

**Centro Universitário do Instituto Mauá de Tecnologia**

**Escola de Engenharia Mauá**

Engenharia Automotiva

Rogério Cassares Pires

**Estudo de viabilidade para desenvolvimento de aplicações automotivas em  
nuvem**

São Caetano do Sul

2019



Rogério Cassares Pires

**Estudo de viabilidade para desenvolvimento de aplicações automotivas em  
nuvem**

Especialização em Engenharia Automotiva, do Centro Universitário do Instituto Mauá de Tecnologia, como requisito ao desenvolvimento do trabalho de conclusão de curso.

Orientador: Valdir Melero Junior

Coorientador: Fernando de Almeida Martins

Área de concentração: Engenharia Automotiva

São Caetano do Sul  
2019

Pires, Rogério Cassares

Estudo de viabilidade para desenvolvimento de aplicações automotivas em nuvem / Rogério Cassares Pires – São Caetano do Sul, SP: CEUN-EEM, 2019.

44p.

Especialização em Engenharia Automotiva, do Centro Universitário do Instituto Mauá de Tecnologia, como requisito ao desenvolvimento do trabalho de conclusão de curso. — Centro Universitário do Instituto Mauá de Tecnologia, São Caetano do Sul, SP, 2019.

Orientador: Valdir Melero Junior. Coorientador: Fernando de Almeida Martins.

1. Viabilidade 2. Nuvem 3. Automotiva 4. Aquisição 5. Plataforma  
6. Dados. I. Pires, Rogério Cassares. II. Escola de Engenharia Mauá do Centro Universitário do Instituto Mauá de Tecnologia. III. Título.

*Dedico este trabalho à minha família, incluindo todos os irmãos da família celestial em Cristo Jesus. Dedico também a todos aos que lerem esta dedicatória para que se sintam amados pelo Amor Maior que primeiro nos amou quando nos formou para não sermos apenas criação, mas filhos de um bom Pai! Que este singelo trabalho possa ajuda-los e ampliar a visão de mundo de alguma forma, a saber, que importa enxergarmos além daquilo que os nossos olhos podem ver.*

## AGRADECIMENTOS

Agradeço primeiramente a Deus, o qual eu posso chama-lo por Aba, Pai! Porque precisamente durante o período cronológico deste curso Ele me fez enxergar aquilo que estava diante dos meus olhos o tempo inteiro e não via porque estava cego, porém agora vejo!; ao meu irmão mais velho Jesus Cristo por qual vivo por este Caminho, Verdade e Vida; ao Espírito Santo, pois habita em mim e, por habitar em mim, torna-me mais parecido com aquele de quem o enviou a mim, a consolar-me e a me ensinar das coisas e os propósitos de Deus, meu Pai.

Agradeço à minha família, especialmente à minha amada mamãe Wilma, ao meu pai da terra Mauro, aos meus irmãos Marcella, Gustavo, Maricota e Pedro. Agradeço também com alegria à irmã missionária Maria de Lourdes, por ter compartilhado comigo e nós ao Senhor Jesus pela escolha deste tema de trabalho e pelas infindáveis orações na madrugada por minha vida e pela vida de nossa família. Agradeço aos irmãos da Igreja Jesus Salvando Vidas, do *Dunamis Movement* e *Dunamis Pockets* por poder ser parte desta família e cumprir nosso chamado de pregar o *Ide* e viver pela graça do evangelho da Paz apregoado segundo a Palavra da Verdade, a Bíblia Sagrada (João 1).

Agradeço ao Instituto Mauá de Tecnologia pela formação em Engenharia de Controle e Automação. Em especial, ao reitor José Carlos de Souza Jr e ao Fernando de Almeida Martins e a todos os meus demais amigos pelos trabalhos que realizamos junto à Divisão de Eletrônica e Telecomunicações do Centro de Pesquisas do Instituto Mauá de Tecnologia.

Agradeço também à família *Maua Racing* pela oportunidade de conhecer e projetar, construir e competir ao lado de cada um, o veículo de Fórmula SAE, por cuja competição pude participar efetivamente desde 2014 até agora. Ao Willian Kurilov de Moraes, meu amigo, que coordena as atividades de Fórmula SAE e BAJA Mauá; e logicamente, também à equipe de Baja SAE (Obrigado também por ter a honra de fazer parte de tudo isso!).

Agradeço aos professores da graduação e pós graduação que me ajudaram a conhecer o que sei deste universo da engenharia que tanto amo. Obrigado também ao professor Valdir Melero Jr pela orientação do trabalho aqui escrito e pela coorientação do Fernando de Almeida Martins. Obrigado pela secretaria da pós graduação por aceitar as prorrogações e prazos e por compreender cada parte do processo de conclusão desta monografia.

*“O ESPÍRITO DO SENHOR Deus está sobre mim, porque o SENHOR me ungiu para pregar as boas novas aos pobres, enviou-me a curar os quebrantados de coração, a proclamar libertação aos cativos e a pôr em liberdade os algemados; a apregoar o ano aceitável do SENHOR e o dia da vingança do nosso Deus; a consolar todos os que choram”*

***Isaías 61:1-2***



## RESUMO

Neste trabalho foram estudadas as viabilidades técnicas e comerciais para a utilização de dados disponibilizados em módulos eletrônicos de controle embarcados em veículos automotores, através de protocolos de comunicação empregados na indústria automotiva, com a finalidade de gerar bancos de dados em nuvem, primordiais para a avaliação do desempenho e vida útil de parte do sistema embarcado de um veículo, disponibilizando tais dados em uma plataforma *online* para possíveis interpretações e análises.

Em parceria com a DET (Divisão de Eletrônica e Telecomunicações) do CP-IMT (Centro de Pesquisas do Instituto Mauá de Tecnologia), foi possível empregar a plataforma *Smart Campus*, a fim de desenvolver o mapeamento de uma rede de telecomunicações, desde o dispositivo embarcado, até a aplicação de usuário final.

Aplicando-se os estudos gerados a uma rede comercial, obteve-se um valor médio inicial de um dispositivo para um projeto para 50 pontos de medição. Também se obtiveram custos em relação à manutenção em uma plataforma comercial de rede LPWAN em grandes cidades brasileiras por dispositivo com envios de até 160 mensagens/mês ao servidor em nuvem, com pacotes de informações de 222 *bytes* de tamanho máximo cada uma, além dos cabeçalhos de cada pacote de dados.

Palavras-chave: Viabilidade. Nuvem. Automotiva. Aquisição. Plataforma. Dados.

## **ABSTRACT**

*In this work, the technical and commercial viability for the use of data made available in electronic control modules embedded in automotive vehicles, through communication protocols used in the automotive industry, was studied in order to generate cloud databases, which are fundamental to the evaluation of the performance and life of part of the embedded system of a vehicle, making such data available on an online platform for possible interpretations and analyzes.*

*In partnership with the DET (Electronics and Telecommunications Division) of the CP-IMT (Research Center of the Mauá Institute of Technology), it was possible to use the Smart Campus platform in order to develop the mapping of a telecommunications network, from the embedded device to the end-user application.*

*Applying the generated studies to a commercial network, an initial mean value of a device for a design for 50 measurement points was obtained. There were also costs related to maintenance on a commercial LPWAN network platform in major Brazilian cities per device with up to 160 messages / month uploads to the cloud server, with information packets of up to 222 bytes each, besides the headers of each data packet*

*Keywords: Viability. Cloud. Automotive. Acquisition. Platform. Data.*

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1 – Topologia da rede telecomunicações de uma rede LPWAN. ....  | 22 |
| Figura 2 – Dispositivo OBD comercial com pareamento <i>Bluetooth</i> para <i>smartphones</i> . ....                  | 24 |
| Figura 3 – Interface do aplicativo de <i>debug</i> implementado através da plataforma <i>App Inventor</i> .<br>..... | 28 |
| Figura 4 – Interface do aplicativo <i>Torque Pro</i> . ....  | 29 |
| Figura 5 – <i>Hardware</i> e <i>software</i> desenvolvidos pela DET do CP-IMT. ....                                  | 30 |
| Figura 6 – Aplicativo <i>debug</i> com resposta ao comando “AT DP”.....  | 33 |
| Figura 7 – Dados de resposta do veículo.....   | 34 |
| Figura 8 – Interface e medição de valores com o aplicativo <i>Torque Pro</i> .....                                   | 36 |
| Figura 9 – <i>Dashboard</i> para a demonstração de um aplicativo. ....   | 41 |

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1 – Descrição dos <i>modes</i> para comandos OBD segundo a norma SAE J1979.....             | 25 |
| Tabela 2 – Protocolos de comunicação disponíveis no <i>ELM327</i> . .....                          | 26 |
| Tabela 3 – Comandos OBD disponíveis para comunicação em um veículo segundo a norma SAE J1979. .... | 35 |
| Tabela 4 – Comparação entre os valores medidos e os calculados. ....                               | 37 |
| Tabela 5 – Lista de materiais e processos.....   | 38 |

