



UNIVERSIDADE FEDERAL DO AMAZONAS – UFAM
FACULDADE DE TECNOLOGIA – FT
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA – PPGEE

WALMIR ACIOLI E SILVA

**SISTEMA EMBARCADO DE DIAGNÓSTICO VEICULAR INTEGRADO A
AMBIENTES INTELIGENTES**

MANAUS-AM

2019



UNIVERSIDADE FEDERAL DO AMAZONAS – UFAM
FACULDADE DE TECNOLOGIA – FT
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA – PPGEE

WALMIR ACIOLI E SILVA

**SISTEMA EMBARCADO DE DIAGNÓSTICO VEICULAR INTEGRADO A
AMBIENTES INTELIGENTES**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica, área de concentração Controle e Automação de Sistemas.

Orientador: Prof. Dr. –Ing. Vicente Ferreira de Lucena Junior

MANAUS-AM

2019

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

S586s Silva, Walmir Acioli e
Sistema Embarcado de Diagnóstico Veicular Integrado a
Ambientes Inteligentes / Walmir Acioli e Silva. 2019
87 f.: 31 cm.

Orientador: Prof. Dr. –Ing. Vicente Ferreira de Lucena Júnior
Dissertação (Mestrado em Engenharia Elétrica) - Universidade
Federal do Amazonas.

1. Veículos conectados. 2. Ambientes Inteligentes (AIs). 3.
Gateway. 4. Nuvem. 5. On-Board Unit II (OBD-II). I. Lucena Júnior,
Prof. Dr. –Ing. Vicente Ferreira de II. Universidade Federal do
Amazonas III. Título

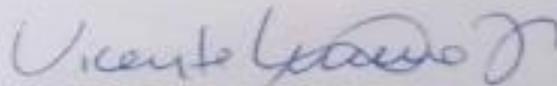
WALMIR ACIOLI E SILVA

SISTEMA EMBARCADO DE DIAGNÓSTICO VEICULAR INTEGRADO A AMBIENTES INTELIGENTES

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovado em 11 de setembro de 2019.

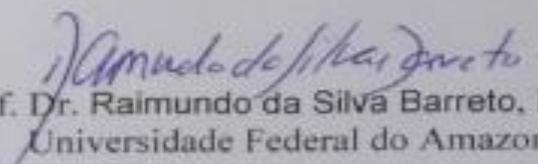
BANCA EXAMINADORA



Prof. Dr. Vicente Ferreira de Lucena Junior, Presidente
Universidade Federal do Amazonas



Prof. Dr. Eduardo James Pereira Souto, Membro
Universidade Federal do Amazonas



Prof. Dr. Raimundo da Silva Barreto, Membro
Universidade Federal do Amazonas

*Dedico este trabalho ao meu filho Murilo
Pinho e Silva, que nasceu durante o
mestrado e me deu forças para superar
todas as dificuldades que apareceram no
decorrer desta jornada.*

Agradecimentos

Agradeço primeiramente à Deus, pela jornada da vida.

Agradeço a minha esposa, Ana Paula Marreira de Pinho e meu filho Murilo Pinho e Silva, por terem me apoiado nos momentos difíceis e de estresse durante o mestrado, pela paciência e os ótimos momentos que passamos juntos.

Agradeço aos meus pais, Walmir Castro e Silva e Sandra Nonata Acioli e Silva, as minhas irmãs, Michele e Helena, aos meus sobrinhos Pedro e Manuella.

Agradeço aos meus tios, Agno Acioli, Emanuel Castro e Maria Damiana pela atenção e ensinamentos que me fizeram amadurecer.

Agradeço ao Professor Vicente pelo trabalho de orientação técnica e por todo apoio fornecido durante a jornada do programa.

Agradeço aos meus sogros, Izabel Marreira e Luiz Pinho.

Agradeço aos amigos do Centro Federal de Educação Tecnológica do Amazonas (CEFET-AM), Cássio Glauber, Valcimar Castro, Rodrigo Azevedo, Djalma Beraldo, João Victor Gonçalves e Flávio Monteiro.

Agradeço aos amigos do Laboratório de Ambientes Inteligentes da Universidade Federal do Amazonas (UFAM), Vinícius Souza, Guido, Yuri, especialmente aos colegas Robson Guimarães e Mário Alves pela ajuda e companheirismo na conclusão deste trabalho.

Agradeço à Prof^ª. Ellen Barbosa de Andrade pelo apoio e ensinamentos dados incondicionalmente.

Agradeço aos amigos da empresa, Lucas Taba, Renan Bezerra, Jackeline Dourado, Paulo Santos, Larissa Queiroz, Brenda Bezerra e Beatriz Ordones.

Agradeço à UFAM, pelo aprendizado construído no decorrer do mestrado.

Agradeço aos professores do PPGEE que ensinaram categoricamente o conteúdo técnico científico do programa.

Agradeço aos colegas do Laboratório de Termociências do curso de Engenharia Mecânica pela disponibilização de tempo e da bancada do motor que realizei os experimentos deste trabalho.

Agradeço a CAPES, CNPq e FAPEAM pelo suporte que viabilizou a pesquisa científica, a viagem para Alemanha e as publicações.

Resumo

A manutenção visa manter o veículo em boas condições de funcionamento e evitam surpresas desagradáveis, como: pane mecânica e quebra de componentes por causa de peças danificadas, aumentando os prejuízos. Buscando evitar essas surpresas, este trabalho propôs e desenvolveu um sistema com uma arquitetura composta por dispositivo embarcado, servidor na nuvem e local. Este sistema conecta o veículo ao Ambiente Inteligente (AmI) de uma casa e é capaz de notificar o motorista através de dispositivo conectado a este ambiente ao detectar parâmetros anormais de funcionamento do veículo. Dessa forma, o motorista pode se antecipar e tomar as devidas providências para não aumentar os problemas e os prejuízos. Para a concepção deste sistema, foram utilizados recursos tecnológicos empregados em redes veiculares, como o *On-Board Diagnostic II* (OBD-II). Nos AmIs foram observados os conceitos e aplicações focados para a concepção de um dispositivo de rede chamado *gateway*. Uma das principais contribuições deste trabalho foi a integração de forma automática do veículo com o ambiente da casa que pode notificar o motorista ao detectar parâmetros anormais sobre o veículo. Assim, veículos fabricados a partir de 2010 podem usar nosso sistema, o que representa no Brasil cerca de 25 milhões de veículos que se conectar aos seus proprietários. Os testes com o sistema foram realizados em uma bancada de motor no Laboratório de Termociências do curso de Engenharia Mecânica da UFAM. Os resultados mostraram que o sistema funcionou adequadamente ao detectar que o sensor de temperatura de arrefecimento do motor estava superior ao parâmetro normal de 100° Celsius. Após os dados serem transmitidos do dispositivo embarcado para o servidor na nuvem e depois para o AmI da casa, o parâmetro anormal detectado é coletado e o AmI da casa envia uma notificação para o *smartphone* do motorista por meio de aplicativo de mensagem informando a temperatura e a hora da detecção.

Palavras-chave: Veículos conectados; Ambientes Inteligentes (AmIs); *Gateway*; Nuvem; *On-Board Diagnostic II* (OBD-II).

Abstract

Maintenance aims to keep the vehicle in good working order and avoid unpleasant surprises such as mechanical breakdown and breakdown of components due to damaged parts, increasing damage. Trying to avoid these surprises, this work proposed and developed a system with an architecture composed by embedded device, cloud server and local. This system connects the vehicle to the Ambient Intelligent (AmI) of a home and is able to notify the driver through a device connected to this environment by detecting abnormal vehicle operating parameters. In this way, the driver can anticipate and take appropriate steps not to increase problems and damage. For the design of this system, technological resources used in vehicular networks were used, such as On-Board Diagnostic II (OBD-II). In the AmIs, we observed the concepts and applications focused on the design of a network device called gateway. One of the main contributions of this work was the automatic integration of the vehicle with the home environment that can notify the driver by detecting abnormal vehicle parameters. Thus, vehicles manufactured from 2010 can use our system, which represents in Brazil about 25 million vehicles that connect to their owners. Tests with the system performed on an engine bench at the Thermoscience Laboratory of the Mechanical Engineering course at UFAM. The results showed that the system functioned properly by detecting that the engine coolant temperature sensor was above the normal 100 ° Celsius parameter. After data is transmitted from the embedded device to the cloud server and then to the home AmI, the detected abnormal parameter is collected and the home AmI sends a notification to the driver's smartphone via a message app informing the temperature and the time of detection.

Keywords: Connected vehicles; Ambient Intelligent (AmI); Gateway; A cloud; On-Board Diagnostic II (OBD-II).

LISTA DE SIGLAS

AmI – Ambiente Inteligente
API – Application Programming Interface
ECU – Electronic Unit Control
OBD-II – On-Board Diagnostic II
OBU – On-Board Unit
CARB – California Air Resources Board
CAN – Controller Area Network
EPA – Environmental Protection Agency
EUA – Estados Unidos da América
GNU – *General Public License*
GPS – Global Positioning System
HTTP – Hypertext Transfer Protocol
IoT – Internet of Things
IoV – Internet of Vehicle
IP – Internet Protocol
ISO – International Standards Organization
JSON – JavaScript Object Notation
M2M – Machine to Machine
PIDs – Parameter IDs
PHP – Hypertext Preprocessor
REST – Representational State Transfer
SAE – Society of Automotive Engineers
SIoV – Social Internet of Vehicle
SQL – *Structured Query Language*
SSH – Secure Shell
TCP – Transmission Control Protocol
VCI – Vehicle Communication Interface
V2I – Vehicle to Infrastructure
V2V – Vehicle to Vehicle
V2X – Vehicle to Everything
UART – Universal Asynchronous Receiver/Transmitter
UDP – User Datagram Protocol

UML – Unified Modeling Language.

