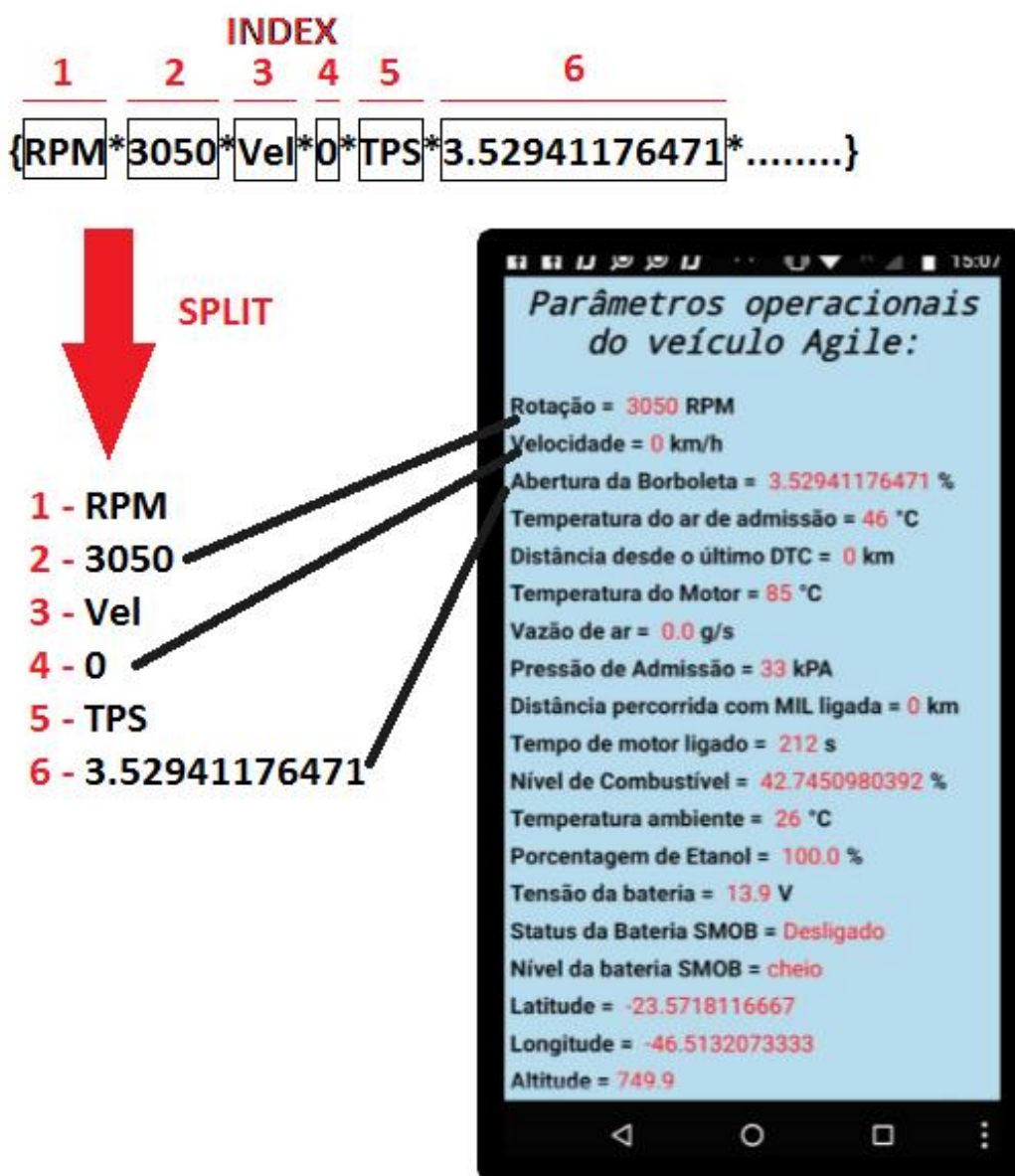


Fonte: O Autor

Para preenchimento dos dados apresentados nessa *screen* o aplicativo recebe um *link* do *dweet.io* com todas as informações escritas por extenso e separadas por asterisco, a programação implementada nessa *screen* é receber o *link* e aplicá-lo em uma função chamada *Split*.

A função *Split*, lê tudo o que há entre um asterisco e outro, separando por *index*, ou índices.

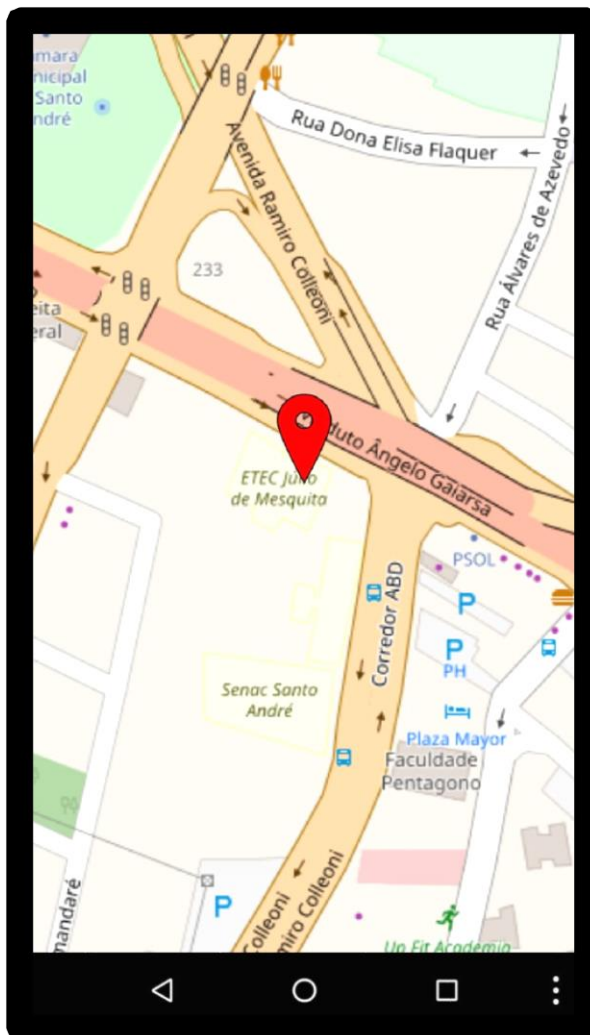
Durante essa separação o próprio aplicativo relaciona os *index* de leitura com o campo a ser preenchido na tela, da forma como é apresentada na Figura 36.