## LISTA DE ABREVIATURAS E SIGLAS

- 2G – segunda geração para tecnologia de Internet sem fio de operadoras de telefonia móvel
- 3G – terceira geração para tecnologia de Internet sem fio de operadoras de telefonia móvel
- 4G – quarta geração para tecnologia de Internet sem fio de operadoras de telefonia móvel
- API – *Application Program Interface* (Interface de Programa de Aplicação)
- ASCII – *American Standard Code for Information Interchange* (Código Padrão Americano para o Intercâmbio de Informação)
- AT – *Attention Modem Commands* (Comandos de Atenção para Modems)
- CAN – *Controller Area Network* (Rede de Controle de Área)
- CP-IMT – Centro de Pesquisas do Instituto Mauá de Tecnologia
- DNS – *Domain Name Service* (Serviço de Nome de Domínio)
- DP – *Describe Protocol* (Descreva Protocolo – Comando OBD)
- DET – Divisão de Eletrônica e Telecomunicações
- ECU – *Electronic Control Unit* (Unidade de Controle Eletrônico)
- HTTP - *HyperText Trasfer Protocol* (Protocolo de Transferência de Hipertexto)
- IoT – *Internet of Things* (Internet das Coisas)
- LED – *Light Emission Diode* (Diodo Emissor de Luz)
- Li – Lítio (elemento químico utilizado para a fabricação de baterias)
- LoRa – *Low Power Long Range* (Baixa Potência Longo Alcance)
- LoRaWAN – *Low Power Long Range Wide Area Network* (Rede de Baixa Potência e Longo Alcance para Grandes Áreas)
- LoRa Alliance* – Organização responsável pelo Protocolo LoRaWAN
- LPWAN – *Low Power Wide Area Network* (Rede de Baixa Potência para Grandes Áreas)
- MQTT – *Message Queue Telemetry Transport* (Transporte de uma Fila de Mensagem por Telemetria)
- NPN - Denota um dispositivo semicondutor em que uma região do “tipo P”, está envolvida por ambas as extremidades com duas regiões do “tipo N”
- OBDII – *On board Diagnostic of the Second Generation* (Diagnóstico a Bordo de Segunda Geração)
- OTG – *On The Go* (tipo de conexão para placa de circuito impresso)
- PIC – *Program Interface Controller* (Controlador de Programa de Interface)
- PID – *Parameter Identifier* (Identificador de Parâmetros – Comando OBD)

PCB – *Printed Circuit Board* (Placa de Circuito Impresso)

PCI – Placa de Circuito Impresso

PSI – *Pounds per Square Inch* (Libras por Polegada Quadrada)

PWM – *Pulse Width Modulation* (Modulação por Largura de Pulso)

REST – *Representational State Transfer* (Transferência de Estado Representacional)

rpm – revoluções por minuto (unidade de rotação do motor de um veículo)

RS232 – *Recommended Standard number 232* (Padrão Recomendado número 232)

SMD – *Surface Mount Device* (Dispositivo Montável em Superfície)

SMT – *Surface Mount Technology* (Tecnologia de Montagem em Superfície)

SP – *Select Protocol* (Selecione Protocolo – Comando OBD)

UART - *Universal Asynchronous Receiver/Transmitter* (Receptor/Transmissor Universal Assíncrono)

UDP – *User Datagram Protocol* (Protocolo de Datagrama do Usuário)

USB – *Universal Serial Bus* (Comunicação Serial Universal)

VIN – *Vehicle Identification Number* (Número de Identificação de Veículo)

VPW – *Variable Pulse Width* (Variação de Largura de Pulso)

Wi-Fi – *Wireless Fidelity* (Fidelidade Sem Fio)

# SUMÁRIO

|            |   |           |
|------------|---|-----------|
| <b>1</b>   | <b>INTRODUÇÃO .....</b>   | <b>15</b> |
| 1.1        | JUSTIFICATIVA .....   | 17        |
| 1.2        | OBJETIVOS .....   | 17        |
| <b>2</b>   | <b>REVISÃO DE LITERATURA .....</b>  | <b>18</b> |
| 2.1        | LPWAN .....   | 18        |
| 2.2        | LORAWAN .....   | 19        |
| 2.2.1      | RÁDIO LoRA .....  | 19        |
| 2.2.2      | <i>END-NODE</i> .....   | 19        |
| 2.2.3      | <i>GATEWAY</i> .....  | 20        |
| 2.2.4      | <i>NETWORK SERVER</i> .....   | 20        |
| 2.2.5      | APLICAÇÃO .....   | 20        |
| <b>3</b>   | <b>MATERIAIS E MÉTODOS .....</b>  | <b>22</b> |
| 3.1        | TOPOLOGIA .....   | 22        |
| 3.2        | VEÍCULO DE ESTUDO .....   | 23        |
| 3.3        | DISPOSITIVO OBDII COMERCIAL .....   | 23        |
| 3.4        | COMANDOS OBD .....  | 24        |
| 3.5        | APLICATIVO PARA <i>DEBUG</i> .....  | 27        |
| 3.6        | APLICATIVO PARA VALIDAÇÃO .....   | 29        |
| 3.7        | PROPOSTA DE INCLUSÃO A UMA REDE <i>WIDE AREA NETWORK</i> .....                      | 29        |
| 3.8        | PROPOSTA PARA RECEBIMENTO DE DADOS E GERAÇÃO DE<br>RELATÓRIOS <i>ONLINE</i> .....   | 30        |
| 3.8.1      | REQUISIÇÃO DE DADOS <i>ONLINE</i> .....   | 31        |
| 3.8.2      | REQUISIÇÃO DE DADOS DE HISTÓRICO .....  | 31        |
| <b>3.9</b> | <b>VIABILIDADE COMERCIAL E TÉCNICA.....</b>   | <b>31</b> |
| 3.9.1      | DISPOSITIVO EMBARCADO .....   | 31        |
| 3.9.2      | REDE DE TELECOMUNICAÇÕES .....  | 31        |
| 3.9.3      | APLICAÇÃO INTERFACE USUÁRIO .....   | 32        |
| <b>4</b>   | <b>RESULTADOS E DISCUSSÃO.....</b>  | <b>33</b> |
| 4.1        | DETERMINAÇÃO E VALIDAÇÃO DE TECNOLOGIA DO DISPOSITIVO<br>ELETRÔNICO EMBARCADO ..... | 33        |

|            |   |           |
|------------|---|-----------|
| 4.1.1      | APLICATIVO DE <i>DEBUG</i> .....  | 33        |
| 4.1.2      | VALIDAÇÃO COM O APLICATIVO <i>TORQUE PRO</i> .....                                    | 36        |
| <b>4.2</b> | <b>INCLUSÃO E REQUISIÇÃO DOS DADOS COLETADOS A UMA REDE DE TELECOMUNICAÇÕES .....</b> | <b>37</b> |
| <b>4.3</b> | <b>VIABILIDADE COMERCIAL E TÉCNICA.....</b>   | <b>38</b> |
| 4.3.1      | DISPOSITIVO EMBARCADO .....   | 38        |
| 4.3.2      | REDE DE TELECOMUNICAÇÕES .....  | 40        |
| 4.3.3      | APLICAÇÃO INTERFACE USUÁRIO .....   | 40        |
| <b>5</b>   | <b>CONCLUSÃO .....</b>  | <b>43</b> |
| <b>6</b>   | <b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>   | <b>44</b> |



## 1 INTRODUÇÃO

Dispositivos denominados vestíveis (do inglês, *wearables*), geralmente são utilizados para monitorar e registrar funções vitais humanas, como batimento cardíaco, pressão e oxigenação sanguínea. Além disso, alguns desses aparelhos possuem capacidade de gerar relatórios de desempenho, mapas de deslocamento e gastos energéticos (STRATH e ROWLEY, 2018).

De maneira análoga, a aquisição de dados de veículos, através de dispositivo eletrônico embarcado, para monitorar índices de desgastes naturais, torna-se conveniente para uma manutenção preventiva e prescritiva mais precisa. Mais que isso, é necessário quantificar para qualificar como a relação com um usuário junto ao veículo pode ser não apenas confortável, mas possibilitar maior vida útil.

Não apenas coletar dados úteis, no que diz respeito ao desempenho da relação humana veicular, torna-se cada vez mais interessante, dentro de um ponto de vista comercial, determinar a forma de como analisar as informações obtidas com o auxílio de bases de dados externas, cruzar informações de canais de tempo *online*, sistemas de engenharia de tráfego e segurança pública.

É notório que o cruzamento desses dados não tem viabilidade comercial se não retornar informações pelas quais os interessados possam ser beneficiados. Desde a introdução do conceito de Internet das Coisas<sup>1</sup> (IoT), fala-se em conectar diversos aparelhos na Internet. Entretanto, com tantos dispositivos conectados à Internet nos dias de hoje, corre-se o risco de incluir tanto esforço e demanda de tempo e dinheiro na transmissão de um dado, que pouco se pensa no valor que uma aplicação pode retornar à comunidade.

No ecossistema da indústria automotiva e seus usuários - sejam montadoras, revendedoras, empresas de seguros, usuários de veículos automotores públicos ou privados - um índice importante de desgaste natural para previsão e prescrição de possível manutenção (entende-se

---

<sup>1</sup> Internet das Coisas: termo popularmente utilizado para demonstrar que um dispositivo está conectado e os seus dados estão sendo disponibilizados a quem também estiver conectado à Internet. Neste documento será referido pelo termo original IoT, do inglês, *Internet of Things*.

por manutenção não apenas o custo de troca de uma peça ou de um sistema, mas todo o tempo gasto e as consequências geradas em função da parada de um veículo que poderia ser prevista e, portanto, amenizadas suas consequências) é, por exemplo, o monitoramento e calibração de pneus, visto que o uso e manutenção incorreta levam a uma dirigibilidade ineficiente, insatisfatória, consumo excessivo de combustível, emissão excessiva de poluentes e desgaste emancipado de vida útil de um item com valor significativo em relação à manutenção de um veículo (GILLESPIE, 1992).

Considerando o conteúdo acima citado, se, no caso, um veículo não apenas possuir os seus dados coletados e enviados a uma plataforma em nuvem, mas integrados à sistemas de inteligência artificial, pode-se determinar, de acordo com alguns dados como da aceleração, velocidade e frenagem, um perfil de agressividade do motorista. Ao serem tratados os dados de pressão de pneus e consumo de combustível, por exemplo, pode-se prever o desgaste desnecessário em que uma medida preventiva de calibração adequada de pressão dos pneus de um veículo evitaria o desgaste natural prematuro dos pneus e traria economia de combustível, ao reduzir o atrito da superfície do pneu com a banda de rodagem de uma via. Outro exemplo é que, ao serem informados códigos de erros por uma ECU (*Electronic Control Unit*) do veículo, poderia ser solicitado um guincho antes que o veículo sofresse uma pane, deixando o usuário em situações de apuros, dependendo das circunstâncias. Ainda, através de acionamentos de botão de emergência, é possível identificar o local de uma ocorrência, sem que o motorista se distraia efetuando ligações telefônicas ao dirigir.