URL – Uniform Resource Locator

LISTA DE FIGURAS

Figura 1: Painel do veículo.....	17
Figura 2: Porta OBD-II.....	22
Figura 3: Dispositivo ELM 327.....	23
Figura 4: Comandos PID do serviço 1	24
Figura 5: Resultado geral de publicação referentes ao tema	30
Figura 6: Esboço geral do sistema proposto.....	51
Figura 7: Módulo veículo	51
Figura 8: Visão geral de funcionamento do sistema.....	52
Figura 9: Diagrama de atividades.....	54
Figura 10: Protótipo do dispositivo OBU.....	57
Figura 11: Script com funções para habilitar conexão e transmissão de mensagens	58
Figura 12: Módulos do OBU.....	59
Figura 13: Funções do programa fsm_main	59
Figura 14: Funções do programa connectOBD2	60
Figura 15: Funções do programa get_data_car	60
Figura 16: Colunas da tabela criada para salvar os dados	61
Figura 17: Funções do programa functionSubString.....	62
Figura 18: Fórmulas de conversão de alguns sensores.....	62
Figura 19: Script de conversão do sensor de temperatura do motor	63
Figura 20: Funções do programa hex_to_int.....	63
Figura 21: Funções do programa data_car_http_request.....	64
Figura 22: Tela inicial da aplicação na nuvem	66
Figura 23: Esboço de entrada e saída dos dados	67
Figura 24: Script que solicita dados da nuvem com API	68
Figura 25: Esboço do fluxo de dados no AmI.....	70
Figura 26: Arquitetura de comunicação e fluxo de dados	71
Figura 27: Bancada do motor do laboratório de mecânica.....	73
Figura 28: Quadro de inserção de falhas	74
Figura 29: Porta OBD-II da bancada.....	74
Figura 30: Cenário de utilização do sistema.....	75
Figura 31: Aplicação web que armazena dados do OBU no servidor na nuvem	76
Figura 32: Histórico dos sensores.....	76

Figura 33: Alertas de temperatura	77
Figura 34: Alertas de velocidade	77
Figura 35: Alertas de Rotação	78
Figura 36: Notificações via aplicativo	79

LISTA DE TABELAS

Tabela 1: Descrição dos pinos da porta OBD-II	22
Tabela 2: Serviços fornecidos pelo OBD-II	24
Tabela 3: Questões de delimitação da pesquisa.....	27
Tabela 4: Questões de delimitação da pesquisa.....	27
Tabela 5: Tabela de adequação das estratégias de busca a cada uma das bases de dados ...	28
Tabela 6: Síntese dos trabalhos sobre AmI	35
Tabela 7: Continuação síntese dos trabalhos sobre AmI.....	36
Tabela 8: Tabela de comparação sobre comunicação veicular.....	42
Tabela 9: Continuação da tabela de comparação sobre comunicação veicular	43
Tabela 10: Tabela de comparação dos trabalhos relacionados.....	48
Tabela 11: Exemplos de comandos utilizados.....	60
Tabela 12: Processamento das mensagens	62
Tabela 13: Descrição das funções	64
Tabela 14: Configuração servidor local.....	68
Tabela 15: Módulos do sistema	69

SUMÁRIO

LISTA DE SIGLAS	VIII
LISTA DE FIGURAS	X
LISTA DE TABELAS	XII
1. Introdução	15
1.1 Contextualização	15
1.2 Definição do problema	16
1.2.1 Cenários para o entendimento do problema	17
1.3 Motivação	18
1.4 Objetivos.....	19
1.4.1 Objetivo Geral	19
1.4.2 Objetivo Específico	19
1.5 Organização dos Capítulos	20
2. Conceitos Fundamentais para o Desenvolvimento do Sistema Proposto	21
2.1 <i>On-Board Diagnostic II</i> (OBD-II).....	21
2.2 Framework.....	25
2.3 <i>Gateway</i>	25
2.4 Comandos AT.....	25
2.5 Ambientes Inteligentes (AmI)	26
2.6 Considerações finais sobre o capítulo	26
3. Trabalhos relacionados	27
3.1 Trabalhos Relacionados sobre Ambientes Inteligentes	30
3.2 Trabalhos Relacionados sobre comunicação veicular	36
3.3 Contribuições dos trabalhos relacionados	45
3.4 Contribuição desta Pesquisa	47
3.5 Considerações finais sobre o capítulo	49
4. Arquitetura do sistema	50
4.1 Concepção da Solução.....	50
4.2 Visão geral e arquitetura do sistema.....	50
4.3 Considerações finais do capítulo	55
5. Implementação do sistema	56
5.1 Conexão veicular	56
5.2 OBU.....	56

5.3 Nuvem	65
5.4 AmI.....	68
5.5 Arquitetura de comunicação	70
5.6 Considerações finais do capítulo	72
6. Resultados Experimentais e Discussão	73
6.1 Experimentos	73
6.2 Resultados Obtidos	75
6.3 Discussão	80
7. Conclusão	81
7.1 Contribuições.....	82
7.2 Limitações e Trabalhos Futuros	82
Referências	84

1. Introdução

1.1 Contextualização

Um Ambiente Inteligente (AmI) é definido como um espaço físico que dispõem de diversos recursos computacionais e serviços embarcados com interação entre humano e computador de forma visível e invisível (Coulouris, Dollimore e Kindberg, 2012). Esses ambientes tem o propósito de auxiliar as pessoas e possuem características, como: serem discretos, interconectados, adaptáveis, dinâmicos, integrados e inteligentes, além de utilizarem dispositivos de baixo custo e fácil manuseio para que o usuário fique imerso em um ambiente automatizado, de maneira a viabilizar a compreensão das suas rotinas (Sadri, 2011).

Um exemplo comum de aplicação de AmI é utilizá-lo em uma casa a fim de monitorar as pessoas que precisam de cuidados, porém sem a presença de outras pessoas; um exemplo são os ambientes assistidos, nos quais geralmente há idosos que necessitam de monitoramento de forma “invisível” por familiares e médicos (Yacchirema e Member, 2016).

Outro elemento essencial para esta pesquisa é a evolução tecnológica dos veículos, os quais eram puramente mecânicos e se tornaram verdadeiros computadores sobre rodas, permitindo que fossem inseridos em contextos como: cidades inteligentes, sistemas de transporte inteligente, internet das coisas e internet dos veículos.

A ideia de conectar o veículo com a estrada, visando a segurança, iniciou com as redes veiculares. Essas redes foram inicialmente formadas por veículos automotores (*vehicle to vehicle* - V2V) e equipamentos fixos (*vehicle to infrastructure* - V2I) geralmente localizados às margens de ruas ou estradas.

Com a evolução tecnológica da internet móvel e o advento da Internet das Coisas (*Internet of Things* - IoT), as comunicações não se limitaram entre os veículos e pontos fixos na estrada. As redes veiculares foram se tornando a internet dos veículos (*Internet of Vehicle* - IoV), que envolve a comunicação entre veículo e qualquer coisa que esteja conectada à internet (*vehicle to everything* - V2X) (Naik, Member e Choudhury, 2019).

No cenário atual, os conceitos de coisas conectadas estão em pleno desenvolvimento e vários dispositivos estão sendo preparados para se conectar à internet; um exemplo é possibilitar que eletrodomésticos de uma casa, dispositivos do ambiente e veículos transmitam seus dados de funcionamento para a nuvem. Dessa forma é possível criar serviços, como monitoramento, controle e gerenciamento do ambiente, os quais visam facilitar ainda mais a vida do ser humano auxiliando-o nas tarefas do dia a dia.

Um sistema que integre o veículo com a casa no contexto de AmI necessita de

tecnologias que coletam, transmitam e distribuam os dados para esses ambientes de forma automática, para que os usuários possam ser notificados sobre possíveis alterações no funcionamento do veículo. Contudo, os trabalhos com resultados mais concretos, utilizam o *smartphone* como principal dispositivo de coleta e transmissão de dados, além de terem como foco o comportamento do motorista.

1.2 Definição do problema

Existem vários desafios para conceber um método que integre dispositivos e sistemas, principalmente quando se leva em consideração as conexões entre os dispositivos de forma automática e os dados que trafegam entre eles. Nesse caso, há a necessidade de um dispositivo capaz de realizar interoperabilidade dos demais dispositivos que estão conectados à uma rede interna ou à internet (Vivek e Sunil, 2016).

Um desafio importante a ser observado na arquitetura e na construção de um sistema que integra diferentes dispositivos e tecnologias, são os requisitos que o compõem. Neste caso, são considerados os dados de aquisição e a estrutura de comunicação que deve ser adotada com relação ao armazenamento e fluxo de dados (Atzori *et al.*, 2018).

Uma aplicação na qual um dispositivo embarcado coleta e armazena dados do veículo possibilitaria que esses dados pudessem ser transmitidos via rede sem fio para um servidor na nuvem, no qual os dados seriam armazenados.

Esse sistema possibilitaria que o AmI da casa acesse os dados armazenados no servidor na nuvem a fim de coletar os dados anormais de funcionamento do veículo e notificar o usuário por meio de dispositivos conectados à rede do AmI da casa.

Considerando que os paradigmas de AmI estão sendo aplicados em uma casa e que um roteador faça a interconexão entre o dispositivo embarcado do veículo, servidor na nuvem e AmI da casa, tem-se o ambiente necessário para aplicar esse sistema.

A maioria dos trabalhos com veículos visam monitorar o comportamento do motorista e o estado do veículo em tempo real. Contudo é necessário um *smartphone* com internet móvel para transmissão dos dados, e, portanto, torna a solução dependente da rede móvel e de um dispositivo multitarefa.

Preparar um ambiente residencial conectado e oferecer serviços de forma transparente ao usuário esbarra em uma problemática investigada neste trabalho, apresentada a seguir.

Como integrar o veículo por meio de um dispositivo embarcado em um AmI de uma

casa, considerando um modelo de tratamento e armazenamento de dados e mensagens de notificação sobre dados anormais de funcionamento do veículo?

A partir desse questionamento, faz-se necessária a construção da hipótese para ser confrontada no decorrer deste trabalho.

H1 – É possível integrar um veículo a um AmI da casa que por meio de dispositivos conectados à rede desse ambiente, seja capaz de notificar o usuário sobre parâmetros anormais de funcionamento do veículo.

Para que H1 seja verdadeira é necessário coletar dados do veículo que podem indicar problema. Serão apresentados dois cenários para ilustrar a problemática tratada neste trabalho.

1.2.1 Cenários para o entendimento do problema

A Figura 1 apresenta o painel de um veículo no qual são destacadas as informações dos principais sensores que devem ser monitorados pelo motorista, que são: velocidade, rotação do motor, temperatura de arrefecimento do motor, sensor da porta, sensor do óleo do motor e a luz espia (cor amarela) que indicam problemas nos sensores do veículo.



Figura 1: Painel do veículo

Para entender melhor a problemática que esta pesquisa visa solucionar, será descrito o seguinte cenário: a utilização diária do veículo durante a semana se locomovendo da casa para o trabalho e vice-versa é uma rotina comum para grande maioria das pessoas. Contudo, muitas vezes não se percebe os sinais que indicam que o veículo pode dar problemas, como a temperatura de arrefecimento crescente do motor ou mesmo quando a luz espia acende indicando problemas em outro sensor do veículo.

Visando alertar o motorista, o sistema consiste em notificar sobre possíveis problemas

de funcionamento do veículo por meio de dispositivos conectados ao AmI da casa, como TV digital, *tablets*, *smartwatch* e *smartphones*. Dessa forma, o problema é detectado logo no início, possibilitando que a manutenção preventiva seja feita antes da quebra total e evitando danos maiores.

1.3 Motivação

A manutenção é algo presente em veículos, máquinas e equipamentos, que devem passar por revisões e consertos de tempos em tempos para garantir seu funcionamento adequado e evitar que problemas mais graves ocorram, comprometendo outras peças e aumentando o prejuízo (MORAIS, 2011).

Existem dois tipos de manutenção: a manutenção preventiva e manutenção corretiva. Cada uma delas tem seu período certo para ser realizada. No entanto, em alguns casos, a manutenção preventiva é a mais recomendada, pois evita que danos maiores sejam causados ao veículo, máquina ou equipamento. Quando não é possível evitar um dano, a manutenção corretiva deve ser feita (Jezzini, Ayache e Elkhansa, 2013).