Figura 36 - Função *Split* do App Inventor

Fonte: O Autor

Além das informações triviais apresentadas por essa *screen*, é possível exibir as DTC's, se houverem, e observar em um mapa a atual localização do veículo (Figura 37).

Figura 37 - Tela de amostragem de localização dos veículos

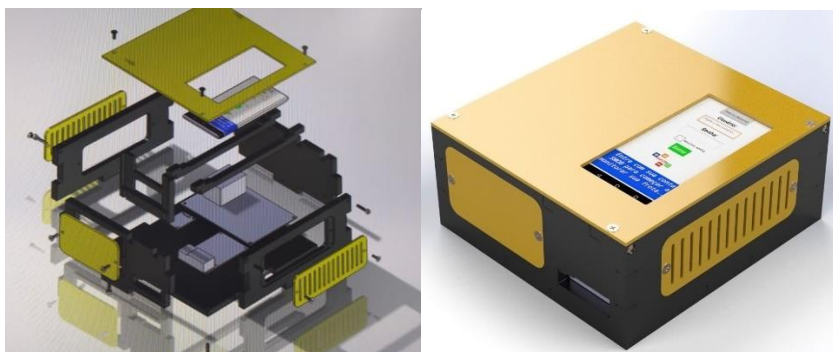


Fonte: O Autor

4.6 Caixa SMOB

Após feita toda a montagem e teste dos *hardwares* implementados nesse projeto foi feita a implementação de uma caixa protótipo para o sistema (Figura 38), que serve como habitat para o *hardware* e também para amostragem e divulgação do produto, visto que sua instalação no veículo como produto final ainda não fora discutida.

Figura 38 - Projeto da caixa SMOB



Fonte: O Autor

A partir do projeto foi encomendada a usinagem das peças de forma que o grupo realizou a montagem da caixa resultando no protótipo final na Figura 39 abaixo.

Figura 39 - Projeto final da caixa SMOB



Fonte: O Autor

5 CONCLUSÃO

Após a realização desse projeto de conclusão de curso foi possível comprovar que é possível a implementação do protótipo aqui descrito para monitoramento remoto de uma frota, pois diversos dados operacionais, localização e códigos de falha puderam ser obtidos a partir desse sistema e estes podem ser lidos de um *smartphone* a distância, conforme planejado nos objetivos.

O teste definitivo de monitoramento simultâneo de dois veículos ou mais infelizmente não foi possível devido a falta de *hardware*, porém alternando o *hardware* de um veículo para outro a leitura ocorreu sem problemas algum, dessa forma, caso os três veículos de teste possuíssem o *hardware* instalado o sistema funcionaria monitorando os três simultaneamente.

Pode-se dizer que este sistema substitui parte das funções de *scanner's* para levantamento de dados operacionais (modo contínuo) e leitura de códigos de falha, entretanto o sistema ainda não possui a capacidade de atuar na saída OBD, o que não estava nos objetivos do grupo, porém pode ser algo a ser implementado em uma continuação desse projeto.

É importante ressaltar que pouca lógica de pós tratamento foi implementada, visto que o objetivo pretendido foi a amostragem dos dados operacionais e localização.

A única coisa para análise profunda de dados que pode ser feita é na parte do *data logger* implementado nesse projeto, o que demandaria certo trabalho para uma análise pertinente.

Constatou-se que um pós tratamento desses dados lidos poderiam render muito conteúdo estatístico para análise, comparando com diversos outros veículos dentro de uma frota será possível ver fatores que se mostram anormais em relação a outros, com isso facilitaria implementar qualquer estratégia de melhoria no gerenciamento, porque vai estar explicito toda dinâmica de comportamento dos dados aumentando consideravelmente a eficiência de qualquer manutenção realizada, visto que o problema ou possibilidade de problema estará explicito ao gerente.

A partir disso é possível comprovar melhorias de consumo após troca de componentes, além de ser também possível comprovar a melhora comportamental

dos motoristas quando estes tiverem um treinamento, enfim, um *feedback* de diversos parâmetros passam a ser possíveis para o frotista.

Vale lembrar também que os dados operacionais disponíveis para leitura podem ser alterados nos códigos do projeto, caso algum parâmetro não seja realmente necessário ou algum outro seja de maior importância e possa possivelmente ser acrescentado para leitura.