Com exceção do parâmetro de localização citado no parágrafo anterior, a grande maioria dos dados requeridos por cada aplicação está disponível em cada veículo comercializado atualmente, porém estes dados não são levados a um patamar em que sejam analisados e inferidos a gerenciar soluções preventivas e prescritivas para problemas que ainda não se tornaram críticos e/ou de difícil identificação pelo usuário do veículo.

Este trabalho, por meio de um estudo, permitiu promover a aquisição de informações para viabilizar os aspectos comerciais e técnicos a fim de disponibilizar uma nuvem de dados obtidos de veículos em um sistema capaz de garantir a escalabilidade horizontal (capacidade de multiplicar-se um número de dispositivos por aplicações, modularidade), bem como garantir a escalabilidade vertical (atingir outras aplicações e capacidade de aumentar um número maior de aplicações).

Não se restringindo à conectividade, é importante que se pense na solução agnóstica à tecnologia. Embora a tecnologia utilizada neste trabalho aborda acerca de *LPWANs* (*Low Power Wide Area Network*), pois não há necessidade de uma rede de dados com alta taxa de envio e recebimento de dados em cada dispositivo para se garantir informações vitais. O conceito principal é: a partir de dados enviados ciclicamente no tempo, é possível obter informações organizadas das principais atividades de um veículo automotor e códigos de erros para manutenções adequadas, com menor tempo e gerenciamento inteligente.

## **1.1 JUSTIFICATIVA**

Um veículo possui sensores em que são medidos e controlados pelas diversas ECUs. Entretanto, como estas informações seguem um padrão, isto é, um protocolo de comunicação (ISO, 2016), é possível fazer a aquisição destes dados e transmiti-los a uma plataforma de desenvolvimento de aplicações automotivas em nuvem. A justificativa deste projeto dá-se pela possibilidade de gerenciar e inferir sobre gestão e manutenção de veículos de frota e/ou particulares, de qualquer marca e modelo, considerando os protocolos de comunicações por padrão na indústria, para reduzir custos de manutenções corretivas, auxiliando em medidas para se obter o menor dano ao veículo em termos de desgaste natural, bem como manter o usuário da aplicação seguro à falhas eletromecânicas acumulativas imperceptíveis ao cotidiano mas que podem gerar uma degradação de seu patrimônio.

## **1.2 OBJETIVOS**

Foi estudada a viabilidade e realizada uma proposta para desenvolvimento de uma aplicação para gerar uma aquisição de dados de sensores de veículos, transmitindo e disponibilizando-os a uma plataforma em nuvem, de desenvolvimento de aplicações de gerenciamento e manutenções automotivas, utilizando-se de uma rede certificada de baixo custo, escalável e de longo alcance.

Por fim, mapeou-se a cadeia de telecomunicações, a partir do levantamento da viabilidade comercial e tecnológica, excluindo-se do objetivo as implementações eletrônicas.

## 2 REVISÃO DE LITERATURA

### 2.1 LPWAN

O termo LPWAN não se traduz em uma tecnologia específica. É um conceito de topologia muito utilizado em IoT. Basicamente, há quatro blocos que se interligam de maneira a gerar um caminho para a informação do sensor à aplicação - o *uplink* (envio de um dado do dispositivo embarcado ao servidor) - e da aplicação ao sensor - o *downlink* (envio de um dado do servidor ao dispositivo embarcado). Dependendo da tecnologia de rádio frequência a ser utilizada para transmitir a informação do sensor, utilizam-se protocolos de comunicação diferentes, entretanto, a topologia se mantém muito similar entre tais opções.

Com o intuito de se obter baixo consumo energético e, portanto, vida útil de alta autonomia, otimizando o alcance dos dados, segundo as características de uma rede *Low Power*, de forma a garantir a escalabilidade, isto é, possibilidade de crescimento em termos de números de sensores, um sensor equipado com o dispositivo de rádio frequência não se comunica diretamente com o outro, logo, o acréscimo de sensores ao sistema resulta no aumento da complexidade para o tratamento de tais informações.

O comportamento e a topologia da rede de telecomunicação utilizada para o gerenciamento dos sensores obedecem à topologia estrela, em que cada dispositivo se comunica com uma antena conectada a um computador denominado *gateway* (vide item 2.2.3), através de um sinal de rádio frequência do tipo LoRa (*Low Power Long Range Wide Area Network*). Esta antena comunica-se com o servidor por uma conexão à Internet uma vez que o servidor, denominado *Network Server*, está instanciado em plataforma de nuvem, isto é, uma máquina virtual em um servidor principal com acesso através de um *DNS* (*Domain Name Service*). Portanto, o *gateway* é o responsável por receber a informação dos sensores equipados com o dispositivo de rádio frequência e transmitir estes dados ao servidor (*uplink*). De semelhante maneira, também podem ser enviados dados do servidor ao sensor através do *gateway* (*downlink*).

Conforme DET-CP-IMT (2018), um servidor característico de uma rede LPWAN, deve exercer o gerenciamento de dispositivos cadastrados, executar a ativação para o início de recebimento dos dados e também gerenciar as características de recebimento e envio de dados aos dispositivos embarcados com o rádio.

## 2.2 LORAWAN

Um protocolo amplamente utilizado pela comunidade focada em IoT é o protocolo LoRaWAN (*Low Power Long Range Wide Area Network*). Este protocolo bem definido pela organização *LoRa Alliance* (ALLIANCE, 2017), possui como comportamento baixo consumo pelo tráfego de pequenos pacotes de dados intermitentes, visando longo alcance em termos de distância e autonomia da vida útil de bateria, enquadrando-se nas características de uma rede *Low Power Wide Area Network*, que nada mais é que uma generalização às diversas tecnologias e protocolos que possuem a baixa potência energética de um dispositivo como objetivo principal.

### 2.2.1 RÁDIO LORA

O rádio LoRa, assim como outras tecnologias voltadas para IoT, possui características de rádio com frequência de sub-giga, ou seja, abaixo da ordem de grandeza de GHz (giga Hertz). A tecnologia LoRa opera em várias faixas de frequências, dependendo da região, variando de país para país, segundo as normas de certificação interna para dispositivos equipados com rádio frequência (ALLIANCE, 2017).

O rádio LoRa é apenas um meio empregado para promover a comunicação entre um sensor e um dispositivo que recebe sinais segundo as definições descritas neste padrão. É possível a comunicação direta entre dispositivos equipados com rádios LoRa, não caracterizando o protocolo LoRaWAN, portanto esta caracterização não será abordada nesta documentação.

#### 2.2.1.1 Legislação

O ATO N° 14448, de 4 de dezembro de 2017, garante que, para a tecnologia de rádio frequência nos padrões de utilização de rádios LoRa no Brasil, os dispositivos embarcados devem ser certificados segundo a norma descrita em tal documento (GOES e MENEZES, 2018).

### 2.2.2 END-NODE

Um *end-node* é um dispositivo eletrônico embarcado caracterizado por um ou mais sensores, podendo ter atuadores, ou não, em que, a partir da grandeza mensurada, são enviados pacotes de informações ciclicamente no tempo através de um rádio LoRa junto a este dispositivo.

### 2.2.3 GATEWAY

*Gateway* trata-se de uma antena conectada a um computador com conexão à Internet. As informações provenientes do dispositivo embarcado são capturadas e enviadas, segundo o protocolo definido a um servidor em nuvem, bastando-lhe apenas um ponto de conexão à Internet e energia elétrica disponível para que este dispositivo funcione.

### 2.2.4 NETWORK SERVER

O *Network Server* é o servidor que gerencia as informações de usuários, ativação, envio e recebimento de dados; guarda as informações e disponibiliza-as às aplicações existentes.

### 2.2.5 APLICAÇÃO

A partir de um dado disponível do sensor por um servidor, é possível, por exemplo, executar as aplicações a nível de supervisor, e a partir de um certo número de dados relevantes, aplicar conhecimentos de *machine learning*<sup>2</sup> para que a aplicação desenvolva estratégias para definir um comportamento dos dados ao longo do tempo, gerando informações de comportamento dos sensores e atuando de forma inteligente na prevenção prescrição e/ou correção do sistema aferido.

É importante notar que ao se utilizar de uma rede LPWAN, não é ideal estabelecer uma alta taxa de amostragem e de *baud-rate*, visto que não é voltado para este fim. Áudios, fotos, vídeos e longas mensagens não são devem ser enviados por possuírem uma quantidade de *bytes* muito maior do que a ser enviada por cada *uplink* do *end-node* à aplicação, segundo o protocolo LoRaWAN.

Também de acordo com as características deste protocolo em território brasileiro, apenas de 51 a 222 *bytes* estão disponíveis para cada *uplink*, visto que o máximo intervalo de tempo em que uma mensagem pode trafegar no ar é de 400 milissegundos (ALLIANCE, 2017).

---

<sup>2</sup> *Machine-learning*: termo utilizado para demonstrar a capacidade de um algoritmo desenvolver estratégias para resolver uma situação de maneira autônoma, geralmente sem a necessidade de intervenção humana no processo.