A manutenção corretiva se resume ao conserto ou troca de uma peça (ou do equipamento) somente após a ocorrência de uma falha ou defeito. Exemplo: se o óleo do motor não for trocado na quilometragem recomendada e continuar rodando por muitos quilômetros, seu motor poderá fundir por falta de lubrificação. Neste caso, o motor deverá ser aberto para trocar as peças com defeitos ou quebradas, resultando em um gasto muito maior (Jezzini, Ayache e Elkhansa, 2013).

A manutenção preventiva tem o propósito de evitar que danos maiores sejam causados. Isso é feito realizando reparos periódicos e impede que o veículo apresente problemas em momentos inoportunos (necessitando de manutenção corretiva), além de prolongar a sua vida útil (Jezzini, Ayache e Elkhansa, 2013).

Direcionando a manutenção preventiva para veículos, modelos de otimização baseados na manutenção preventiva e na programação da rota, buscando reduzir o impacto das falhas do veículo, baseando-se em dados das manutenções e quilômetros percorridos (Husniah, Pasaribu e Iskandar, 2015; Wang, Ma e Fu, 2019). A manutenção preventiva é necessária não só para evitar prejuízos financeiros, evita também as consequências que podem acarretar acidentes, engarrafamentos e a inoperância de serviços essenciais.

O veículo possui diversos fatores que contribuem para o mal funcionamento, como: pneus com calibração errada, componentes do motor desregulados ou danificados e

combustível adulterado. Em muitos casos o motorista não percebe as pequenas variações mesmo com o veículo com a manutenção em dia.

Esses problemas podem ser diagnosticados a partir da unidade central do veículo por meio de adaptadores *On-Board Diagnostic II* (OBD-II) conectados na porta OBD-II (Lalli, 2015). Contudo, esse procedimento é realizado nas revisões ou quando o veículo apresenta alterações perceptíveis ao motorista que o obriga a procurar ajuda profissional.

O sistema proposto será capaz de conectar o veículo com a casa com AmI poderá notificar o motorista sobre possíveis problemas de funcionamento, permitindo que ele realize a manutenção preventiva ou corretiva.

Nosso sistema poderá ser aplicado nos veículos que possuem a porta OBD-II, o que representa no Brasil, cerca de 25 milhões de veículos (Amaral, 2018), possibilitando que seus usuários sejam notificados em dispositivos conectados à rede da casa.

1.4 Objetivos

1.4.1 Objetivo Geral

Desenvolver um sistema que integre um veículo a um Ambiente Inteligente, enviando dados à nuvem de forma automática, permitindo que o usuário seja notificado sobre dados anormais de funcionamento do veículo por meio de dispositivos conectados a este ambiente.

1.4.2 Objetivo Específico

- Avaliar os trabalhos relacionados e desenvolver uma arquitetura robusta e que facilite o fluxo de dados do veículo;
- Desenvolver um dispositivo capaz de coletar dados de funcionamento do veículo fabricados a partir de 2010 e transmitir via rede sem fio para o servidor na nuvem de forma automática;
- Construir algoritmos no servidor na nuvem para receber, armazenar e filtrar os dados de funcionamento do veículo, para disponibilizar links de acesso para o AmI da casa realizar as consultas e coletar os dados;
- Desenvolver programas para o AmI da casa capazes de notificar o usuário ao detectar parâmetros anormais de funcionamento do veículo por meio dos dispositivos conectados a este ambiente.

1.5 Organização dos Capítulos

Este trabalho está organizado da seguinte forma: o Capítulo 2 apresenta os principais fundamentos sobre as tecnologias e ferramentas aplicadas, o que inclui o OBD-II e as ferramentas utilizadas no desenvolvimento do sistema proposto. O Capítulo 3 apresenta os métodos e as bases de pesquisa dos trabalhos relacionados sobre AmI e redes veiculares, em ambos é realizada uma síntese de cada trabalho e ao final uma tabela de comparação é montada. O Capítulo 4 apresenta a concepção da solução e introduz uma visão geral da arquitetura do sistema. O Capítulo 5 descreve a implementação de cada parte do sistema a partir da arquitetura. O Capítulo 6 apresenta os resultados e discussões dos experimentos. Finalizando, o Capítulo 7, aborda as considerações finais e trabalhos futuros.

2. Conceitos Fundamentais para o Desenvolvimento do Sistema Proposto

Este capítulo tem como objetivo apresentar os principais conceitos sobre as tecnologias e ferramentas utilizadas nesta pesquisa. Primeiramente são abordados conceitos sobre redes internas de comunicação dos veículos, que são fundamentais para o desenvolvimento do sistema proposto. Finalmente são apresentados conceitos e ferramentas para o desenvolvimento dos *softwares* do sistema proposto.

2.1 On-Board Diagnostic II (OBD-II)

Nas últimas décadas, os veículos tiveram uma evolução notável impulsionada pela preocupação com o meio ambiente, pois foi necessário desenvolver motores mais eficientes, tornando-os verdadeiros computadores sobre rodas. A Unidade de Comando Eletrônico (UCE), em inglês *Electronic Unit Control (ECU)*, é o cérebro do veículo, responsável por controlar todos os sensores e atuadores do motor para mantê-los funcionando corretamente e alertando no painel o mal funcionamento. No entanto, quando há problemas, só é possível obter acesso a UCE através da porta OBD-II (Manavella, 2018), o qual é um padrão desenvolvido para atender às especificações de controle de emissões, padronização da porta e protocolos de comunicação estabelecidas pelo *California Air Resources Board (CARB)* e *Environmental Protection Agency (EPA)*, agência de meio ambiente dos Estados Unidos da América (EUA).

O padrão OBD-II, estabelecido pelo CARB, padroniza tudo relacionado ao diagnóstico veicular seguindo protocolos de comunicação estabelecidos pela *Society of Automotive Engineers (SAE)* e pela *International Standards Organization (ISO)* (California, 2013), as quais estabeleceram que todos os automóveis fabricados a partir de 1996, nos EUA e Europa, adotem esse padrão (Obdii *et al.*, 1996; Greening, 2001). No Brasil, veículos fabricados a partir de 2010 vêm obrigatoriamente com esse padrão.

A Figura 2 apresenta a interface OBD-II, ela contém cinco protocolos de sinalização. Contudo, a maioria dos veículos implementa apenas um protocolo. Também é possível decidir qual protocolo usar dependendo dos pinos disponíveis no conector J1962 (OBD-II) (Sawant, 2018). Esses são os cinco protocolos de sinalização:

1. SAE J1850 PWM (*Pulse Width Modulation*);
2. SAE J1850 VPW (*Variable Pulse Width*);
3. ISO 9141-2 (K-LINE or L-LINE);

4. ISO 14320 KWP2000 (*Keyword Protocol 2000*);
5. ISO 15765 CAN.

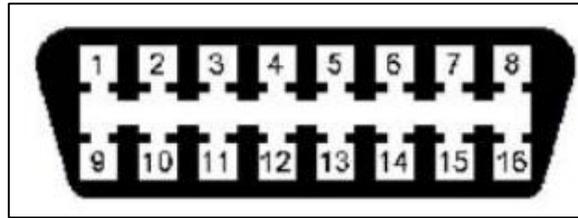


Figura 2: Porta OBD-II

A descrição dos pinos de conexão pode ser visualizada na Tabela 1.

Tabela 1: Descrição dos pinos da porta OBD-II

PIN	Description	PIN	Description
1	Manufacturer Specific	9	Manufacturer Specific
2	J1850 Bus+	10	J1850 Bus-
3	Manufacturer Specific	11	Manufacturer Specific
4	Chassis GND	12	Manufacturer Specific
5	Signal GND	13	Manufacturer Specific
6	CAN High	14	CAN Low
7	ISO-9141-2 K-LINE	15	ISO-9141-2 L-LINE
8	Manufacturer Specific	16	+12 V (Battery Power)

A porta OBD-II possibilita a análise de desempenho e o diagnóstico de sistemas veiculares com os dispositivos adaptadores ou *scanners* OBD-II que permitem o monitoramento e autodiagnóstico com a obtenção de diversas informações a respeito do veículo (Marin *et al.*, 2017). Esses dispositivos são responsáveis por abstrair o protocolo de baixo nível e apresentar uma interface simples que pode ser chamada através de um receptor / transmissor assíncrono universal, em inglês, *Universal Asynchronous Receiver/Transmitter* (UART), que tipicamente é uma ferramenta de diagnóstico de mão ou um programa de computador conectado por USB, RS-232, *Bluetooth* ou Wi-Fi. O UART é um dispositivo de *hardware* de computador que converte os dados entre as formas seriais e paralelas.

O ELM 327 foi o dispositivo UART escolhido neste projeto para coletar os dados do veículo, com o qual será possível obter diversos dados que podem ser repassados para outros

dispositivos por rede sem fio (Wi-Fi ou *Bluetooth*), vide Figura 3. O dispositivo ELM 327 possui um manual detalhado e de fácil acesso (ELM327, 2018).

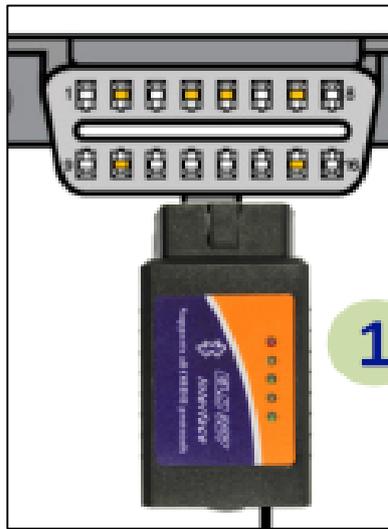


Figura 3: Dispositivo ELM 327

Nesse manual é possível ter acesso aos comandos necessários para obter os dados da ECU do veículo, bem como os procedimentos a serem realizados. Alguns comandos devem ser enviados antes de coletar os dados, dentre os quais, tem-se:

- ATZ – tem a função de fazer um soft-reset do dispositivo ELM 327;
- ATSP0 – coloca o ELM 327 em modo automático quanto ao protocolo (ISO ou SAE);

A tecnologia OBD-II suporta 10 modos de operação. Cada um deles possui uma série específica de comandos que retorna dados dos sensores e atuadores presentes no veículo. Para solicitar esses dados, são usados códigos chamados de Identificação de Parâmetro, em inglês *Parameter Id* (PID). A Tabela 2 mostra os respectivos modos de operação (Silva *et al.*, 2019).