Fica também a questão do projeto abrir portas para realização de quaisquer outras aplicações que possam envolver de alguma forma a leitura da saída OBD, uso de aplicativo celular, localização GPS, dentre outros.

Com os dados do GPS é possível que o gestor da frota possa identificar a produtividade e se motorista faz caminhos mais demorados que o normal, com isso ele pode controlar o percurso de cada motorista de forma que a empresa obtenha melhor rendimento e tenha redução de custos.

Com todos os dados obtidos o gestor poderá fazer uma análise melhor para que possa achar estratégias que tenham como o principal objetivo diminuir gastos e ao mesmo tempo manter o veículo apto a trabalhar todos os dias, sem que haja algum problema crítico.

5.1 Propostas Futuras

- Otimização do hardware.
- A utilização de multiprocessamento para cada tarefa empregada no projeto a fim de otimizar os tempos de execução e envio dos dados operacionais.
- Implementar um sistema de carregamento para bateria suplementar, além da mudança de pilhas para baterias de *ion lithium*.
- Construção de um hardware único, com todas as funções utilizadas.
- Elaboração de um modem 3g de baixo custo.
- Programa para gerenciamento de frota com otimização de rotas.
- Em uma aplicação em produto final, seria interessante uma implementação de um site que se conecte ao aplicativo e disponibilize todos os recursos do software em um computador, para ampliar a versatilidade do sistema.

6 REFERÊNCIAS BIBLIOGRÁFICAS

BASTOS, E. *Estudo das Diferenças dos Requerimentos das Principais Legislações de On Board Diagnostics para Padronização de Testes de Desenvolvimento e Validação de Transmissão Automática de Automóveis*. 2012. Disponível em: <<https://maua.br/files/monografias/estudo-das-diferencas-dos-requerimentos-das-principais-legislacoes-de-on-board.pdf>>. Acesso em: 20 abr. 2018.

BLUETOOTH. *Homepage*. 2018. Disponível em: <<https://www.bluetooth.com>>. Acesso em: 28 mar. 2018.

CBR. **What is IOS?**. 2018. Disponível em: <<https://www.cbronline.com/what-is/what-is-ios-4899783/>>. Acesso em: 27 mai. 2018.

CLUBE SCRATCH. **App Inventor 2**. 2015. Disponível em: <www.clubescratch-lamego.webnode.pt/news/app-inventor-2>. Acesso em: 12 jun. 2018.

COSTA. **GSM & GPRS**. 2005. Disponível em: <<http://www.logicengenharia.com.br/mcamara/alunos/GSM&GPRS.pdf>>. Acesso em: 20 mai. 2018.

DALL'AGNOL, Marcel de Sena. **Projeto De Dicipлина Para O Aprendizado Em Linux E Raspberry Pi**. 2014. Disponível em: <http://www.tcc.sc.usp.br/tce/disponiveis/18/182500/tce-26112015-094640/publico/DallAgnolMarcelde_Sena_tcc.pdf>. Acesso em: 25 abr. 2018.

DWEET.IO. *Homepage*. 2018. Disponível em: <www.dweet.io.com>. Acesso em: 30 set. 2018.

EBAH. **Tecnologia Telecom**. 2016. Disponível em: <www.ebah.com.br/content/ABAAAfozQAL/tecnologia-telecom?part=2#>. Acesso em: 12 abr. 2018.

ELM ELECTRONICS. **ELM 327**. 2018. Disponível em: <<https://www.elmelectronics.com/ic/elm327>>. Acesso em: 24 mai. 2018.

ESCOLA ANDROID. **Como Mudar Uma Função Padrão do Android**. 2015. Disponível em: www.escolaandroid.com/como-mudar-uma-funcao-padrao-do-android Acesso em: 20 abr. 2018.

FRANCO; DAMASCENO; PEREIRA. **GPS e GLONASS: Aspectos Teóricos e Aplicações Práticas**. 2002. Disponível em: <<https://www.decea.gov.br/sirius/index.php/2011/06/14/gnss-sistema-global-de-navegacao-por-satelite/>>. Acesso em: 23 abr. 2018.

FURLAN, Lidiane. **Aplicativo Auto localizador para Pessoas com Deficiência Visual**. 2015. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/7155/1/PB_COADS_2015_2_04.pdf>. Acesso em: 23 abr. 2018.

GIS RESOURCES. **Errors Can Be Reduced in VRS Page**. 2018. Disponível em: <www.gisresources.com/errors-can-be-reduced-in-vrs_page/>. Acesso em: 21 abr. 2018.

GUISS, Alexandre. **Google App Inventor: o criador de apps para Android para quem não sabe programar**. 2011. Disponível em: <<https://www.tecmundo.com.br/google/11458-google-app-inventor-o-criador-de-apps-para-android-para-quem-nao-sabe-programar.htm>>. Acesso em: 20 maio 2018.

INDE – Infraestrutura Nacional de Dados Espaciais. **Recomendações para levantamentos relativos estáticos – GPS**. 2008. Disponível em: <http://www.inde.gov.br/images/inde/recom_gps_internet.pdf>. Acesso em: 01 mai. 2018.

ISO. **ISO 9141: Road Vehicles – Diagnostic Systems**. 1994. Disponível em: <<https://www.iso.org/standard/55592.html>>. Acesso em: 20 abr. 2018.

ISO. **ISO 15765: Road Vehicles – Diagnostic on Controller Area Network (CAN)**. 2004. Disponível em: <<https://www.iso.org/standard/54498.html>>. Acesso em: 20 abr. 2018.