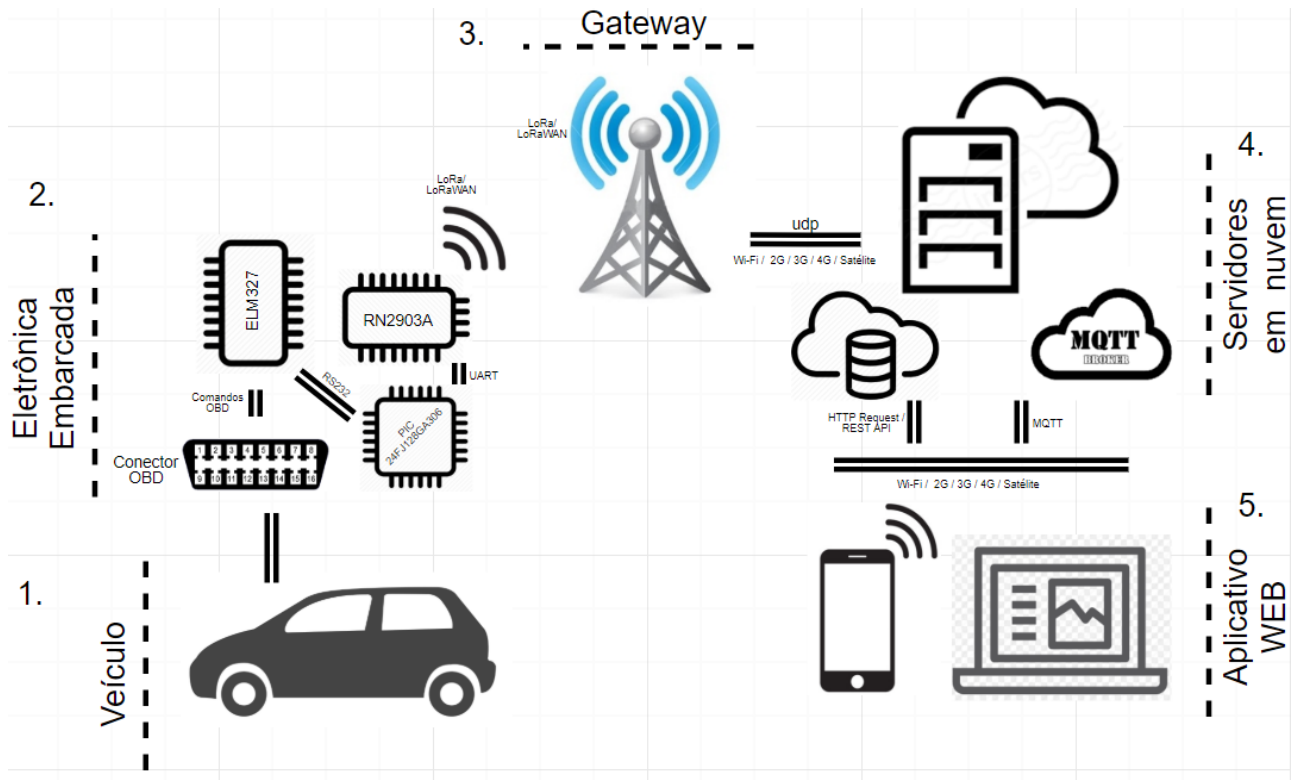
Deve se levar em consideração também que, quanto maior o pacote de informação, menor distância de transmissão via rádio frequência será alcançada com uma informação válida à aplicação. Por isso, não se trata apenas de transmitir todas as informações disponíveis no momento, mas aquelas primordiais e, com isso, trabalhar no nível da aplicação com o menor número de informações, para alcançar, através da inteligência de dados, informações que inferem para o “Comportamento das Coisas”, isto é, um padrão de comportamento de um dispositivo equipado com o sensor ao longo do tempo.

### 3 MATERIAIS E MÉTODOS

#### 3.1 TOPOLOGIA

Conforme a topologia descrita na Figura 1, descreveu-se como foi estabelecida a rede de telecomunicações entre o medido no veículo até o ponto em que se pôde obter o dado em uma aplicação WEB, isto é, uma aplicação conectada à Internet.

Figura 1 – Topologia da rede telecomunicações de uma rede LPWAN.



Fonte: O Autor.

De acordo com a topologia proposta, pode-se verificar que:

1. o dado veicular é obtido através de um circuito integrado *ELM327* (cuja função é obter dados e detectar códigos de erro de um veículo). Quando requisitado, através de uma comunicação serial, segundo o protocolo RS232 (*Recommended Standard number 232*), por um outro dispositivo, no caso foi empregado o microcontrolador *PIC24FJ128GA306* (da Microchip), o *ELM327* requisita uma informação ao veículo,



- através do protocolo correto de comunicação definido pela fabricante e de acordo com normas especificadas (vide item 3.4);
2. o microcontrolador é responsável por obter os dados, gerenciá-los e disponibilizá-los a um circuito integrado *RN2903A*. Este dispositivo, também microcontrolado, é equipado com um rádio frequência, isto é, através de sinais de rádio transmite uma informação através de um meio, no caso, este meio é o ar. Ao receber do microcontrolador *PIC*, via protocolo UART (*Universal Asynchronous Receiver/Transmitter*) um comando, envia, segundo o protocolo LoRaWAN, uma mensagem de rádio frequência a uma estação base em que se encontra um *gateway* LoRa, também segundo o protocolo LoRaWAN;
  3. o *gateway*, por sua vez, comunica-se com o servidor da plataforma *Smart Campus* do Instituto Mauá de Tecnologia através de uma conexão à Internet segundo o protocolo UDP (*User Datagram Protocol*, do inglês, Protocolo de Datagrama do Usuário);
  4. caso o servidor certifique-se de que o envio do dado foi relacionado a um dispositivo previamente cadastrado no sistema, ao receber esta informação, ele a insere em um banco de dados, de maneira criptografada, segundo o protocolo LoRaWAN, ao mesmo tempo em que publica este dado em um tópico segundo o protocolo MQTT (*Message Queue Telemetry Transport*);
  5. finalmente, a aplicação se conecta ao servidor de duas formas: *i.* através de uma requisição HTTP (*HyperText Transfer Protocol*) para reaver os dados de histórico; *ii.* através de uma inscrição MQTT, em que, de maneira *online*, inscreve-se no tópico da mensagem, recebendo o dado no momento enviado pelo dispositivo de rádio frequência LoRa. O dado é recebido pela aplicação através de uma interface WEB de qualquer dispositivo com navegador conectado à Internet.

### 3.2 VEÍCULO DE ESTUDO

O veículo de estudo para este trabalho foi, segundo escolha arbitrária e disponível, um Volkswagen Up! TSI Ano de fabricação/Ano modelo 2015/2016.

### 3.3 DISPOSITIVO OBDII COMERCIAL

O dispositivo OBD (*On Board Diagnostic*) utilizado foi o *OBD2 Wireless Bluetooth Fault Code Live Data Reader V1.5*, da marca VGate, conforme Figura 2. Este dispositivo é capaz de parear através do protocolo *Bluetooth* com um *smartphone* por meio de um aplicativo.

Conforme descrito no item 3.1, os comandos recebidos pelo dispositivo OBD através da comunicação *Bluetooth* via aplicativo em *smartphones*, são enviados ao circuito integrado *ELM327* através do protocolo *RS232*, então, do *ELM327* ao veículo (vide item 3.4). A partir daí, são requisitadas respostas dos módulos de controle eletrônico (ECUs) do veículo pelo *ELM327*. Uma vez obtida a resposta, ela é encaminhada ao dispositivo *Bluetooth* e este, por sua vez, envia ao aplicativo no *smartphone*, que mostra ao usuário a informação.

Figura 2 – Dispositivo OBD comercial com pareamento *Bluetooth* para *smartphones*.



Fonte: O Autor.

### 3.4 COMANDOS OBD

Comandos OBD são verificados pelo *ELM327* e enviados ao veículo para a obtenção da resposta, segundo a requisição esperada. De acordo com o manual de utilização do circuito integrado *ELM327* (ELMELECTRONICS, 2011), se os *bytes* enviados para o circuito integrado *ELM327*, segundo o protocolo *RS232*, não começarem com os caracteres "A" ou "T", então assume-se que são comandos OBD os que são enviados ao veículo pelo *ELM327*. Cada par de *bytes* do tipo ASCII<sup>3</sup> (*American Standard Code for Information Interchange*) é testado para

---

<sup>3</sup> ASCII: *American Standard Code for Information Interchange*, do inglês, Código Padrão Norte-Americano para Intercâmbio de Informações). Segundo EDUARDO KARASINSKI, 2009, o código visou a padronização de códigos para caracteres alfa numéricos (letras, sinais, números e acentos). Possibilitando que computadores de diferentes fabricantes conseguissem entender os códigos descritos segundo este padrão.

certificar-se de que sejam valores válidos em hexadecimal e então possam ser transmitidos ao veículo.

Dígitos hexadecimais são utilizados para todas as trocas de dados com o *ELM327*, pois são os formatos mais utilizados pelas normas OBD. Eventualmente, os dados recebidos do veículo através do *ELM327* devem ser manipulados dependendo da requisição efetuada.

As normas requerem que, para comandos OBD ou requisições que são enviadas ao veículo, deve-se obedecer a um formato. O primeiro *byte* enviado, conhecido como *mode*, descreve o tipo de dado a ser requisitado. Os *bytes* seguintes ao *mode* são conhecidos como PID (*Parameter Identification*). Os *modes* e os PIDs são descritos em detalhes segundo as normas SAE J1979 e ISO 15031-5 (INTERNATIONAL, 2017), outrossim também definidos por cada fabricante. Ainda, segundo a norma SAE J1979 (INTERNATIONAL, 2017), são definidos dez possíveis modos de testes, dentre os quais estão segundo a Tabela 1.

Tabela 1 – Descrição dos *modes* para comandos OBD segundo a norma SAE J1979.

| <b>Mode</b> | <b>Descrição</b>                                   |
|-------------|--|
| 01          | mostrar dados atuais                               |
| 02          | mostrar conjunto de dados congelados               |
| 03          | mostrar diagnósticos de códigos de erro            |
| 04          | limpar códigos de erro e valores armazenados       |
| 05          | resultado de testes, sensores de oxigênio          |
| 06          | resultados de teste, não continuamente monitorados |
| 07          | mostrar códigos de erro em “pendência”             |
| 08          | modo de controle especial                          |
| 09          | requisitar informação do veículo                   |
| 0A          | requisitar códigos de erro permanentes             |

Fonte: (ELMELECTRONICS, 2011).

Dentre os modos acima listados, são os de interesse para a aplicação proposta no trabalho, dados referentes aos sensores que são utilizados pelo veículo para que esteja em estado normal de funcionamento (*mode* “01”), para que se conheça o perfil de utilização de um veículo, por exemplo, requisitar diagnósticos de códigos de erros (*mode* “03”), bem como requisitar informações do veículo (*mode* “09”). Os demais *modes* não necessitam ser implementados no dispositivo eletrônico embarcado, pois não há, a priori, necessidade de transmiti-los como informação de vida útil e/ou manutenção preventiva e prescritiva do veículo para a plataforma

em nuvem, visto que não se tratam de dados que indicam funcionamento e/ou referentes aos estudos de uma dinâmica veicular.