Tabela 2: Serviços fornecidos pelo OBD-II

Serviço hex	Descrição
01	Mostrar dados atuais
02	Mostrar dados do quadro congelado
03	Mostrar códigos de problemas de diagnóstico armazenados
04	Limpar códigos de problemas de diagnóstico e valores armazenados
05	Resultados do teste, monitoramento do sensor de oxigênio (somente não <i>Controller Area Network (CAN)</i>)
06	Resultados de testes, outros componentes / monitoramento do sistema (resultados de testes, monitoramento do sensor de oxigênio somente para CAN)
07	Mostrar códigos de problemas de diagnóstico pendentes (detectados durante o ciclo de condução atual ou anterior)
08	Operação de controle do componente / sistema de bordo
09	Solicitar informações do veículo
0A	Códigos de Problema de Diagnóstico Permanente

Logo abaixo na Figura 4, há exemplos de comandos do serviço 01 (mostrar dados atuais) e das fórmulas de conversão dos sensores, uma vez que os dados enviados e coletados estão em formato hexadecimal (hex).

Serviço 01 [editar]							
PID (hex)	PID (dez)	Bytes de dados retornados	Descrição	Valor mínimo	Valor máximo	Unidades	Fórmula ^[a]
00	0	4	PIDs suportados [01 - 20]				Bit codificado [A7..D0] == [PID \$ 01..PID \$ 20] Veja abaixo
01	1	4	Monitorar o status desde que os DTCs foram limpos. (Inclui o status da lâmpada indicadora de mau funcionamento (MIL) e o número de DTCs.)				Bit codificado. Ver abaixo
02	2	2	Congelar DTC				
03	3	2	Status do sistema de combustível				Bit codificado. Ver abaixo
04	4	1	Carga calculada do motor	0	100	%	$\frac{100}{255} A$ (ou $\frac{A}{2.55}$)
05	5	1	Temperatura do líquido de arrefecimento do motor	-40	215	° C	$A - 40$
06	6	1	Redução de combustível de curto prazo				

Figura 4: Comandos PID do serviço 1

2.2 Framework

Framework de software é um conjunto de códigos genéricos, geralmente em forma de classes, implementada em alguma linguagem de programação e usadas para resolver um problema de um domínio específico. Atua em funcionalidades comuns entre várias aplicações, utilizando o conceito de reuso em engenharia de software, contribuindo redução de esforço e tempo necessário para desenvolver trabalhos, como aplicações que acessam banco de dados e validação de dados.

O framework utilizado foi o Yii2 baseado em PHP e embasada em uma arquitetura componentizada que é especialmente adequada para o desenvolvimento de aplicações de grande escala, como portais, fóruns, sistemas de gerenciamento, serviços web e *RESTful*.

Dentre as suas principais características, podem ser citadas o modelo de arquitetura *Model-View-Controller*, *Query builders* e *ActiveRecord* para transações com banco de dados de forma rápida e fácil, suporte a desenvolvimento de *API RESTful*, armazenamento de cache de várias camadas e arquitetura expansível por meio de extensões.

Seu código é aberto e desenvolvido por uma grande equipe que possui larga comunidade de profissionais constantemente contribuindo com desenvolvedores de aplicações, além de dispor de uma enorme quantidade de tutoriais e materiais de apoio na internet. Outro ponto importante é que a cada atualização, seu código fonte incorpora as últimas tendências e melhores práticas de desenvolvimento web (Safronov e Winesett, 2014).

2.3 Gateway

O *gateway* pode ser um computador com duas (ou mais) placas de rede, ou um dispositivo dedicado, utilizado para unir duas redes. Existem vários usos possíveis, desde interligar duas redes que utilizam protocolos diferentes, até compartilhar a conexão com a Internet entre várias estações (Zhang, 2010).

2.4 Comandos AT

Os comandos AT são linguagens de comandos para modems com uma série de cadeias curtas de texto que se combinam para emitir comandos completos (ELM327 *Datasheet*, 2018).

2.5 Ambientes Inteligentes (AmI)

Os Ambientes Inteligentes são descritos na literatura como uma visão do futuro na qual as tecnologias estarão sempre presentes no dia a dia das pessoas, automatizando tarefas corriqueiras por meio da utilização dos dados de sensores, aplicados a um determinado contexto (Yachir *et al.*, 2016).

Segundo Ramos, Augusto e Shapiro (2008), um ambiente inteligente requer um esforço considerável para desenvolver aplicações. A abordagem de AmI requer conceitos de Inteligência Artificial (IA) e isso segundo os autores levarão a evolução da IA para o próximo passo como a evolução dos ambientes inteligentes (Ramos, Juan e Augusto, 2008).

2.6 Considerações finais sobre o capítulo

Neste capítulo foram apresentados os principais conceitos e fundamentos sobre as tecnologias aplicadas nesta pesquisa. Dessa forma foi possível fazer um mapeamento do estado da arte e dos conceitos relacionados necessários para o embasamento teórico do trabalho e o direcionamento da investigação dos trabalhos relacionados que serão apresentados a seguir, no Capítulo 3.

Este Capítulo foi importante para compreender o problema que se pretende resolver a partir dos conceitos e ferramentas existentes em aplicações semelhantes. Assim, foi possível delimitar a busca de trabalhos relacionados, utilizando palavras chaves nos métodos de busca mais avançado.

3. Trabalhos relacionados

Neste capítulo é apresentada a revisão da literatura sobre os assuntos que envolvem o trabalho proposto, que são: ambientes inteligentes e comunicação veicular, ambos com foco no *gap* da pesquisa.

As pesquisas relacionadas a Ambientes Inteligentes (AmIs) foram focadas em dispositivos que conectam o ambiente com a internet a fim de automatizá-los. Neste caso, será apresentado um estudo sobre as aplicações desses sistemas e exemplos no mundo real, considerando como requisito todo tipo de comunicação que existe nesses ambientes, dentre os quais podem-se citar o Wi-Fi, Bluetooth, ZigBee, rádio frequência, entre outros. Para delimitar a pesquisa dos artigos, foram formuladas as questões na Tabela 3.

Tabela 3: Questões de delimitação da pesquisa

Questão	Descrição
1	Onde estão sendo aplicados os conceitos de ambientes inteligentes?
2	Quais dispositivos e/ou ferramentas são utilizados na comunicação desses ambientes?

As pesquisas relacionadas a comunicação veicular têm o objetivo de verificar o estado da arte, exemplos de aplicações no mundo real e as tecnologias utilizadas. As buscas pelos trabalhos relacionados nas bases literárias foram baseadas nas questões da Tabela 4.

Tabela 4: Questões de delimitação da pesquisa

Questão	Descrição
1	A evolução da comunicação veicular.
2	Aplicações existentes em comunicação veicular.
3	Quais recursos tecnológicos são usados nessa comunicação?
4	Quais dados trafegam nas redes veiculares?
5	Quais informações são retornadas para o usuário?
6	Qual objetivo das informações retornadas aos usuários?

A primeira fase da metodologia usada para o desenvolvimento da pesquisa consiste na identificação das palavras chaves das questões das tabelas acima. Segundo Krishnan e Kathpalia (2002) os artigos tem uma estrutura de organização do texto que ajuda a identificar os pontos principais de cada trabalho, o que facilita na inclusão ou exclusão do artigo nos trabalhos relacionados (Krishnan e Kathpalia, 2002).

A condução da pesquisa dos trabalhos relacionados foi feita utilizando as bases de dados do IEEE Xplore Digital Library, Web of Science, Engineering Village, ACM Digital Library e MDPI, as quais são bases renomadas nas áreas das engenharias elétrica, computação e telecomunicação. As estratégias de buscas com palavras chaves e filtros são mostradas na Tabela 5.

Tabela 5: Tabela de adequação das estratégias de busca a cada uma das bases de dados

Fonte¹	Ferramenta de pesquisa²	Composição da pesquisa³	Filtros adicionais
IEEE Xplore	Avançada	(((((("Abstract":OBD) OR "Abstract":on board diagnostic) AND "Abstract":vehicle) OR automotive) AND datasets) AND application) AND machine learning)	Faixa de ano 2010 até 2017 Journals and magazines
Engineering Village	Básica	<u>(((((vehicle) WN All fields) OR ((automotive) WN All fields)) AND ((smart) WN All fields)) AND ((on board diagnostic) WN All fields)) OR ((OBD) WN All fields) AND ((datasets) WN All fields))</u>	Faixa de ano 2010 até 2017.
Web of Science	Avançada	TS = (((vehicle) OR (automotive)) AND (smart) AND ((OBD) OR (on board diagnostic)) OR (Machine learnig))	Faixa de ano 2010 até 2017.
ACM Digital Library	Avançada	acmdlTitle:(ambient intelligence)	Faixa de ano 2011 até 2017. Classificado por relevância
MDPI	Básica	OBD	Faixa de ano 2014 até 2019.

¹ Base de dados utilizada

² Tipo de pesquisa (básica e avançada)

³ Tradução da estratégia de busca para o tipo de pesquisa e para a base utilizada

As pesquisas nessas bases foram realizadas no período de março de 2017 até maio de 2019 e as publicações foram classificadas de acordo com os padrões estabelecidos nos critérios de inclusão e exclusão.

Foram aplicados dois filtros nas publicações retornadas, nos quais o primeiro filtro consistiu na leitura dos resumos e palavras chaves das publicações para em seguida analisá-las de acordo com os critérios de exclusão e inclusão listados a seguir.

Critérios de Exclusão (CE) do primeiro filtro:

- CE1-01: Não serão selecionadas as publicações antes de 2010;
- CE1-02: Os artigos repetidos foram descartados;
- CE1-03: Não serão selecionadas publicações em que a contextualização das palavras-chave utilizadas não demonstra que o trabalho cita a abordagem de AmI ou comunicação veicular;
- CE1-04: Serão priorizadas publicações em revistas.

Critérios de Inclusão (CI) primeiro filtro:

- CII-01: Serão selecionados artigos que tenham a palavra chave no título;
- CII-02: Podem ser selecionadas publicações que não são de revistas, mas que em seu contexto as palavras-chave utilizadas cite uma abordagem do uso AmI nas aplicações de automação residencial ou em outros ambientes;
- CII-03: Podem ser selecionadas publicações em que o contexto das palavras-chave utilizadas cite as abordagens de comunicações veiculares.

O segundo filtro consistiu na leitura mais aprofundada dos artigos classificados no primeiro filtro. O objetivo do segundo filtro foi identificar os artigos que se relacionavam com AmI e comunicação veicular, observando em ambos os assuntos as aplicações e arquiteturas.

Critérios de Exclusão:

- CE2-01: Não serão selecionadas publicações em que a contextualização das palavras-chave utilizadas não demonstra que o trabalho cita a abordagem de AmI na automação de ambientes residenciais ou outros ambientes;
- CE2-02: Não serão selecionadas publicações em que a contextualização das palavras-chave utilizadas não demonstra que o trabalho cita abordagens ou aplicações de comunicação veicular;
- CE2-03: Serão selecionados trabalhos observando o ano da publicação, as figuras e diagramas.

A base do Web of Science e Engineering Village indexam artigos do ACM Digital Library e do IEEE Xplore, portanto, alguns artigos retornados foram repetidos nas buscas. As pesquisas dos assuntos foram realizadas modo básico e avançado aplicando filtros de ano e relevância (quantidade de citação e downloads), os resultados da pesquisa e dos filtros podem ser visualizados no gráfico da Figura 5.