LEITE, Felipe. **Monitoramento De Vídeo Por Meio Do Computador Raspberry Pi: RASPBERRY PI**. 2015. Disponível em: <http://bdm.unb.br/bitstream/10483/14756/1/2015_FelipeLeiteScheidemantel_tcc.pdf>. Acesso em: 18 abr. 2018.

LEVHETA, Ricardo R. **Google Android aprenda a criar aplicações pra dispositivos móveis com o Android SDK**. 2ª. Ed. Editora: Novatec. 2010.

MACHADO, A. L., & Oliveira, B. R. **O Sistema OBD (On-Board Diagnosis)**. 2007. Disponível em: <http://ave.dee.isep.ipp.pt/~mjf/act_lect/SIAUT/Trabalhos%202007-08/Trabalhos/SIAUT_OBD.pdf>. Acesso em: 20 abr. 2018.

MARINO, Tiago. **GPS – Sistema de Posicionamento por Satélites Artificiais**. 2018. Disponível em: <<http://www.ufrj.br/lga/tiagomarino/aulas/7%20-%20GPS.pdf>>. Acesso em: 23 abr. 2018.

MCALINDEN; FELNSTEIN, SMITH. **Michigan Automotive Partnership Research Memorandum No. 2**. UMTRI Report No. 2000-24-2

MCCORD, Keith. **Automotive diagnostic systems**. Editora: Cartech. 2011.

MERCADO LIVRE. **Cables Adaptadores para Escaner OBD2 OBD1 Vários Modelos**. 2018. Disponível em: <www.articulo.mercadolibre.com.ve/MLV-476185982-cables-adaptadores-para-escaner-obd2-obd1-varios-modelos-_JM?quantity=1>. Acesso em: 23 abr. 2018.

MICROSOFT. **Windows 10 Internet das Coisas**. 2018. Disponível em: <<https://www.microsoft.com/pt-br/WindowsForBusiness/windows-iot>>. Acesso em: 22 abr. 2018.

MORIMOTO, Carlos. **Redes: Guia Prático 2ª Edição**. 2008. Disponível em: <<https://www.hardware.com.br/livros/redes/>>. Acesso em: 20 maio 2018.

MOTORSPOT. **Evolução do design automóvel, da caixa feia para a suavidade das curvas**. 2017. Disponível em: <www.motorspot.pt/2017/03/10/5732-evolucao-design-automovel/>. Acesso em: 20 mai. 2018.

NERGIZA. **OBD2 Conecta Tu Movil a Tu Coche y Conoce que Pasa Ahi Dentro**. 2014. Disponível em: <www.nergiza.com/obd2-conecta-tu-movil-a-tu-coche-y-conoce-que-pasa-ahi-dentro>. Acesso em: 12 jul. 2018.

NOVAES; VALENTE; PASSAGLIA; VIEIRA. **Gerenciamento de Transporte e Frotas**. 1. Ed. São Paulo: Perim, 2001.

OBD2-ELM327. **Obd2 Connector Location Appearance Layout**. 2018. Disponível em: <www.obd2-elm327.com/obd2-connector-location-appearance Acesso 20 maio 2018.-layout> Acesso em: 21 abr. 2018.

OLIVER, J. **Implementing the J1850 Protocol**. 2012. Disponível em: <www.intel.com/design/intarch/papers/j1850_wp.htm>.

PARÁ, Univercidade Federal. **MIT APP INVENTOR**. 2010. Disponível em: <http://www.aedmoodle.ufpa.br/pluginfile.php?file=/170097/mod_book/chapter/2317/MITAPP_Inventor_apostila.pdf>. Acesso em: 20 mai. 2018.

PEREIRA, Silvio do Lago. **Linguagem C**. 2001. Disponível em: <<https://www.ime.usp.br/~slago/slago-C.pdf>>. Acesso em: 13 maio 2018.

KNIO. **Pynmea2**. 2018. Disponível em: <www.github.com/Knio/pynmea2>. Acesso em: 12 out. 2018.

PYTHON. **Homepage**. 2018. Disponível em: <<https://www.python.org>>. Acesso em: 15 nov. 2018.

RAPPAPORT. **Comunicações sem fio – Princípios e práticas**. 2ª. Ed. Editora: Pearson Education do Brasil. 2009.

RIBAS, Wanderlei Kampa. **A importância do planejamento para missões de levantamentos GPS**. 2007. Disponível em: <http://www.esteio.com.br/downloads/2007/missoes_gps.pdf> Acesso em: 13 nov. 2018.

ROMER, Rafael. **GSM, 3G, EDGE, HPSA, 4G e LTE: entenda as siglas de conexão mobile**. 2013. Disponível em: <<https://canaltech.com.br/telecom/gsm-edge-hpsa-lte-entenda-as-siglas-de-conexao-mobile/>> Acesso em: 21 nov. 2018.

SAE. **SAE International**. 2002. Disponível em: <https://www.sae.org/attend/obd-ftp://ave.dee.isep.ipp.pt/~mjf/act_lect/SIAUT/Trabalhos%202007-08/Trabalhos/SIAUT_OBD.pdf>. Acesso em: 20 abr. 2018.

SEOS PROJECT. **GPS**. 2018. Disponível em: <www.seos-project.eu/modules/GPS/GPS-c01-p03.html>. Acesso em: 20 abr. 2018.