Para que cada comando citado acima funcionasse corretamente, foi preciso definir um protocolo de comunicação entre o circuito integrado *ELM327* e o veículo. Dentre os protocolos possíveis de comunicação (Tabela 2), o *ELM327*, identifica e estabelece automaticamente a comunicação de protocolo correta ao ser enviado o comando “AT SP0” a ele. Segundo a Tabela 2 a seguir, há doze protocolos que podem ser definidos manualmente, porém o mais indicado é utilizar a identificação automática de protocolos. Deste modo, o dispositivo eletrônico pode ser utilizado para a maioria dos veículos, sem a necessidade de um protocolo em particular. Esta propriedade de detecção automática de protocolo garante a escalabilidade da aplicação para qualquer veículo, mesmo que não seja o estudado neste documento (item 3.2).

Tabela 2 – Protocolos de comunicação disponíveis no *ELM327*.

| ID | Protocolo Selecionado                                 |
|----|---|
| 0  | Automático  |
| 1  | SAE J150 PWM (41,6 <i>kbaud</i> )                     |
| 2  | SAE J150 VPW (10,4 <i>kbaud</i> )                     |
| 3  | ISO 9141-2 (5 <i>baud init</i> , 10,4 <i>kbaud</i> )  |
| 4  | ISO 14230-4 (5 <i>baud init</i> , 10,4 <i>kbaud</i> ) |
| 5  | ISO 14230-4 ( <i>fast init</i> , 10,4 <i>kbaud</i> )  |
| 6  | ISO 15765-4 (11 <i>bit ID</i> , 500 <i>kbaud</i> )    |
| 7  | ISO 15765-4 (29 <i>bit ID</i> , 500 <i>kbaud</i> )    |
| 8  | ISO 15765-4 (11 <i>bit ID</i> , 250 <i>kbaud</i> )    |
| 9  | ISO 15765-4 (29 <i>bit ID</i> , 250 <i>kbaud</i> )    |
| A  | SAE J1939 CAN (29 <i>bit ID</i> , 250* <i>kbaud</i> ) |
| B  | USER 1 CAN (11* <i>bit ID</i> , 125* <i>kbaud</i> )   |
| C  | USER 2 CAN (11* <i>bit ID</i> , 50* <i>kbaud</i> )    |

Fonte: (ELMELECTRONICS, 2011).

Iniciada a comunicação com o protocolo devido e enviado um comando OBD ao veículo para a verificação de parâmetro como a temperatura do líquido de arrefecimento do motor, cujo *mode* é “01” e o PID é “05” (ambos em base hexadecimal), envia-se “01 05” ao *ELM327* e é retornado um valor hexadecimal “41 05 5D”. O primeiro *byte* da resposta (“41”) refere-se à resposta a uma requisição para o *mode* “01” (40 + 01), enquanto o segundo *byte* (“05”) é a repetição do comando referente à temperatura do líquido de arrefecimento do motor. Por fim,

o terceiro *byte* (“5D”) significa o valor medido (em °C). No entanto, tal informação ainda deve ser convertida de hexadecimal para decimal e subtraído o valor de 40. Segundo INTERNATIONAL (2017), por padrão, a indústria automotiva utiliza de temperaturas de até 40 °C negativos. Para evitar o cálculo de valores hexadecimais com operador negativo, utilizou-se a técnica de deslocamento de unidades, isto é, um *offset* para que 40 °C corresponda ao *byte* hexadecimal “00”. Nota-se que o espaço entre os *bytes* a serem enviados é facultativo. Por exemplo, o comando “41 05 5D” é compatível com o comando “41055D”.

Um segundo exemplo de comando OBD que pode ser realizado é a requisição de um parâmetro atual (*mode* “01”) para obter a rotação do virabrequim do motor à combustão interna, ou seja, a quantidade de revoluções por minuto (PID “0C”). Ao enviar o comando OBD “01 0C” para o *ELM327* tem-se como resposta “41 0C 1A F8”. O valor hexadecimal “41” é a resposta ao comando do *mode* “01”, o *byte* “0C” é a repetição do parâmetro PID, e os próximos dois *bytes* “1A F8” referem-se ao valor em hexadecimal que deve ser convertido para decimal e dividido por 4, visto que esta medição rotação do motor incrementa um valor inteiro a cada um quarto de volta do virabrequim.

Por fim, outro exemplo de comando OBD para requisitar a informação do número de identificação do veículo (VIN – *Vehicle Identification Number*) com o PID “02”, pode ser realizado através do *mode* “09” (requisitar informação do veículo, Tabela 1). Este comando OBD retorna um valor em múltiplas linhas. Ao ser enviado para o *ELM327* um comando “09 02”, por exemplo, obtém-se um valor do tipo “49 02 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17”. O *byte* “49” é o retorno ao comando *mode* “09”. O *byte* em seguida (“02”) é a repetição do parâmetro PID enviado. Os próximos 17 *bytes* devem ser traduzidos segundo tabela ASCII, para formar o número de identificação do veículo.

### 3.5 APLICATIVO PARA *DEBUG*

De acordo com os comandos descritos no item 3.4, foi implementado um aplicativo para sistemas *Android* que envia e recebe comandos ao dispositivo OBD comercial (item 3.3). Ao receber as informações referentes aos comandos enviados, o aplicativo mostra em uma área de texto a resposta dada pelo dispositivo OBD veicular.

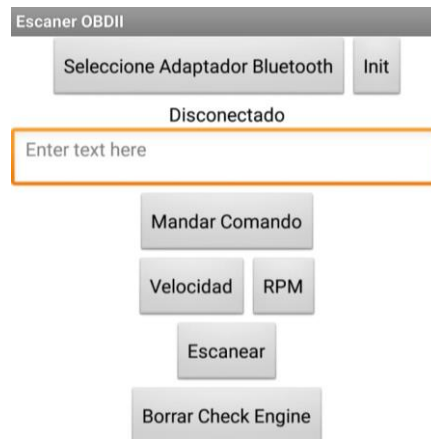
Este *software* está disponível em um ambiente de desenvolvimento para aplicativos *Android*, baseado na plataforma *App Inventor*, do *Massachusetts Institute of Technology*, com o nome de OBDII (BENTLEY, 2015).

Após ter sido realizada a instalação do aplicativo no *smartphone*, deve-se pará-lo com o dispositivo OBD comercial, seguindo-se as recomendações do fabricante de tal dispositivo.

Selecionando-se no aplicativo o dispositivo correto a ser conectado após o pareamento *Bluetooth*, o sistema executa um comando de “*RESET*” do circuito integrado *ELM327*, através do comando “*AT Z*”. A seguir, procura automaticamente pelo protocolo correto para conectar-se com o veículo através do comando “*SP 0*”. A partir daí, está pronta a comunicação entre *smartphone*, o *ELM327* e o veículo.

Conforme descrito acima, a Figura 3 ilustra que ao ser enviado um comando no espaço acima do botão “Mandar Comando”, este comando é recebido pelo *ELM327*, que verifica a integridade da mensagem e a executa segundo o protocolo correto para cada veículo. Feito isso, a resposta é transmitida do veículo ao *ELM327*, repassada ao aplicativo e então exibido o retorno da mensagem na área a ele especificado, como pode ser visto no item 4.1.1.

Figura 3 – Interface do aplicativo de *debug* implementado através da plataforma *App Inventor*.



Fonte: O Autor.

### 3.6 APLICATIVO PARA VALIDAÇÃO

Para validar a ferramenta de *debug* implementada, foi executado um aplicativo de mercado para *Android*, denominado *Torque Pro*, cuja interface pode ser vista na Figura 4. Este aplicativo utiliza uma conexão via *Bluetooth* para receber os dados do dispositivo OBD comercial descrito no item 3.3. Neste caso, são comparadas as respostas por ambas aquisições para a validação dos resultados obtidos manualmente.

Figura 4 – Interface do aplicativo *Torque Pro*.



Fonte: O Autor.

Uma vez os dados obtidos manualmente, é possível implementá-los no microcontrolador *PIC* empregado neste trabalho (item 3.7).

### 3.7 PROPOSTA DE INCLUSÃO A UMA REDE *WIDE AREA NETWORK*

Das informações coletadas, é possível incluí-las em uma plataforma de dados para se disponibilizar em nuvem (Internet). Para se viabilizar tecnicamente, a plataforma escolhida para a inclusão de dados em nuvem foi a *Smart Campus*, desenvolvida pela DET do CP-IMT, oferecendo acesso gratuito aos alunos que a utilizam, como parte de seus desenvolvimentos com a Instituição. Mais detalhes de viabilidade técnica e comercial estão descritos no 4.3.

Substituindo-se o dispositivo OBD comercial (item 3.3) e uma vez conectado com o *hardware* proposto (Figura 5) da plataforma *Smart Campus* (DET-CP-IMT, 2018), é possível a

comunicação do circuito integrado *ELM327* diretamente com o *PIC*, através de comunicação serial definida pelo protocolo RS232 (3.1).

Figura 5 – *Hardware e software* desenvolvidos pela DET do CP-IMT.



Fonte: Disponível em ([https://smartcampus.maua.br/?page\\_id=211](https://smartcampus.maua.br/?page_id=211)).

O microcontrolador *PIC*, ao receber os dados do circuito integrado *ELM327*, é capaz de armazenar e enviar os dados a um outro circuito integrado, através de uma comunicação serial UART, o *RN2903A*. Este circuito (*RN2903A*) é uma placa de *rádio* frequência, construída de acordo com o protocolo LoRaWAN, onde tal protocolo trata-se de uma rede do tipo LPWAN, o qual permite enviar dados por longas distâncias, com baixo consumo energético e baixo custo em relação à rede 2G, 3G, 4G e satélites. Esta viabilidade será evidenciada no item 4.3.2.