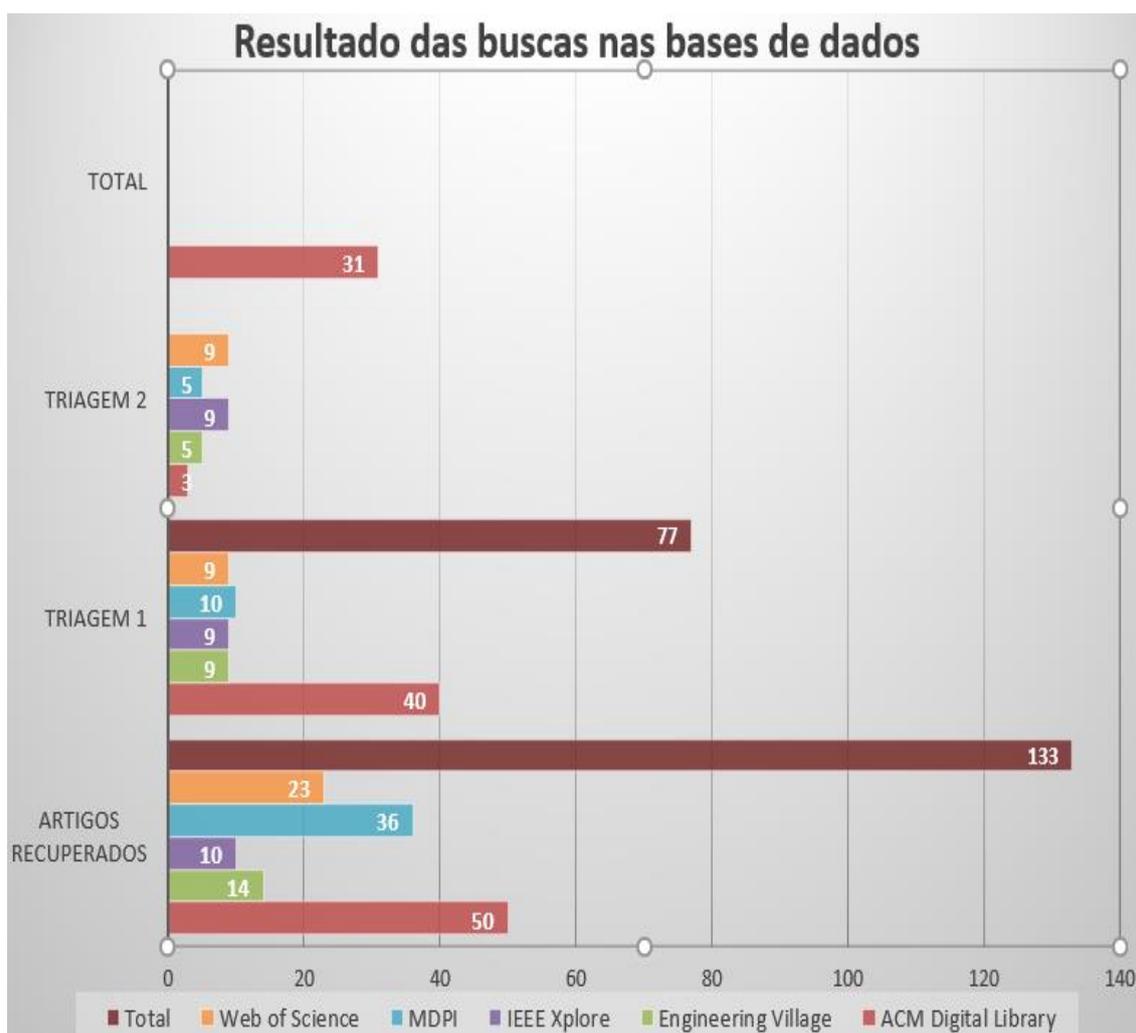


Figura 5: Resultado geral de publicação referentes ao tema

Pode-se observar na Figura 5 que o total de artigos retornados a partir das buscas pelas palavras-chave foi de 133 publicações das cinco bases representadas pelas cores na legenda da figura.

Após as triagens foram recuperados 31 artigos dos quais 16 são sobre AmI e 15 sobre comunicação veicular, que serão sintetizados apontando as principais características nas subseções a seguir.

3.1 Trabalhos Relacionados sobre Ambientes Inteligentes

Sampaio, Reis e Rodrigues (2012) apresentaram uma pesquisa que realiza uma análise comparativa de alguns dos projetos relacionados a Ambientes Inteligentes (AmI), com o objetivo de compreender as necessidades atuais, os dispositivos e os principais

resultados. Nesse trabalho foram selecionados 9 projetos, dentre os quais o *IRoom* teve maior relevância, pois utilizou diversos recursos do ambiente, como: sensores, atuadores e redes dedicadas no ambiente. A conclusão dos autores foi que a maioria dos projetos não exploram o potencial de perfis humanos no contexto da adaptação do ambiente e nem o potencial total que esses ambientes oferecem, assim alguns recursos são subutilizados (Sampaio, Reis e Rodrigues, 2012). A observação que os autores fizeram sobre a subutilização dos recursos foi comprovada nos trabalhos relacionados. Isso abriu margens para utilizar melhor esses recursos, inserindo novas funcionalidades e integrar ferramentas ao ambiente, o qual pode proporcionar uma interação humano-computador mais invisível, aplicando dessa forma um dos conceitos fundamentais de AmI, a computação ubíqua.

Moseley (2017) propôs a junção de agentes de comunicação de um ambiente e reconhecimentos de padrões para tornar o ambiente inteligente, pois segundo o autor, os seres humanos vivem e trabalham em um ambiente conectado e sem inteligência, uma vez que, tais áreas possuem comunicação entre dispositivos com uma carga relevante de informação, mas não contribuem para o principal objetivo de um AmI (Moseley, 2017). Esse trabalho mostrou que os AmIs possuem recursos computacionais que permitem a utilização de técnicas de análise de dados, as quais demandam maior poder de processamento, tornando o ambiente mais completo e funcional.

Obukata, Oda e Barolli propuseram um sistema capaz de monitorar a casa, utilizando vários dispositivos de computação onipresentes, nos quais os ambientes físicos interagem de forma inteligente e discreta com as pessoas. Alguns exemplos de aplicação desses ambientes são nas residências, escritórios, salas de reunião, escolas, hospitais, centros de controle, veículos, atrações turísticas, lojas e instalações esportivas. Nesse sistema foi feito um protótipo com várias camadas de comunicação embarcadas em um computador de placa única (Obukata, Oda e Barolli, 2016). Nesse trabalho foi implementado um sistema simples com poucos recursos e uma arquitetura bem definida, a qual utilizou o Raspberry Pi para fazer a comunicação dos nós dentro da casa.

Frey (2013) pesquisou sobre as aplicações de sistemas embarcados, ambientes assistidos e ambientes inteligentes. O autor pontuou que mesmo inserindo sensores e atuadores em um ambiente com regras de automação, não o torna inteligente. Para que isso ocorra, o sistema deve aprender as rotinas utilizando algoritmos que utilizam técnicas de aprendizagem de máquina, além de se adaptar aos comportamentos do usuário. Como prova de conceito, foi desenvolvido um sistema capaz de prever e rastrear as atividades da vida diária em um ambiente preparado (Frey, 2013). Esse trabalho apresentou uma arquitetura

capaz de conectar dispositivos que utilizam protocolos de comunicação distintos dentro de uma casa. Outro componente utilizado é o *gateway*, no qual foram implementados protocolos de comunicação que possibilitaram a interoperabilidade dos dispositivos. As tecnologias utilizadas foram: HTTP, JSON, além das conexões ZigBee, Wi-Fi, entre outros.

Montori *et al.* (2018) propuseram uma plataforma capaz de gerenciar dispositivos domésticos de diferentes redes e tecnologias. Nessa plataforma foi desenvolvido um roteador responsável pela interoperabilidade chamado Routex, o qual é um *gateway* com o objetivo de controlar tudo através de um aplicativo de *smartphone* (Montori *et al.*, 2018). Esse trabalho é uma aplicação de AmI e mostrou as tecnologias utilizadas na implementação do roteador, o qual foi um RaspBerry Pi 3 utilizando comunicação Wi-Fi, ZigBee, Rádio frequência, Bluetooth e protocolos de comunicação, como Machine to machine (M2M), HTTP, Redes de sensores Globais, entre outros. Além disso, foram mostrados diagramas de atividades exemplificando a dinâmica das mensagens.

Putthapipat, Woralert e Sirinimnuankul (2018) propuseram um *gateway* para controlar a casa por comando de voz utilizando palavras chaves. O dispositivo utilizado para o *gateway*, foi um *Raspberry Pi* com bibliotecas instaladas responsáveis por facilitar o trabalho com reconhecimento de comandos por voz (Putthapipat, Woralert e Sirinimnuankul, 2018). Esse trabalho mostrou mais uma aplicação de AmI utilizando o RaspBerry Pi 3, no qual foram instalados receptores e atuadores a fim de demonstrar a funcionalidade do protótipo. Na implementação do *software* foram utilizados o Node.js e Python, além de funções API para reconhecer os comandos de voz.

Vivek e Sunil (2016) propuseram um sistema de automação residencial no qual foi desenvolvido um *gateway* para integrar os principais dispositivos de uma casa utilizando tecnologias sem fio, tais como Wi-Fi e ZigBee. O sistema consiste em uma tela com interação humano-computador sensível ao toque, sensores e atuadores associados aos módulos ZigBee, como: sensor de temperatura, sensor de presença e sensor da porta. Os atuadores foram instalados nos interruptores das luzes, na TV e na ventilação. A comunicação Wi-Fi foi utilizada em dispositivos de monitoramento da corrente para gerenciar e criar relatórios de consumo de energia, enviando-os para o servidor (Vivek e Sunil, 2016). Esse trabalho utilizou tecnologias de comunicação sem fio, Wi-Fi e ZigBee, para conectar os dispositivos com a internet. Os dispositivos utilizados para possibilitar essa conexão foram: um processador Arm A7 com um módulo ZigBee, o módulo Wi-Fi ESP8266 com sensor de corrente e um roteador Wi-Fi para conectar via cabo e via sem fio.

Seif e El-Saber (2017) propuseram uma arquitetura genérica que busca resolver os

problemas de interoperabilidade entre os dispositivos conectados à internet, dividindo hierarquicamente em camadas de processamento, os quais são computação *dew*, computação *fog* e computação *Cloud*, respectivamente, base, meio e topo. Essas camadas de processamento envolvem redes de curto alcance e diferentes protocolos de comunicação como: *Bluetooth*, Wi-Fi, *Zigbee*, *zwave*, MQTT, DDS e CoAP (Seif e El-Saber, 2017). Esse trabalho mostrou a classificação de locais que podem ser realizados processamentos dos dados de acordo com a rede utilizada. Esses conceitos são necessários para direcionar e avaliar quais tecnologias de comunicação devem ser empregadas de acordo com a aplicação, e assim, escolher uma ou mais para implementar o sistema.