SILVA, Aricio Medeiros; PAIVA, Igor Galindo; FORTES, Denise Xavier. **Desenvolvimento de aplicativo para Android com uso do MIT APP INVENTOR**. 2017. Disponível em: <https://www.fasete.edu.br/revistarios/media/revistas/2017/desenvolvimento_de_aplicativo_para_android_com_uso_do_mit_app_inventor.pdf>. Acesso em: 20 maio 2018.

SIMÕES, Danielle dias; PEREIRA, Júlio César. **Sistemas Operacionais Móveis-Android X IOS**. 2014. Disponível em:

<<http://web.unipar.br/~seinpar/2014/artigos/graduacao/daniellediassimoes.pdf>>.

Acesso em: 20 maio 2018.

SIRIUS. **GNSS – Sistema Global de Navegação por Satélite**. 2011. Disponível em:

<<https://www.decea.gov.br/sirius/index.php/2011/06/14/gnss-sistema-global-de-navegacao-por-satelite/>>. Acesso em: 23 abr. 2018.

SJ IFSC. **Introdução aos Sistemas de Comunicação Celular**. 2018. Disponível em:

<www.sj.ifsc.edu.br/~mdoniak/ComunicacoesMoveis1/Cap1_IntroducaoComCelular_v3.pdf>. Acesso em: 25 abr. 2018.

SOUZA, Fábio. **O Hardware da Raspberry Pi3**. 2016. Disponível em:

<<https://www.embarcados.com.br/hardware-da-raspberry-pi-3/>>. Acesso em: 22 abr. 2018.

STADLER. **Segurança Em Redes Móveis**. 2013. Disponível em:

<https://www.ppgia.pucpr.br/~jamhour/RSS/TCCRSS11/Alisson%20Stadler%20de%20Paula%20_%20Seguranca%20em%20Redes%20Moveis.pdf>. Acesso em: 27 mai. 2018.

TELECO. **4G: Tecnologias de Celular**. 2017. Disponível em:

<http://www.teleco.com.br/4g_tecnologia.asp>. Acesso em: 27 mai. 2018.

TELECO. **Gerenciamento de Projetos I: Tecnologias das Redes Móveis**.

2018. Disponível em:

<http://www.teleco.com.br/tutoriais/tutorialgerswap1/pagina_4.asp>. Acesso em: 27 mai. 2018.

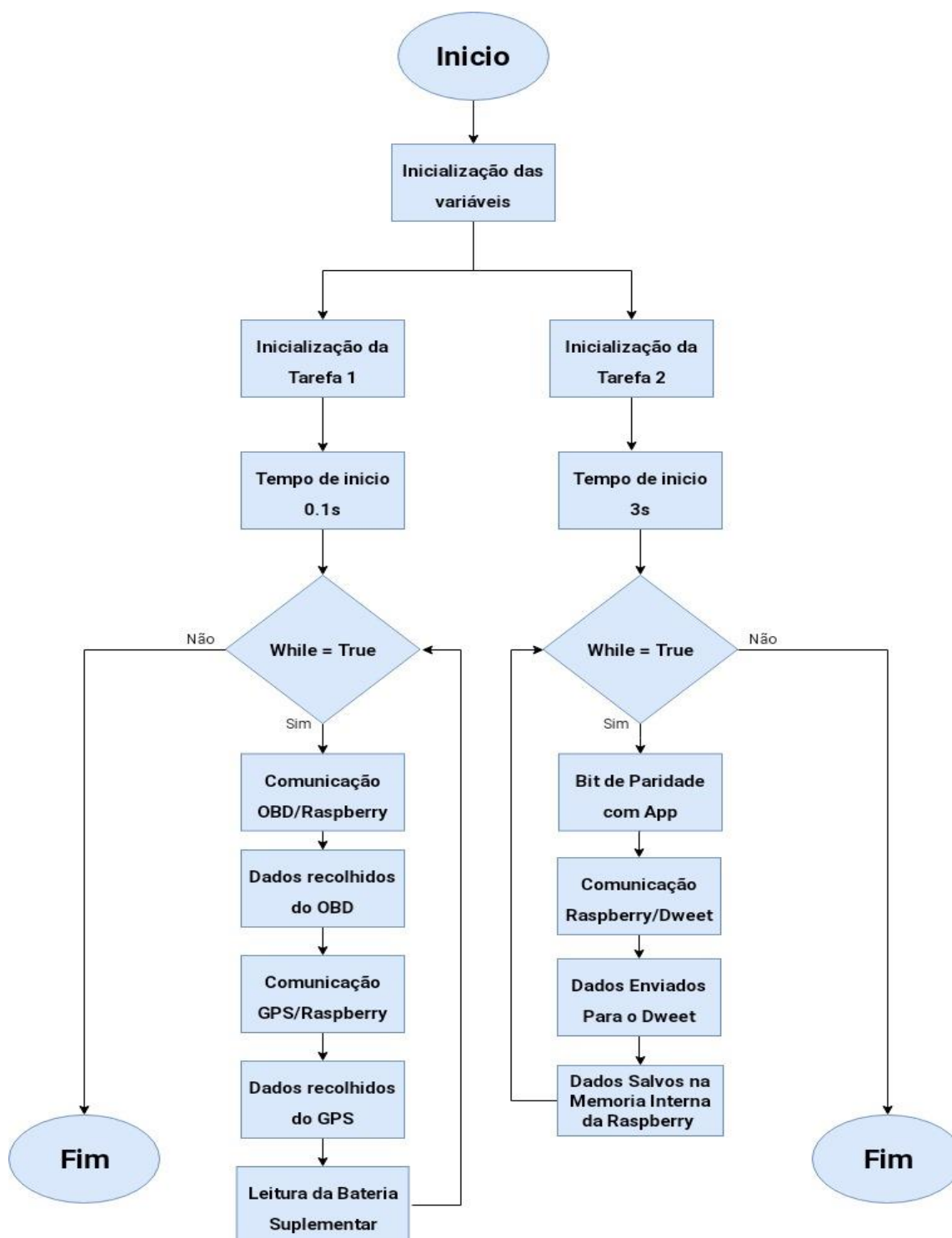
UBLOX. **U-Blox 6 GPS Modules**. 2011. Disponível em: <[https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_\(GPS.G6-HW-](https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf)