### **3.8 PROPOSTA PARA RECEBIMENTO DE DADOS E GERAÇÃO DE RELATÓRIOS *ONLINE***

No evento de um dado recebido através da plataforma *Smart Campus*, este é disponibilizado para o usuário da aplicação correspondente.

Há duas maneiras bem definidas de se receber os dados de uma aplicação. Conforme documentação de aplicação descrita no *website Smart Campus* (DET-CP-IMT, 2018), é possível exibir um dado de maneira *online*, bem como reaver os dados de histórico para cada dispositivo embarcado ao veículo, separadamente. Os detalhes de como reaver esses dados não estão no escopo deste documento.



### **3.8.1 REQUISIÇÃO DE DADOS *ONLINE***

Chegando um dado ao servidor da plataforma *Smart Campus*, o mesmo é publicado em um tópico, segundo o protocolo MQTT. Ao conectar-se neste mesmo tópico, a aplicação recebe e/ou envia o dado publicado no mesmo instante, tornando a visualização de dados de maneira *online* (item 4.3.3).

### **3.8.2 REQUISIÇÃO DE DADOS DE HISTÓRICO**

Para a formação de relatórios e gráficos temporais de parâmetros desejados, é possível realizar uma requisição HTTP do tipo GET. Esta requisição, em verdade, trata-se de uma consulta indireta realizada por uma API (*Application Program Interface*) REST (*Representational State Transfer*) ao banco de dados, criptografado, da plataforma *Smart Campus*. Requisitando, através deste comando GET os parâmetros de: *i. appSkey* (*Application Session Key*), que é a senha cadastrada no dispositivo embarcado LoRaWAN e no servidor da plataforma *Smart Campus* para descriptografia dos dados); *ii. device\_EUI* (*End Device Unique Identifier*); *iii.* bem como o número de resultados em ordem decrescente; é possível reaver estes dados para se produzir os relatórios conforme desejado (item 4.3.3).

## **3.9 VIABILIDADE COMERCIAL E TÉCNICA**

### **3.9.1 DISPOSITIVO EMBARCADO**

Descartando-se os componentes discretos, como indutores, capacitores, resistores e reguladores, apenas incluindo-se os componentes mais significativos, quanto aos seus valores de compra, destacam-se: o circuito integrado *ELM327*, o microcontrolador *PIC24FJ128GA306*, o módulo LoRa *RN2903A*, o *chip* antena para a frequência desejada, o conector OBD e a placa de circuito impresso. Os valores referentes a cada componente aqui descrito estão evidenciados na Tabela 5, do item 4.3.1.

### **3.9.2 REDE DE TELECOMUNICAÇÕES**

Há diversas redes de comunicação e protocolos para disponibilizar um dado em nuvem, seja este qual for. Conforme item 3.7, foi escolhido o protocolo LoRaWAN, efetuando-se os

primeiros testes em parceria com a DET do CP-IMT. Entretanto, para aplicação comercial, há redes LPWANs gerenciadas por empresas de cobertura em grandes cidades que oferecem a disponibilidade dos dados capturados por protocolos também LoRaWAN, permitindo que a aplicação seja escalável em termos de possibilidade de aumento de quantidade de dispositivos, quanto incluir outros serviços na aplicação final ao usuário.

### **3.9.3 APLICAÇÃO INTERFACE USUÁRIO**

Finalmente, para que se atingisse a proposta de experiência de usuário que utiliza os valores gerados pelas informações, foi proposto uma aplicação WEB em uma plataforma denominada *Node-RED*. Trata-se de uma plataforma gratuita para diversos sistemas operacionais, e possibilita ser executada em um servidor WEB, isto é, um serviço em nuvem que disponibiliza uma interface a qualquer dispositivo conectado à Internet. Esta ferramenta gratuita e de código aberto (*open source*), foi escolhida para que os primeiros protótipos possam ser rapidamente escaláveis, independentes da plataforma utilizada por cada dispositivo, uma vez que é possível o acesso através de um navegador na Internet e não restrita apenas a um aplicativo nativo em *smartphones*.

## 4 RESULTADOS E DISCUSSÃO

### 4.1 DETERMINAÇÃO E VALIDAÇÃO DE TECNOLOGIA DO DISPOSITIVO ELETRÔNICO EMBARCADO

De acordo com o dispositivo OBD comercial citado no item 3.3, e operando segundo as normas indicadas no item 3.4, implementou-se um aplicativo OBD para *debug*, conforme destacado no item 3.5. Pareando-se o dispositivo com o aplicativo, foi possível obter dados de protocolo utilizado pela montadora no veículo escolhido pelo item 3.2.

#### 4.1.1 APLICATIVO DE *DEBUG*

Executando-se o aplicativo de *debug*, conforme explicado no item 3.5, foi enviado o comando “AT DP” (comando “AT” - *Descreva Protocolo*). Obteve-se que o veículo (item 3.2) utiliza o protocolo OBD segundo a norma ISO 15.765-4 (CAN 11/500) (ISO, 2016) para se comunicar com os módulos eletrônicos e para trocar informações com o *ELM327*, como pode ser verificado na Tabela 2. Este procedimento resumido e organizado é mostrado segundo a Figura 6.

Figura 6 – Aplicativo *debug* com resposta ao comando “AT DP”.



Fonte: O Autor.

Reconhecida a norma utilizada pelo veículo definido no item 3.2, pôde-se obter dados de monitoramento do veículo conforme descritos na Tabela 3. De acordo com os itens 3.4 e 3.5, foram executados comandos OBD manualmente. Alguns destes resultados estão evidenciados na Figura 7.

Figura 7 – Dados de resposta do veículo

```

Escaner OBDII
RAW DATA(0143 41 43 00 4C >)
RAW DATA(0142 41 42 34 16 >)
RAW DATA(013C 41 3C 0D 7C >)
RAW DATA(0134 41 34 7E 6F 7F E6 >)
RAW DATA(0133 41 33 65 >)
RAW DATA(0131 41 31 0D E6 >)
RAW DATA(0130 41 30 B0 >)
RAW DATA(0130 41 30 B0 >)
RAW DATA(41 2F C6 >)
TOO SMALL(012F)
RAW DATA(41 2E 12 >)
TOO SMALL(012E)
RAW DATA(0123 41 23 04 73 >)
RAW DATA(0121 41 21 00 00 >)
RAW DATA(011F 41 1F 01 23 >)
RAW DATA(011C 41 1C 1D >)
RAW DATA(0115 41 15 56 FF >)
RAW DATA(0113 41 13 03 >)
RAW DATA(0111 41 11 27 >)
RAW DATA(010F 41 0F 60 >)
RAW DATA(010E 41 0E 84 >)
RAW DATA(010D 41 0D 00 >)
RAW DATA(010C 41 0C 0E 72 >)
RAW DATA(010B 41 0B 37 >)
RAW DATA(0107 41 07 64 >)
RAW DATA(0106 41 06 7E >)
RAW DATA(0105 41 05 7C >)
RAW DATA(0104 41 04 25 >)
RAW DATA(0103 41 03 02 00 >)
RAW DATA( ELM327 v1.5 >)

```

Fonte: O Autor.

Percebe-se que a resposta de cada comando foi mostrada conforme a resposta do aplicativo entre os parêntesis da *string* “RAW\_DATA”. A resposta está como prevista no item 3.4 e o caractere “>” indica o fim da mensagem de resposta para cada comando enviado.

Os valores de todos os comandos testados no aplicativo de *debug* estão relacionados na Tabela 3, bem como os seus respectivos retornos. Os valores hexadecimais foram extraídos da mensagem principal e convertidos para decimal, conforme indica o manual do *ELM327* (ELMELECTRONICS, 2011) junto à norma de envio e recebimentos de comandos OBD, SAE J1979 (INTERNATIONAL, 2017).

Tabela 3 – Comandos OBD disponíveis para comunicação em um veículo segundo a norma SAE J1979.

| Comando OBD | PID (Identificação do parâmetro)                      | Dados do aplicativo | Valor em hexadecimal | Valores em decimal |
|-------------|---|---------------------|----------------------|--------------------|
| 0103        | <i>Fuel System Status</i>                             | 41030200            | 0200                 | 512                |
| 0104        | <i>Calculated Engine Load</i>                         | 410425              | 25                   | 37                 |
| 0105        | <i>Engine Coolant Temperature</i>                     | 41057C              | 7C                   | 124                |
| 0106        | <i>Short Term Fuel Trim - Bank 1</i>                  | 41067E              | 7E                   | 126                |
| 0107        | <i>Long Term Fuel Trim - bank 1</i>                   | 410764              | 64                   | 100                |
| 010B        | <i>Intake Manifold Absolute Pressure</i>              | 410B37              | 37                   | 55                 |
| 010C        | <i>Engine rpm</i>                                     | 410C0E72            | 0E72                 | 3698               |
| 010D        | <i>Vehicle Speed Sensor</i>                           | 410D00              | 00                   | 0                  |
| 010E        | <i>Ignition Timing Advance for #1 Cylinder</i>        | 410E84              | 84                   | 132                |
| 010F        | <i>Intake Air Temperature</i>                         | 410F60              | 60                   | 96                 |
| 0111        | <i>Absolute Throttle Position</i>                     | 411127              | 27                   | 39                 |
| 0113        | <i>Location of Oxygen Sensores</i>                    | 411303              | 03                   | 3                  |
| 0115        | <i>Bank 1 - sensor 2 (O2S)</i>                        | 411556FF            | 56FF                 | 22271              |
| 011C        | <i>OBD requirements to which vehicle is certified</i> | 411C1D              | 1D                   | 29                 |
| 011F        | <i>Time since Engine Start</i>                        | 411F0123            | 0123                 | 291                |
| 0121        | <i>Distance Traveled while MIL is activated</i>       | 41210000            | 0000                 | 0                  |
| 0123        | <i>Fuel Rail Pressure</i>                             | 41230473            | 0473                 | 1139               |
| 012E        | <i>Commanded Evaporative Purge</i>                    | 412E12              | 12                   | 18                 |
| 012F        | <i>Fuel Level Input</i>                               | 412FC6              | C6                   | 198                |
| 0130        | <i>Number of Warm-ups since DTCs cleared</i>          | 4130B0              | B0                   | 176                |
| 0131        | <i>Distance Traveled since DTCs cleared</i>           | 41310DE6            | 0DE6                 | 3558               |
| 0133        | <i>Barometric Pressure</i>                            | 413365              | 65                   | 101                |
| 0134        | <i>Bank 1 - Sensor 1 (wide range O2S)</i>             | 41347E6F7FE6        | 7E6F7FE6             | 2121236454         |
| 013C        | <i>Catalyst Temperature Bank 1, Sensor 1</i>          | 413C0D7C            | 0D7C                 | 3452               |
| 0142        | <i>Control Module Voltage</i>                         | 41423416            | 3416                 | 13334              |
| 0143        | <i>Absolute Load Value</i>                            | 4143004C            | 004C                 | 76                 |
| 0144        | <i>Fuel/Air Commanded Equivalence Ratio</i>           | 41448000            | 8000                 | 32768              |
| 0145        | <i>Relative Throttle Position</i>                     | 41450C              | 0C                   | 12                 |
| 0146        | <i>Ambient Air Temperature</i>                        | 414646              | 46                   | 70                 |
| 0147        | <i>Absolute Throttle Position B</i>                   | 414726              | 26                   | 38                 |
| 0149        | <i>Accelerator Pedal Position D</i>                   | 414924              | 24                   | 36                 |
| 014A        | <i>Accelerator Pedal Position E</i>                   | 414A25              | 25                   | 37                 |
| 014C        | <i>Commanded Throttle Actuator Control</i>            | 414C11              | 11                   | 17                 |
| 0151        | <i>Fuel Type being utilized</i>                       | 415103              | 03                   | 3                  |
| 0152        | <i>Alcohol Fuel Percentage</i>                        | 4152FE              | FE                   | 254                |