Silva *et al.* (2016) propuseram um sistema onipresente e inteligente, capaz de monitorar a tomada de medicamentos e identificar se o paciente está atendendo aos requisitos prescritos pelo médico. O sistema é responsável pela coleta de dados de um armário de remédios, sensores e dispositivos eletrônicos de um ambiente. Possui um *gateway* residencial responsável pela integração do sistema, o qual divulga mensagens sobre as tomadas de remédio nas redes sociais, SMS e aparelhos eletrônicos de consumo, como: TV digital, *smartphone* e *tablets*. O sistema funciona sem interferência humana e com o objetivo de ajudar o paciente a tomar a medicação no tempo correto (Silva *et al.*, 2016). Esse trabalho é mais um exemplo de AmI aplicado na casa e apresentou uma arquitetura formada respectivamente pelos sensores da casa, processamento, técnicas de aprendizagem de máquina e distribuição da informação nos dispositivos conectados a esse ambiente, os quais enviavam mensagens pelas redes sociais quando o usuário tomava ou não os medicamentos.

Seabra, Costa e Lucena (2016) propuseram um sistema inteligente que monitora o comportamento das magnitudes elétricas dos eletrodomésticos em tempo real, o qual é capaz de analisar os dados coletados, detectar possíveis falhas e alertá-las ao usuário. O trabalho utiliza um microprocessador que coleta as grandezas elétricas, mecânicas e os envia para um servidor, o qual processa, identifica falhas instantâneas e prever a necessidade de manutenção preventiva. A interação humano-computador é realizada pela TV digital, dispositivos móveis ou computadores convencionais que alertam os usuários sobre as intervenções necessárias (Seabra, Costa e Lucena, 2016). Esse trabalho é uma aplicação de AmI na casa, pois tem o objetivo de monitorar aparelhos domésticos. Utilizaram-se dispositivos sensores no aparelho em conjunto com um microprocessador que coletava os dados e enviava para um servidor na nuvem. Na nuvem foram aplicadas técnicas de inteligência artificial para detectar falhas no aparelho, além de disponibilizar os dados das falhas nos outros dispositivos conectados à rede da casa. A aplicação desse trabalho visou notificar o usuário sobre possíveis problemas

no aparelho monitorado pelo sistema.

Soultanopoulos *et al.* (2016) apresentaram o uso de sensores vestíveis e a sua conectividade à internet, mostrando os benefícios de armazenar os dados para serem utilizados de maneira inteligente por aplicativos de monitoramento no domínio de assistência médica. Os autores propuseram um *gateway* capaz de se conectar a esses dispositivos vestíveis, coletar e enviar em tempo real os dados para a nuvem. Esses *gateways* são *smartphones*, dos quais são utilizadas tecnologias como o Bluetooth e rede móvel para realizar esses procedimentos (Soultanopoulos *et al.*, 2016). Esse trabalho apresentou uma arquitetura com camadas detalhadas, que são formadas respectivamente por: usuário com sensor vestível, IoT *gateway* (*smartphone*), servidor na nuvem e a interface do sistema. Foram utilizadas tecnologias conhecidas, como: Bluetooth, APIs, HTTP, JSON e *Web Server*.

Bhargava, Mcmanus e Ivanov (2017) propuseram um sistema capaz de monitorar e localizar pessoas no contexto de ambientes assistidos utilizando computação *fog* e redes de sensores sem fio. A arquitetura do sistema é composta por um dispositivo sensor vestível e um *gateway*. O sistema é direcionado para pessoas idosas e tem o objetivo de detectar movimentos bruscos ou de queda. O *gateway* do sistema é um Raspberry Pi que coleta dados do dispositivo vestível e possibilita o envio desses dados para a nuvem (Bhargava, Mcmanus e Ivanov, 2017). Esse trabalho é um exemplo de aplicação de AmI em uma casa. A arquitetura é formada por um dispositivo vestível, o *gateway* e o servidor na nuvem. A comunicação entre os dispositivos é realizada por rádio frequência no dispositivo vestível e Wi-Fi para o *gateway* se conectar à internet e transmitir os dados.

Como forma de sintetizar os estudos referentes a ambientes inteligentes, foi criada uma tabela que apresentará resumidamente pontos dos trabalhos. As Tabelas 6 e 7 apresentam as sínteses e análises técnicas indicando os recursos dos trabalhos relacionados, facilitando uma visão resumida e generalizada das tecnologias aplicadas.

Tabela 6: Síntese dos trabalhos sobre AmI

Referências	Aplicações	Análise técnica
Sampaio <i>et al.</i> , 2012	Projetos com AmI com a utilização sensores e atuadores para auxiliar o usuário	Conceituou o AmI
Corno e De Russis, 2017	Abordagens de ensino para matéria de AmI na universidade	Listou os conceitos de AmI
Moseley, 2017	Adicionou mais funcionalidades no AmI, reconhecimento de padrões a fim de auxiliar o ser humano	Criou um robô com Arduino
Obukata <i>et.al.</i> , 2016	Arquitetura que utilizou um dispositivo que conecta os dispositivos do AmI com a nuvem	<i>Gateway</i> : RaspBerry Pi
Frey, 2013	Dispositivo que conecta vários dispositivos com protocolos de comunicação diferentes	Tipos de dados: JSON Protocolos de comunicação: HTTP e ZigBee Tecnologia de comunicação: Wi-Fi
Montori <i>et al.</i> , 2017	<i>Gateway</i> para comunicação de dispositivos com diferentes protocolos	<i>Gateway</i> : RaspBerry Pi 3 Tecnologia de comunicação: Wi-Fi, Radio frequência, Bluetooth Protocolos de comunicação: Machine to machine (M2M), HTTP e ZibBee
Putthapipat <i>et al.</i> , 2018	<i>Gateway</i> por comando de voz para comunicação de dispositivos com diferentes protocolos	<i>Gateway</i> : RaspBerry Pi 3 Ferramentas: Node.js, Python e APIs
Vivek e Sunil, 2016	<i>Gateway</i> para comunicação de dispositivos com diferentes protocolos em sistemas de automação antigos	<i>Gateway</i> : Processador Arm A7 Protocolo de comunicação: ZigBee Tecnologia de comunicação: Wi-Fi
Seif e El-Saber, 2017	Arquitetura genérica que conectou vários dispositivos em camadas de acordo com o alcance desses dispositivos	Conceitos sobre os tipos de processamento: nuvem, intermediário e local
Campbell <i>et al.</i> , 2016	<i>Gateway</i> capaz de conectar vários outros dispositivos fazendo o gerenciamento por meio de links URL	Foram utilizadas formas diferentes de conexão de vários dispositivos na rede passando por um nó central
Rahmani e Kanter, 2017	Simulação de uma rede imersiva utilizando abordagens de comunicação 5G	Foram apresentados conceitos

Tabela 7: Continuação síntese dos trabalhos sobre Aml

Referências	Aplicações	Análise técnica
Silva <i>et al.</i> , 2016	Integração de dispositivos com o ambiente por meio da internet	Ambiente notifica o usuário por meio das redes sociais e mensagens de alertas no <i>smartphone</i>
Seabra, <i>et al.</i> , 2016	Monitoramento de um equipamento com acesso aos dados gerados por meio da internet	Ambiente possibilita o monitoramento dos equipamentos da casa por meio de site do servidor na nuvem
Soultanopoulos <i>et al.</i> , 2016	Integração de tecnologias de comunicação para coleta, processamento e transmissão de dados para a nuvem	<i>Gateway: smartphone</i> Tecnologia de comunicação: Bluetooth Tipos de dados: JSON Ferramentas: APIs e HTTP
Bhargava e Ivanov, 2018	Integração de tecnologias de comunicação e técnicas de inteligência artificial utilizando um dispositivo como nó central	<i>Gateway: smartphone</i> Tecnologia de comunicação: Wi-Fi

Após a análise e a montagem das tabelas de síntese foi possível observar que as aplicações dos conceitos de ambientes inteligentes direcionam as aplicações para os seguintes tópicos:

- Aplicar nas casas para automatizar tarefas do dia-a-dia;
- Monitorar aparelhos domésticos utilizando abordagens de processamento de dados para detectar anomalias no padrão de funcionamento;
- Ambientes assistidos direcionados para pessoas idosas.

Sobre os dispositivos e/ou ferramentas utilizadas na comunicação, observou-se que o mais usual foi o *gateway*, o qual pode ser empregado diferentes dispositivos ou equipamentos. Nos trabalhos foram encontrados: RaspBerry Pi, roteador, processador com módulos e computadores de mesa.

3.2 Trabalhos Relacionados sobre comunicação veicular

Kiening *et al.* (2013) apresentaram aspectos sobre a segurança da comunicação veículo para tudo (*vehicle to everything* - V2X). Na última década, a indústria automotiva, governos e pesquisadores investiram intensamente em esforços para estabelecer a base V2X, com o objetivo de melhorar a segurança no trânsito e a eficiência do tráfego. Nesse artigo, foram analisados os requisitos de segurança da plataforma dos sistemas V2X, os diferentes

níveis de garantia de confiança (*Trust Assurance Levels* - TAL) e uma estrutura de certificação para apoiar o estabelecimento de confiança entre os usuários da comunicação V2X (Kiening *et al.*, 2013). Esse trabalho fez uma breve apresentação sobre a segurança dos dados nas redes veiculares e apresentou conceitos de segurança da informação, com a motivação que os veículos estão cada vez mais utilizando tecnologia de ponta em comunicação e controle.

Ang, Member e Seng (2019) apresentaram em sua pesquisa o estado da arte da tecnologia utilizada na Internet dos Veículos (*Internet of Vehicles* - IoV), o qual é a convergência da Internet móvel e da Internet das Coisas (*Internet of Things* - IoT). No IoV, os veículos funcionam como nós inteligentes móveis em movimento ou objetos dentro da rede de sensores. Os autores realizaram uma revisão abrangente dos paradigmas atuais e emergentes da IoV. Também foram abordados os modelos de comunicação com ênfase na implantação em cidades inteligentes, pois muitos autores têm focado no IoV como: aplicações para transporte inteligente, segurança do motorista, eficiência do tráfego e entretenimento (Ang, Member e Seng, 2019). Esse trabalho apresentou uma revisão bibliográfica sobre redes veiculares listando todos os tipos de comunicação, como: veículo para veículo, serviço, casa, dispositivo, infraestrutura, entre outros. Além de apresentar aplicações da IoV, arquiteturas e *Big Datas* relacionados com cidades inteligentes.