[09005\).pdf](https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf)>

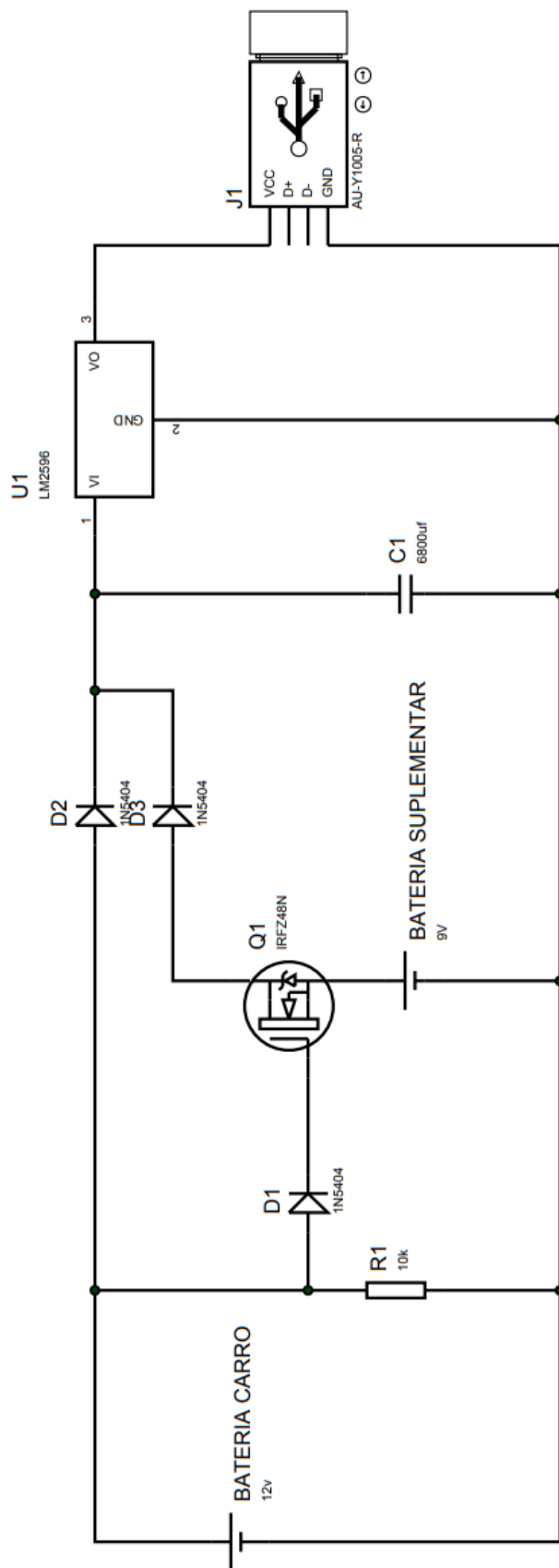
VAZ. **Tecnologias De Rastreamento De Veículos.** 2015. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/2009003/mod_resource/content/1/Telefonia%20Celular%20-%20Sistemas%20de%20Comunica%C3%A7%C3%A3o%20WAN2.pdf>. Acesso em: 27 mai. 2018.

WHITFIELD, Brendan. **Python-OBD.** 2018. Disponível em: <www.github.com/brendan-w/python-OBD>. Acesso em: 5 ago. 2018.