Fonte: O Autor.

#### 4.1.2 VALIDAÇÃO COM O APLICATIVO *TORQUE PRO*

De acordo com o item 3.6, o aplicativo *Torque Pro* foi instalado em um *smartphone*, pareado e conectado com o dispositivo OBDII comercial (item 3.3) de maneira semelhante ao aplicativo de *debug* (item 3.5) para validar e comparar os resultados dos valores dos dados obtidos através de ambos.

Validando os valores finais referentes a cada parâmetro, pôde-se também validar os comandos referentes ao aplicativo de *debug*, associados a um dispositivo OBD comercial que se utiliza do *ELM327* para obter as medições. Uma vez que os valores do aplicativo *debug* foram confirmados, é possível viabilizar tais comandos no microcontrolador *PIC*, assim, a resposta que este dispositivo obtiver, será enviada ao servidor em nuvem, conforme o *hardware* e *software* citados no item 3.7.

Figura 8 – Interface e medição de valores com o aplicativo *Torque Pro*.



Fonte: O Autor.

Através dos dados obtidos, para medida de rotação do motor, percentual de abertura do corpo de borboletas<sup>4</sup>, temperatura do líquido de arrefecimento do motor, pressão barométrica, temperatura ambiente e temperatura de entrada de ar, obtidos pelo aplicativo *Torque Pro* (Figura 8), foi elaborada a Tabela 4, para comparar os dados medidos com os dados calculados, de acordo com o manual do *ELM327* (ELMELECTRONICS, 2011).

Tabela 4 – Comparação entre os valores medidos e os calculados.

| Parâmetros                                | Aplicativo        | Aplicativo de <i>debug</i> |                           |              |
|---|-------------------|----------------------------|---------------------------|--------------|
|   | <i>Torque Pro</i> | decimal                    | correção                  | Valor final  |
| Revoluções por minuto (rpm)               | <b>918,0</b>      | 3698                       | divide por 4              | <b>924,5</b> |
| Abertura do corpo de borboletas (%)       | <b>15,30</b>      | 39                         | 255 --> 100%; 39<br>--> x | <b>15,29</b> |
| Temperatura líquido de arrefecimento (°C) | <b>84,00</b>      | 124                        | diminuir 40 °C            | <b>84,00</b> |
| Pressão Barométrica (kPa)                 | <b>101,0</b>      | 101                        | ---                       | <b>101,0</b> |
| Temperatura Ambiente (°C)                 | <b>30,00</b>      | 70                         | diminuir 40 °C            | <b>30,00</b> |
| Temperatura de entrada de ar (°C)         | <b>53,00</b>      | 96                         | diminuir 40 °C            | <b>56,00</b> |

Fonte: O Autor.

Os ensaios foram realizados consecutivamente, visto que a comunicação *Bluetooth*, neste caso, permite apenas um aplicativo de cada vez, para coletar as informações. Dado que as condições de temperaturas, pressão e abertura de corpo de borboletas foram similares, além de que, a variação do valor da rpm ser explicada pela natural variação da marcha lenta de um motor à combustão interna, pode-se declarar validados os valores obtidos manualmente e, por consequência desta amostra, o método utilizado.

## 4.2 INCLUSÃO E REQUISIÇÃO DOS DADOS COLETADOS A UMA REDE DE TELECOMUNICAÇÕES

A inclusão dos dados em nuvem é formatada pelo código fonte inserido no microcontrolador *PIC* (item 3.7). Entretanto, podem ser transmitidos até 222 *bytes*, até 160 vezes ao dia, de acordo

---

<sup>4</sup> Corpo de borboletas: termo utilizado para mensurar o conjunto responsável por permitir e regular a vazão mássica de ar no coletor de admissão de um veículo.

com o item 4.3.2. Os dados a serem enviados ao servidor podem ser concatenados até este tamanho de pacote, para serem enviados ao servidor em nuvem.

Ao serem transmitidos ao servidor, a aplicação requisita os dados através dos métodos descritos no item 3.8.1 e 3.8.2. Manipulando-se os dados requisitados de maneira reversa aos concatenados pelo dispositivo embarcado (item 3.7), extrai-se a informação de maneira segura e íntegra, uma vez que a rede de telecomunicações garante a confiabilidade dos dados segundo protocolo LoRaWAN.

## 4.3 VIABILIDADE COMERCIAL E TÉCNICA

### 4.3.1 DISPOSITIVO EMBARCADO

O dispositivo embarcado (item 3.7), desenvolvido pela DET do CP-IMT, possui um custo médio de R\$ 228,60 por seus respectivos componentes e processos para a fabricação de uma placa de circuito impresso, descrita no item 3.7, com os valores implementados de um circuito *ELM327* e um conector OBD para a placa de circuito impresso (Tabela 5).

Tabela 5 – Lista de materiais e processos

| Descrição                     | Footprint | Qtd | Parâmetro | Custo (US\$) | SubTotal (US\$) | SubTotal (R\$) |
|-------------------------------|-----------|-----|-----------|--------------|-----------------|----------------|
| Capacitor de tântalo          | CASE-A    | 1   | 10uF; 6V3 | \$ 0,35      | \$ 0,35         | R\$ 1,40       |
| Capacitor de tântalo          | CASE-A    | 1   | 10uF; 6V3 | \$ 0,35      | \$ 0,35         | R\$ 1,40       |
| Capacitor Cerâmico <i>SMD</i> | 0402      | 1   | 100nF     | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Capacitor Cerâmico <i>SMD</i> | 0402      | 1   | 100nF     | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Capacitor Cerâmico <i>SMD</i> | 0402      | 1   | 100nF     | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Capacitor de tântalo          | CASE-A    | 1   | 10uF; 6V3 | \$ 0,35      | \$ 0,35         | R\$ 1,40       |
| Capacitor Cerâmico <i>SMD</i> | 0402      | 1   | 100nF     | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Capacitor de tântalo          | CASE-A    | 1   | 10uF; 6V3 | \$ 0,35      | \$ 0,35         | R\$ 1,40       |
| Capacitor de tântalo          | CASE-A    | 1   | 10uF; 6V3 | \$ 0,35      | \$ 0,35         | R\$ 1,40       |
| Capacitor de tântalo          | CASE-A    | 1   | 10uF; 6V3 | \$ 0,35      | \$ 0,35         | R\$ 1,40       |
| Capacitor de tântalo          | CASE-A    | 1   | 10uF; 6V3 | \$ 0,35      | \$ 0,35         | R\$ 1,40       |
| Capacitor de tântalo          | CASE-A    | 1   | 10uF; 6V3 | \$ 0,35      | \$ 0,35         | R\$ 1,40       |
| Capacitor Cerâmico <i>SMD</i> | 0402      | 1   | 100nF     | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Capacitor Cerâmico <i>SMD</i> | 0402      | 1   | 100nF     | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Capacitor Cerâmico <i>SMD</i> | 0402      | 1   | OR        | \$ 0,10      | \$ 0,10         | R\$ 0,40       |