Silva *et al.* (2019) propuseram uma plataforma de *feedback* para clientes na indústria de veículos no contexto da Indústria 4.0. Essa plataforma é capaz de coletar e analisar os sensores disponíveis nos veículos. Os dados são coletados através de um scanner OBD-II e o RaspBerry Pi 3 com localizador GPS e módulo de internet móvel. O objetivo dessa plataforma é auxiliar a gestão, prevenção e mitigação de diferentes problemas veiculares. Como prova de conceito, foi realizada uma avaliação intercontinental entre Brasil e Itália que mostrou a viabilidade da plataforma e o potencial de uso para melhorar o processo de fabricação de veículos (Silva *et al.*, 2019). Esse trabalho apresentou uma revisão bibliográfica sobre redes veiculares focando na coleta dos dados dos veículos, com o intuito de serem analisados e montar um histórico de uso do veículo. Na construção da plataforma foram utilizados recursos tecnológicos, como: adaptador OBD-II, aplicativos de *smartphone*, servidor na nuvem, APIs, JSON, Wi-Fi, Bluetooth, 3G e 4G.

Atzori *et al.* (2018) introduziram os paradigmas sobre a Internet dos Veículos (IoV), o qual considera que os veículos estão sempre conectados à internet consumindo serviços e enviando dados. A exploração das noções de redes sociais na IoV trouxe a definição do paradigma Social IoV (SIoV), ou seja, uma rede social em que cada veículo é capaz de

estabelecer relações sociais de forma autônoma com outros veículos ou infraestrutura rodoviária. Os autores propuseram uma implementação da SIOV em uma plataforma de IoT baseada em nuvem, para gerenciar a atividade social dos veículos. Dentre as contribuições, a principal foi a solução eficaz de baixo custo e flexível para a unidade de bordo (*On Board Unit* - OBU) que permitiu que todos os veículos participassem do SIOV (Atzori *et al.*, 2018). Esse trabalho fez um breve resumo sobre as redes veiculares e apresentou as tendências de acordo com trabalhos similares. Foi apresentada a arquitetura em detalhes, na qual mostra que o sistema proposto é capaz de adequar o veículo para participar de uma rede social de veículos. A instalação do OBU é indispensável para fazer parte da SIOV, pois por ele é feita a coleta de dados dos sensores do veículo, localização global e acesso à internet. O OBU é formado pela RaspBerry Pi 3 com adaptador 3G e módulo *Global Positioning System* (GPS) e coleta os os dados do veículo por meio do adaptador OBD-II com comunicação Bluetooth.

Shi *et al.* (2016) propuseram um sistema capaz de reprogramar a ECU por meio de um sistema utilizando a interface OBD-II do veículo. Para a implementação, foi utilizado o sistema *Vehicle Communication Interface* (VCI), adaptador OBD-II, *smartphone* com aplicativo e servidor remoto. Os testes funcionais foram realizados com mais de 3000 usuários conectados ao servidor (Shi *et al.*, 2016). Esse trabalho apresentou uma arquitetura na qual foi desenvolvido um dispositivo para ficar instalado no veículo, a fim de coletar dados por meio do adaptador OBD-II. Esse dispositivo se conecta ao *smartphone* para atualizar a ECU. O *smartphone* se comunica via 3G com o servidor na nuvem que está ligado ao sistema de gerenciamento de atualização das ECUs.

Amarasinghe *et al.* (2015) apresentaram um sistema capaz de monitorar o comportamento do motorista, analisar a viagem e diagnosticar o veículo. O *smartphone* com aplicativo foi utilizado para se conectar ao adaptador OBD-II e coletar dados do veículo. O *smartphone* é responsável pelo envio e recebimento de dados processadas do servidor remoto, ambos com Processador Complexo de Eventos, uma ferramenta responsável por detectar anomalias nos dados. O sistema é capaz de alertar o motorista nos seguintes aspectos: situações de risco na estrada, comportamento inadequado do motorista e status do veículo (Amarasinghe *et al.*, 2015). Esse sistema possui uma arquitetura formada pelo *smartphone* e servidor na nuvem. O sistema usou o adaptador OBD-II, comunicação HTTP, aplicações web e algoritmos de aprendizagem de máquina para detectar padrões.

Magana e Munoz-Organero (2016) apresentaram um assistente de direção que faz recomendações ao motorista, a fim de reduzir o consumo de combustível. Os aconselhamentos são feitos através de um aplicativo de *smartphone*, o qual faz a coleta de

dados por meio do adaptador OBD-II. O aplicativo detecta a forma de direção, incidentes nas estradas e utiliza recursos de *gameficação* para pontuar ou penalizar os motoristas. A solução proposta utiliza o adaptador OBD-II com conexão *Bluetooth*, *smartphone* e aplicativo, posicionamento por meio do GPS, câmera a bordo e acesso à internet. Os testes foram realizados em três modelos de veículos, os quais foram dirigidos por vinte e cinco motoristas, em um total de 2250 testes de condução. O resultado foi a redução média do consumo de combustível em aproximadamente 11,04% (Magana e Munoz-Organero, 2016). Esse sistema utilizou diferentes tecnologias para extrair e analisar os dados. A arquitetura foi mostrada em alto nível, mas é possível identificar que o sistema possui três camadas, na qual a primeira é o veículo, a segunda é o *smartphone* com o adaptador OBD-II e por último a nuvem, na qual são aplicados os processamentos. As tecnologias usadas na comunicação foram Bluetooth e Wi-Fi, além de processamento de imagens utilizando o a câmera do *smartphone*.

Wallace *et al.* (2015) utilizaram informações do projeto *Candrive*, o qual tem o propósito de coletar e armazenar dados dos veículos dirigidos por pessoas idosas. Esses foram coletados por um dispositivo instalado no veículo através da porta OBD-II. O projeto inclui 256 motoristas idosos no Canadá e é um exemplo de estudo na coleta de grandes quantidades de dados. O objetivo do trabalho é propor soluções de privacidade dos dados de direção, utilizando técnicas de anonimato para qualquer combinação de parâmetros, pois são coletados dados de veículos particulares e de localização pelo GPS dos motoristas integrantes do projeto. Esse trabalho discute também, métodos para adequar um grande conjunto de dados brutos em dados prontos para serem analisados, filtrando e organizando em formatos de arquivos compactos. Além disso, utiliza mecanismo de fusão que mescla dados dos sensores do automóvel e localização para incluir informações da estrada e clima. Para extrair os dados, é necessário retirar o dispositivo de coleta do veículo e realizar a transferência manualmente para computadores que realizam a análise. Os dispositivos utilizados foram um localizador GPS, um adaptador OBD-II e dispositivos de armazenamento de dados. Os testes foram realizados com 256 motoristas idosos através de coleta de dados que duraram entre 4 e 6 anos (Wallace *et al.*, 2015). Esse trabalho apresentou uma arquitetura simples, formada por um dispositivo que coleta dados em conjunto com o localizador GPS. Foram utilizadas técnicas para os dados ficarem anônimos, além da seleção de sensores que devem ser coletados.

Chen, Pan e Lu (2015) propuseram um sistema de assistência ao motorista que utilizou métodos de análise do comportamento de condução baseados nos dados dos sensores do veículo. Os algoritmos AdaBoost foram utilizados no sistema, os quais utilizam

aprendizagem de máquina de classificação que coleta dados por meio do adaptador OBD-II, como: velocidade, rotação do motor, posição do acelerador e a carga calculada do motor. Em seguida, o método faz uso dos algoritmos para criar um modelo de classificação de comportamento de condução. Os resultados experimentais mostraram que o comportamento de condução pode alcançar precisão média de 99,8% em várias simulações de direção (Chen, Pan e Lu, 2015). Esse trabalho também fez um breve resumo sobre os códigos de falhas e sensores dos adaptadores OBD-II. O sistema foi modelado com a ferramenta MATLAB e foi utilizado um simulador para gerar os valores das variáveis do OBD-II.

Siegel *et al.* (2014) propuseram um sistema que alerta o motorista para a troca de óleo do motor. Para a concepção do sistema, foi necessário um adaptador OBD-II, localização via GPS, além de acesso à rede móvel de telefonia para enviar os dados para a nuvem. O dispositivo responsável pela coleta dos dados foi o *CARduino*, o qual também se comunica com o servidor que possui funções *API RESTful*, a fim de realizar a análise dos dados. Nos testes realizados, foi constatado que a taxa de aumento da temperatura do líquido arrefecedor do motor pode servir como um parâmetro para indicar a vida útil do óleo do motor (Siegel *et al.*, 2014). A comunicação aconteceu por *Groupe Special Modibe* (GSM), no qual são enviados o posicionamento global e os dados no formato JSON.

Malekian *et al.* (2017) propuseram um sistema gerenciador de frota de veículos, na qual foi construído um dispositivo que se conectava diretamente à porta OBD-II. O objetivo desse sistema era medir a velocidade, distância e o consumo de combustível dos veículos para fins de rastreamento e análise. Foi projetado um leitor OBD-II que mede o fluxo de ar e velocidade, que a partir desses dois parâmetros a distância e o consumo de combustível também são calculados. Esses dados foram transmitidos via Wi-Fi para um servidor remoto. O sistema possui rastreamento via GPS para determinar a localização, um sistema de gerenciamento de banco de dados e uma interface gráfica para o usuário (Malekian *et al.*, 2017). Nesse trabalho o dispositivo desenvolvido foi formado por um módulo GPS, que processa a informação coletada diretamente da ECU, uma interface de comunicação com a porta OBD-II e comunicação sem fio que era responsável por transmitir os dados para um servidor.

Meseguer *et al.* (2017) propuseram uma aplicação móvel chamada *DrivingStyles*, com o propósito de avaliar o estilo de direção e caracterização do consumo de combustível. A arquitetura da aplicação adotou técnicas de mineração de dados e redes neurais para analisar e gerar uma classificação de estilos de direção e consumo de combustível com base na caracterização do motorista. Foi implementado um algoritmo capaz de caracterizar o grau

de agressividade de condução do motorista. Foi desenvolvido também, uma metodologia para calcular o consumo e o impacto ambiental de veículos movidos à gasolina e diesel coletados por meio da porta OBD-II. A conclusão do trabalho apresentou o impacto do estilo de condução no consumo de combustível e a correlação com as emissões de gases de efeito estufa gerados pelos veículos (Meseguer *et al.*, 2017). Nesse trabalho foram utilizados um adaptador OBD-II *Bluetooth, smartphone* com aplicativo responsável por enviar dados ao servidor remoto, a fim de realizar o processamento com técnicas de aprendizagem de máquinas.

Prytz (2014) investigou métodos não supervisionados e supervisionados para a manutenção de veículos, utilizando métodos orientados por dados em grandes quantidades, dados de fluxo, dados a bordo ou dados de históricos e agregados de bancos de dados off-board. Os dados foram enviados para um sistema externo que os analisava quanto aos desvios que eram posteriormente associados ao histórico de reparos e formavam uma base de conhecimento que foram ou puderam ser usadas para prever falhas futuras em outros veículos que mostrassem os mesmos comportamentos. Uma segunda abordagem descrita foi uma classificação supervisionada baseada em estatísticas de veículos, na qual o histórico de reparos foi usado para rotular as estatísticas de uso. Um classificador foi treinado para aprender padrões nos dados de uso que precedem reparos específicos. Assim, puderam ser utilizados para a manutenção do veículo. Esse método foi demonstrado para falhas do compressor de ar do sistema de frenagem e baseado em estatísticas de uso de veículos da AB Volvo (Prytz, 2014). Esse trabalho utilizou técnicas de inteligência artificial para detectar padrões de peças prestes a quebrar de acordo com observações passadas na qual há o desgaste e a quebra do componente do veículo. Foram utilizadas tecnologias como o padrão OBD-II, comunicação via satélite e telefonia móvel para transmitir os dados dos sensores coletados.