APÊNDICE A – FLUXOGRAMA DO SOFTWARE DA RASPBERRY

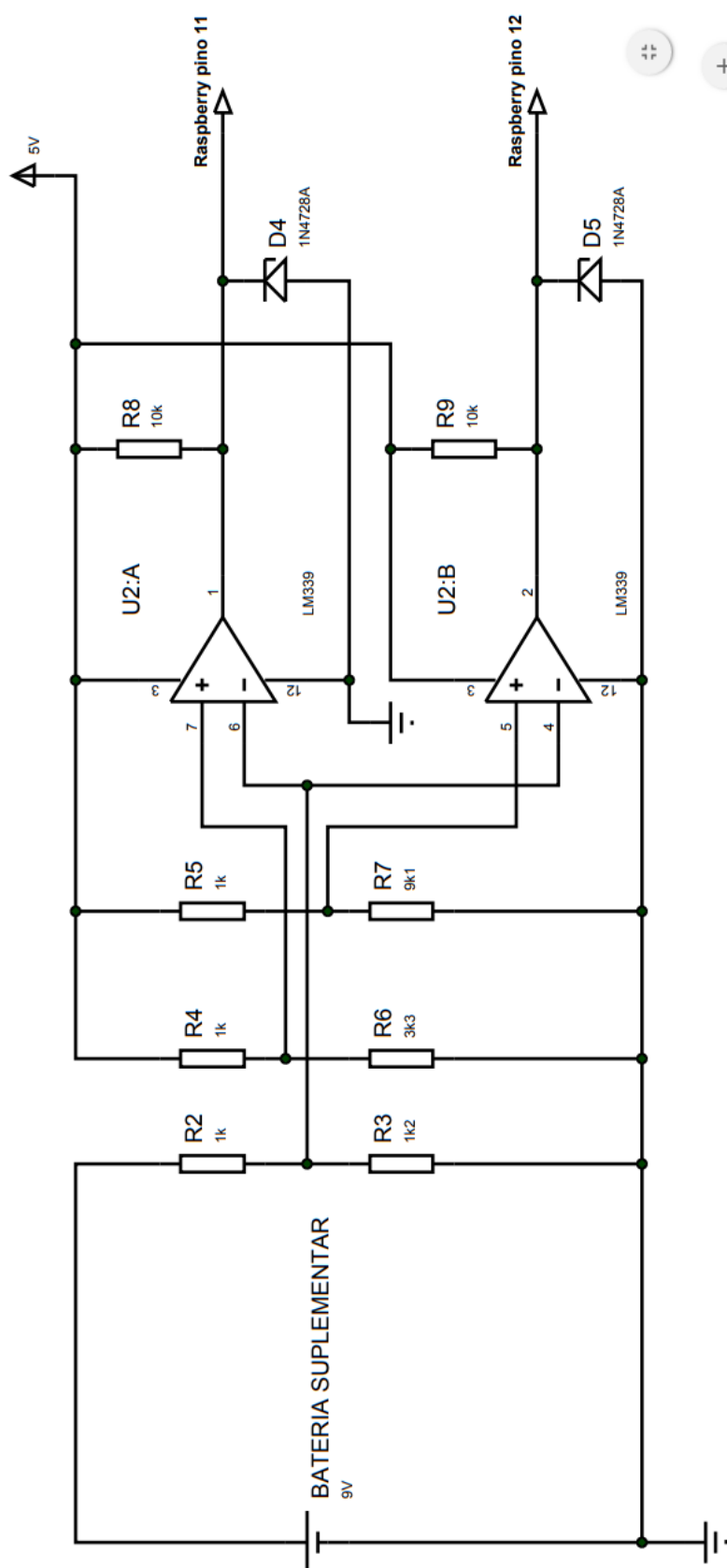


APÊNDICE B – CIRCUITO BATERIA SUPLEMENTAR



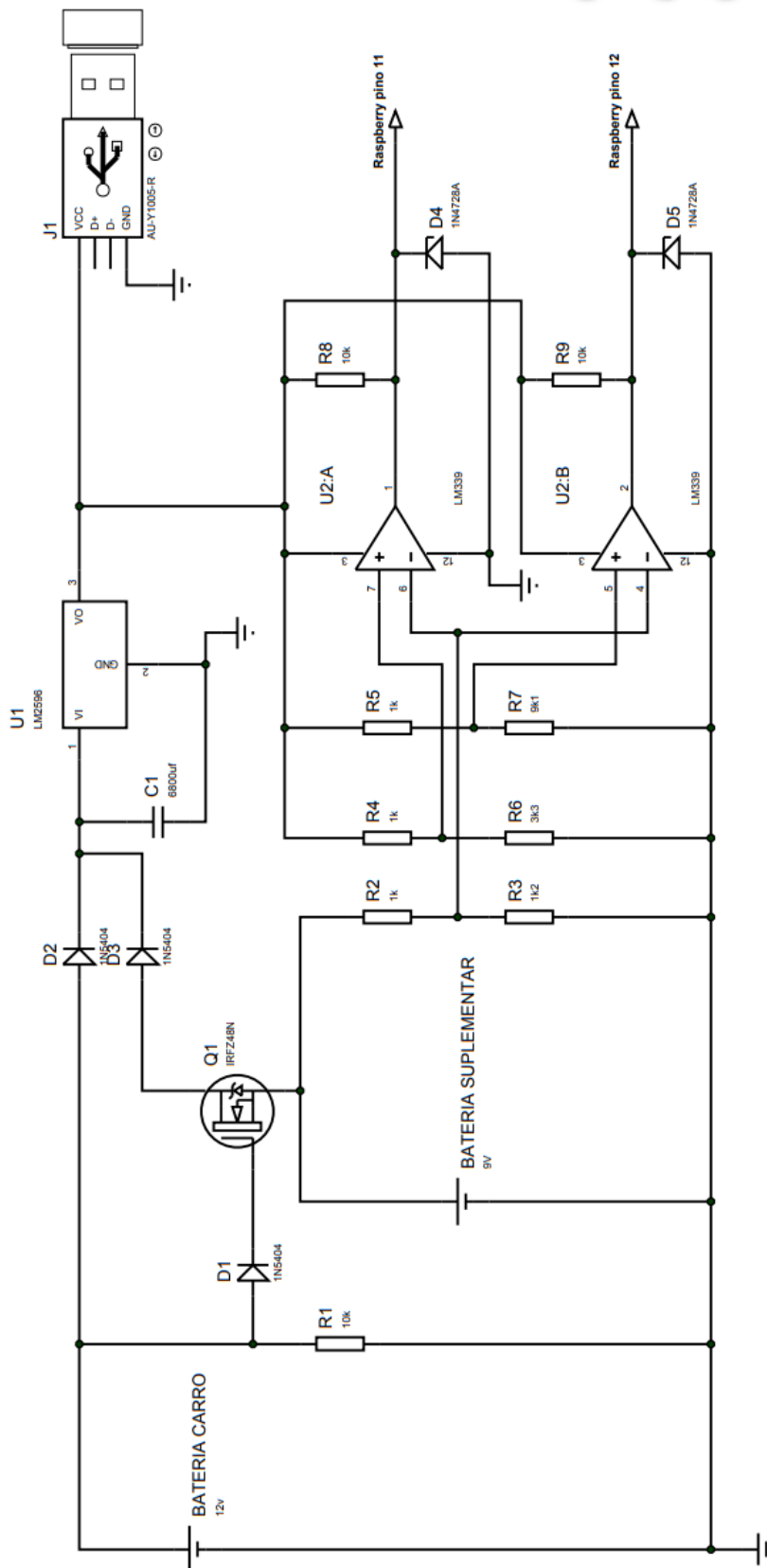
Fonte: O Autor

APÊNDICE C – CIRCUITO MEDIDOR DE BATERIA