| Descrição  | Footprint       | Qty | Parâmetro            | Custo (US\$) | SubTotal (US\$) | SubTotal (R\$) |
|--|-----------------|-----|----------------------|--------------|-----------------|----------------|
| LED  | LED 0805 AM     | 1   | VD                   | \$ 0,15      | \$ 0,15         | R\$ 0,60       |
| LED  | LED 0603 AM     | 1   | STATUS - VM          | \$ 0,15      | \$ 0,15         | R\$ 0,60       |
| LED  | LED 0603 AM     | 1   | STATUS - VD          | \$ 0,15      | \$ 0,15         | R\$ 0,60       |
| Antenna PCI LoRa   | HADA31          | 1   |                      | \$ 1,00      | \$ 1,00         | R\$ 4,00       |
| USB On-The-Go (OTG) Mini-B Receptacle, Right Angle, SMT, 0.80mm (.031") Pitch, Solder Tabs with Back Cover, Recessed Type                          | 481900001       | 1   |                      | \$ 0,50      | \$ 0,50         | R\$ 2,00       |
| Transistor NPN   | SOT23-3N        | 1   | bc817                | \$ 0,15      | \$ 0,15         | R\$ 0,60       |
| Resistor   | 0402            | 1   | 100K; 5%             | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Resistor   | 0402            | 1   | 100K; 5%             | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Resistor   | 0402            | 1   | 100K; 5%             | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Resistor   | 0402            | 1   | 1K; 5%               | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Resistor   | 0402            | 1   | 470R; 5%             | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Resistor   | 0402            | 1   | 2K2; 5%              | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Resistor   | 0402            | 1   | 470R; 5%             | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Resistor   | 0402            | 1   | 100K; 5%             | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Resistor   | 0402 - RESISTOR | 1   | 1K; 5%               | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Resistor   | 0402 - RESISTOR | 1   | 10K; 5%              | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Resistor   | 0402            | 1   | NC                   | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Resistor   | 0402            | 1   | NC                   | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Resistor   | 0402            | 1   | 100K; 5%             | \$ 0,10      | \$ 0,10         | R\$ 0,40       |
| Microcontrolador PIC   | TQFP-64         | 1   | PIC24FJ128GA306-I/PT | \$ 3,00      | \$ 3,00         | R\$ 12,00      |
| Miniature Single-Cell, Fully Integrated Li-Ion, Li-Polymer Charge Management Controller, 4.2V, 5-Pin SOT-23, Industrial Temperature, Tape and Reel | SOT-23-OT5_N    | 1   | MCP73831T-2ACI/OT    | \$ 0,60      | \$ 0,60         | R\$ 2,40       |
| Regulador de tensão  | SOT23-3N        | 1   | MCP1702T-3302-E/CB   | \$ 0,50      | \$ 0,50         | R\$ 2,00       |
| LoRa Transceiver Module  | RN2903          | 1   | RN2903A              | \$ 11,00     | \$ 11,00        | R\$ 44,00      |
| Placa de circuito impresso   | Micropress      | 1   | Lorasimplesv2        |              |                 | R\$ 43,00      |
| Caixa plástica   |                 | 1   | impressão 3d         |              |                 | R\$ 15,00      |
| Pilhas   |                 | 2   | li-ion recarregável  | \$ 3,00      | \$ 6,00         | R\$ 24,00      |
| Conector OBDII pcb   |                 | 1   |                      | \$ 3,00      | \$ 3,00         | R\$ 12,00      |
| ELM327   |                 | 1   | V2.2                 | \$ 12,00     | \$ 12,00        | R\$ 48,00      |
| Subtotal   |                 |     |                      |              |                 | R\$ 228,60     |

Fonte: Disponível em ([https://github.com/imt-smartcampus/imt-developmentboard/blob/master/bom\\_2\\_1.xlsx](https://github.com/imt-smartcampus/imt-developmentboard/blob/master/bom_2_1.xlsx)).

Um dos valores mais significativos é o custo de fabricação da placa, que pode ser diluído de acordo com a quantidade a ser produzida. O seu preço inicial foi de R\$ 43,00, para este caso, tendo-se como base uma fabricação de 50 placas de circuito impresso.

#### **4.3.2 REDE DE TELECOMUNICAÇÕES**

Apesar de a plataforma *Smart Campus* do CP-IMT oferecer conectividade para o desenvolvimento de soluções de uma rede LPWAN, não é possível o envio de informações de dispositivos embarcados em outras regiões que não sejam aquelas cobertas pelos *gateways* conectados a ela. Há empresas que possuem redes de comunicação comerciais que cobrem regiões metropolitanas e principais cidades de estados, tornando possível a viabilidade e expansão desta aplicação em redes LPWAN.

Em termos de custos e operações, é possível enviar até 160 dados de um dispositivo com até 222 *bytes* de mensagem (não contemplando os cabeçalhos referentes aos protocolos) por um preço estimado em média de 2,00 R\$/mês por dispositivo embarcado.

O número de *bytes* a serem enviados por pacote é factível diante da realidade proposta, visto que o microcontrolador *PIC* (item 3.7) gerencia os dados a serem enviados e recebidos ao veículo. Não é necessário enviar dados a cada segundo ao servidor em nuvem, visto que importa que estes dados sejam monitorados pelo microcontrolador e este envie um resumo das informações à nuvem e não todas as informações requisitadas do veículo.

#### **4.3.3 APLICAÇÃO INTERFACE USUÁRIO**

Através de uma ferramenta de código aberto, conforme proposta pelo item 3.9.3, é possível estabelecer a conexão entre servidor em nuvem e interface WEB de acordo com o item 4.2.

Utilizou-se o *software Node-RED* para demonstrar a interface de informações características do veículo em um *dashboard* acessível de qualquer dispositivo com acesso à Internet através de um navegador (Figura 9), desde que o endereço em que o servidor WEB de inicialização do *Node-RED* esteja publicado na Internet.

Caso não esteja publicado na Internet, esta interface apenas estará disponível à rede local em que se encontra a aplicação.

Pode-se observar que o endereço de acesso à página do servidor *Node-RED* é característico de um endereçamento de uma rede local, iniciando-se por “192.168.43.X”.

Figura 9 – *Dashboard* para a demonstração de um aplicativo.



Fonte: O Autor.

Com o *dashboard* sendo executado conforme a plataforma escolhida (item 3.9.3) e o acesso sendo feito por um dispositivo com acesso à uma página da Internet por meio de um navegador, como visto na Figura 9, o único parâmetro que não pode ser requisitado pelo dispositivo embarcado é a placa do veículo, entretanto, o seu cadastro pode ser inserido manualmente na aplicação. O número de identificação do veículo, códigos de erros, *status* de leitura do conector OBD, registro do odômetro, pressão dos pneus, nível de combustível, consumo médio, tensão de bateria, dentre todos os mostrados na Tabela 3, são parâmetros que podem ser atualizados ciclicamente no tempo, para a formação de uma nuvem de dados na Internet através da requisição *online* (item 3.8.1).

Ainda na Figura 9, o valor de velocidade e sua derivada no tempo (aceleração), podem ser requisitados indiretamente do banco de dados através de uma API para gerar valores de gráficos e análises em outras plataforma com maior capacidade de equações matemáticas, como é o caso do *ThingSpeak*, uma plataforma em que se é possível gerar gráficos complexos e reconhecimento de padrões através de *softwares online* como *Matlab* e *Simulink*.

A Figura 9, portanto, mostra um exemplo de implementação e viabilidade para a escalabilidade em termos de quantidades de dispositivos embarcados e implementações disponíveis. Reiterando a Tabela 3, qualquer valor enviado pelo dispositivo embarcado ao servidor em nuvem pode ser mostrado no momento do envio de maneira *online* ou requisitado seu histórico por meio de uma API (consulta indireta ao banco de dados), conforme item 3.8.

## 5 CONCLUSÃO

Foi estudada a viabilidade e realizada uma proposta para desenvolvimento de uma aplicação para se obter remotamente os dados de sensores de veículos, de forma a serem transmitidos e disponibilizados em uma plataforma em nuvem de desenvolvimento de aplicações, utilizando-se de uma rede LPWAN certificada de baixo custo, escalável e de longo alcance.

Por fim, mapeou-se a cadeia de telecomunicações, a partir do levantamento da viabilidade comercial e tecnológica. Nota-se que neste trabalho não se vislumbrou o desenvolvimento dos circuitos eletrônicos, mas sim as suas implementações como componentes do conjunto e o escopo para viabilizar este projeto.

Este trabalho determinou que é possível, não apenas monitorar, mas buscar um desenvolvimento de soluções relacionadas à telemetria de dados veiculares associadas a uma plataforma de desenvolvimento de aplicações em nuvem de baixo custo, escalável e de longo alcance.

Levando-se em consideração o custo inicial mencionado, em relação à manutenções corretivas ou desgastes naturais, que tem por consequência o custo de parada e mão de obra envolvidas, notou-se que o preço relativo à conectividade é inferior ao da solução, possibilitando uma maior vida útil do veículo e a viabilidade técnica e comercial de uma aplicação, nos termos aqui defendidos, para este nicho.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

ALLIANCE, LoRa. **LoRaWAN 1.1 Specification**. LoRa Alliance, n. 1.1, p. 101, 2017. Disponível em: <<https://lora-alliance.org/resource-hub/lorawantm-specification-v11>>.

BENTLEY. **OBDII Control**. Disponível em: <<http://ai2.appinventor.mit.edu/?locale=en&galleryId=6562490234699776>>. Acesso em: 1 jan 2019.

DET-CP-IMT. **Smart Campus Mauá**. Disponível em: <<https://smartcampus.maua.br/>>. Acesso em: 26 mar 2018.

EDUARDO KARASINSKI. **O que é Código ASCII**. Disponível em: <<https://www.tecmundo.com.br/imagem/1518-o-que-e-codigo-ascii.htm>>. Acesso em: 4 abr 2019.

ELMELECTRONICS. **OBD to RS232 Interpreter ELM327**. Electronics, p. 1- 9,22,24-32,57-61, 2011. Disponível em: <<http://elmelectronics.com/DSheets/ELM327DS.pdf>>.

GILLESPIE, Thomas D. **Fundamentals of Vehicle Dynamics**. [S.l: s.n.], 1992.

GOES, Vitor Elisio e MENEZES, Oliveira. **16/06/2018 SEI/ANATEL - 2184849 - Ato**. p. 1–14, 2018.

INTERNATIONAL, SAE. **E/E Diagnostic Test Modes J1979\_201702**. . [S.l: s.n.], 2017. Disponível em: <[https://www.sae.org/standards/content/j1979\\_201702/](https://www.sae.org/standards/content/j1979_201702/)>.

ISO. **ISO 15765-4:2016 (Road vehicles -- Diagnostic communication over Controller Area Network (DoCAN) -- Part 4: Requirements for emissions-related systems)**. . [S.l: s.n.], 2016. Disponível em: <<https://www.iso.org/standard/67245.html>>.

STRATH, Scott J. e ROWLEY, Taylor W. **Wearables for promoting physical activity**. Clinical Chemistry, v. 64, n. 1, p. 53–63, 2018.