Fonte: O Autor

APÊNDICE D - CIRCUITO COMPLETO DE ALIMENTAÇÃO DO SISTEMA



Fonte: O Autor

ANEXO A – CIRCUITO DO CONECTOR ELM327

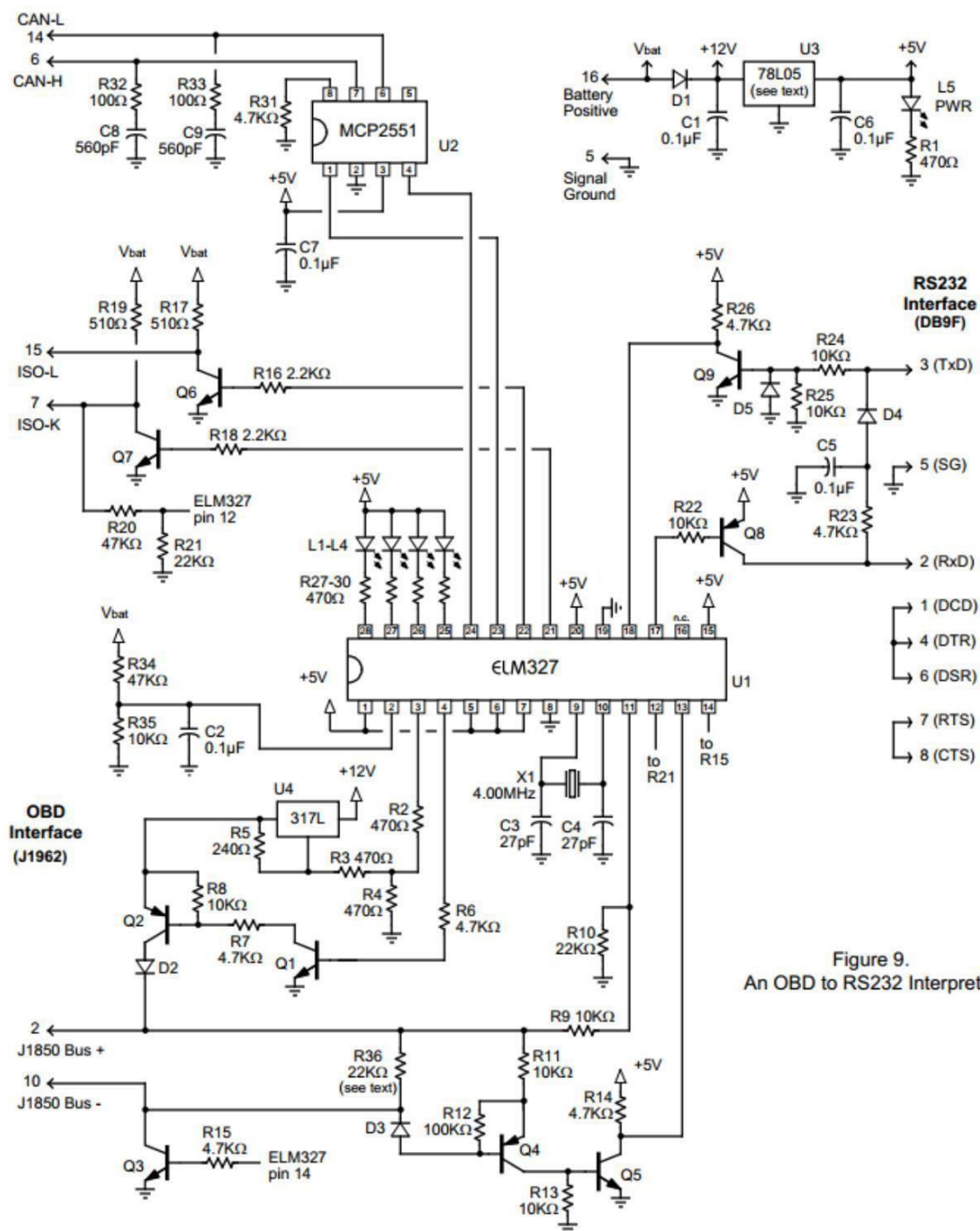


Figure 9.
An OBD to RS232 Interpreter

Fonte: ELM Electronics (2018)