Morabito *et al.* (2017) propuseram uma plataforma que implementa aplicativos configuráveis para veículos inteligentes com o intuito de gerenciar componentes como multimídia, dados de funcionamento do veículo, câmeras de bordo, entre outros. Para demonstrar a aplicação dessa plataforma, foi utilizado o *Raspberry Pi 3*, no qual foram realizados testes visando garantir que a segurança da plataforma é confiável (Morabito *et al.*, 2017). Esse trabalho fez uma pesquisa detalhada sobre o *Raspberry Pi 3*, apontando características, como: processamento, custo, sistema operacional, linguagens de programação utilizadas, entre outros.

Como forma de sintetizar os estudos referentes à comunicação veicular, foi criada uma tabela que apresenta os resultados dos filtros de buscas. As Tabelas 8 e 9 apresentam as

sínteses e análises técnicas dos trabalhos relacionados, facilitando uma visão resumida e generalizada das tecnologias aplicadas.

Tabela 8: Tabela de comparação sobre comunicação veicular

Referências	Aplicações	Análise técnica / tecnologias usadas
Kiening et al., 2013	Segurança dos dados que circulam entre os veículos e objetos conectados à internet	Estado da arte sobre segurança de dados veiculares
Naik et al., 2019	Revisão bibliográfica sobre redes veiculares	Estado da arte sobre redes veiculares
Ang et al., 2018	Revisão bibliográfica sobre redes veiculares, além de apresentar aplicações detalhadas sobre o IoV	Exemplos de aplicações da IoV
Silva et al., 2018	Construção de um dispositivo para coleta de dados do veículo, a fim de formar um histórico da utilização veicular	Dispositivo: Adaptador OBD-II Ferramentas: APIs <i>Gateway</i> : Smartphone com aplicativo Tecnologia de comunicação: Wi-Fi, Bluetooth, 3G e 4G Tipos de dados: JSON
Atzori et al., 2018	Construção de um dispositivo que habilita o veículo para a rede social IoV	Dispositivo: adaptador OBD-II e adaptador modem 3G <i>Gateway</i> : RaspBerry Pi 3 Tecnologia de localização: GPS Tecnologia de comunicação: Wi-Fi, Bluetooth e 3G
Shi et al., 2016	Atualizar ECU do veículo por meio do <i>smartphone</i>	Dispositivo: adaptador OBD-II <i>Gateway</i> : VCI e <i>smartphone</i> Tecnologia de comunicação: 3G
Amarasinghe et al., 2015	Define o comportamento do motorista utilizando o <i>smartphone</i>	Dispositivo: adaptador OBD-II <i>Gateway</i> : <i>smartphone</i> Protocolo de comunicação: HTTP Tipo de comunicação: 3G/4G
Magaña et al., 2016	Assistente de direção	Dispositivo: adaptador OBD-II <i>Gateway</i> : <i>smartphone</i> Tecnologia de comunicação: Bluetooth e Wi-Fi

Tabela 9: Continuação da tabela de comparação sobre comunicação veicular

Referências	Aplicações	Análise técnica / tecnologias usadas
Wallace <i>et al.</i> , 2015	Monitoramento do veículo de pessoas idosas	Dispositivo: adaptador OBD-II Gateway: indefinido Tecnologia de localização: GPS
Chen <i>et al.</i> , 2015	Simulação dos dados do carro com um software	Dispositivo: adaptador OBD-II Ferramentas: Simulador e MATLAB
Siegel <i>et al.</i> , 2014	Sistema que informa a necessidade da troca do óleo do motor pelo parâmetro do sensor de temperatura do motor	Dispositivo: adaptador OBD-II Gateway: CARduino Ferramentas: APIs Tecnologia de comunicação: GSM Tipos de dados: JSON
Malekian <i>et al.</i> , 2017	Sistema de gerenciamento de frota veicular	Dispositivo: indefinido Tecnologia do veículo: conector OBD-II Tecnologia de comunicação: Wi-Fi Tecnologia de localização: GPS
Meseguer <i>et al.</i> , 2017	Sistema de detecção do estilo de direção do motorista	Dispositivo: adaptador OBD-II Gateway: <i>smartphone</i> Tecnologias de comunicação: Bluetooth, Wi-Fi e 3G
Martínez <i>et al.</i> , 2015	Monitoramento do motorista com um veículo dedicado com sensores instalados	Dispositivo: adaptador OBD-II
Prytz, 2014	Detecção da quebra de peças de caminhões da Volvo antecipadamente	Dispositivo: adaptador OBD-II Gateway: indefinido Tecnologias de localização: GPS
Morabito <i>et al.</i> , 2017	Plataforma para gerenciar recursos do veículo	Dispositivo: adaptador OBD-II Gateway: RaspBerry Pi 3

Após a análise das publicações sobre a evolução da comunicação veicular, observaram-se os seguintes pontos:

- Os conceitos sobre redes veiculares (V2V e o V2I) vêm sendo aplicadas, a fim de proporcionar mais segurança nas estradas;
- Os veículos e estradas não estavam preparados para receber essa infraestrutura de

comunicação, tornando restrito a algumas zonas;

- Com o advento da internet e o surgimento de vários conceitos de redes, como: redes móveis, internet das coisas, tornou-se mais simples ter o veículo conectado à internet. Sobre as aplicações existentes em comunicação veicular, foram verificados os

seguintes pontos nos trabalhos relacionados:

- Surgiram aplicações voltadas à comunicação entre veículo e qualquer coisa conectada à internet;
- Processamento de dados local, na nuvem, *smartphone* ou em qualquer coisa conectada à internet;
- O emprego de *gateways* para conectar o veículo a um local destinado ao processamento de dados ou armazenamento.

Sobre quais recursos tecnológicos são utilizados nas aplicações de redes veiculares, foi verificado nas pesquisas, o seguinte:

- Os dispositivos mais comuns foram os adaptadores OBD-II, nos trabalhos similares o mais utilizado foi o ELM 327;
- Foram utilizados *smartphones*, dispositivos e computadores obter os dados do veículo por meio dos adaptadores OBD-II;
- As tecnologias de comunicações mais usadas foram o Bluetooth, Wi-Fi e internet móvel (3G e 4G).

Sobre os dados que trafegam nessas redes, foram notados os seguintes pontos:

- Os dados coletados são referentes aos sensores de funcionamento do motor do veículo;
- Códigos de erro da central do veículo (por exemplo: sensor com mal funcionamento);
- Localização do veículo;
- Sensores que monitoravam o motorista, como sensor de fadiga.

Sobre as informações das mensagens visualizadas pelos usuários, foram observados os seguintes pontos dos trabalhos relacionados:

- Dados em tempo real dos sensores de funcionamento do motor, como: velocidade, rotação do motor, temperatura do motor, carga do motor, entre outros;
- Pontuação do modo de direção;
- Alertas de atenção indicando perigos na cidade, rodovia e no veículo.

Sobre os objetivos das informações visualizadas pelos usuários, foram verificados os seguintes pontos:

- Direção de forma econômica;
- Relatórios de desempenho;
- Notificação de alertas.

3.3 Contribuições dos trabalhos relacionados

Os trabalhos relacionados à AmI de Sampaio *et al.* (2012) e Corno e De Russis (2017) mostraram o que foi desenvolvido e o que há de mais novo nas aplicações de AmI, portanto foram trabalhos que nortearam esta pesquisa quanto ao rumo de aplicações dos paradigmas de AmI.

Da mesma forma, os trabalhos relacionados a redes veiculares de Kiening *et al.* (2013) e Chen *et al.* (2015) apresentaram uma coletânea de exemplos de aplicações e conceitos sobre esse assunto, no entanto, eram aplicações que visavam apenas a comunicação de veículo para veículo e a infraestrutura da rodovia para mantê-la mais segura.

Por outro lado, os trabalhos de Ang *et al.* (2018) e Naik *et al.* (2019) apresentaram um conjunto de artigos que mostraram o que há de mais de novo nas comunicações dos veículos. As aplicações envolvem os dois trabalhos citados anteriormente e apontam para a internet dos veículos, que estão conectados e transmitindo informações.

Nas tabelas de síntese dos trabalhos relacionados sobre AmIs, foi verificado que o *gateway* foi o dispositivo mais usado nas aplicações propostas. Com as análises da arquitetura, foi constatado que a estrutura de comunicação comum foi um servidor local no ambiente e um servidor na nuvem, ambos conectados pelo *gateway*. Isso foi observado nos trabalhos de Montori *et al.* (2017), Putthapipat *et al.* (2018), Vivek e Sunil (2016) e Silva (2016).

Os trabalhos de Montori *et al.* (2017) e Putthapipat *et al.* (2018) formaram uma das bases desta proposta, pois apresentaram detalhes da arquitetura mostrando as ferramentas, protocolos e tecnologias de comunicação. Os recursos tecnológicos empregados nesses trabalhos são listados a seguir:

- *Gateway*: RaspBerry Pi 3 com conexão à internet;
- Tecnologias de comunicação: Wi-Fi;
- Protocolos de comunicação: HTTP;
- Ferramentas: Python, Node.js e APIs.

Os recursos listados acima serão utilizados neste sistema para desenvolver os

servidores local e na nuvem, pois são ferramentas de fácil acesso. O computador RaspBerry Pi 3 tem custo baixo e as ferramentas são livres e com diversas bibliotecas de implementação.

Com relação à comunicação veicular, verificamos nos trabalhos relacionados que a maioria utilizou o adaptador OBD-II ELM 327, por ser acessível e conter duas configurações de comunicação, o Bluetooth e o Wi-Fi, além de possuir um manual bem detalhado (ELM327, 2019).

De acordo com as tabelas de sínteses sobre veículos, dos 16 trabalhos relacionados, cinco usaram o *smartphone* como *gateway* entre o veículo e aplicações externas, três utilizaram dispositivos instalados no veículo para armazenar e transmitir dados, quatro apresentaram revisões bibliográficas dos conceitos e segurança sobre redes veiculares, além disso, cinco utilizaram dispositivos que não foram identificados, mas foram responsáveis por coletar, armazenar e transmitir os dados.

Para este sistema, pretende-se instalar um dispositivo no veículo, para deixá-lo sempre ativo e automático tanto para coletar como para transmitir os dados. Dessa forma, os trabalhos que os autores instalaram um dispositivo, foi verificado que oito fizeram dessa forma, porém, somente três definiram o dispositivo usado.

Desses três, dois utilizaram o RaspBerry Pi 3 e elencaram diversos motivos pela escolha do computador, dentre os quais são citados os aspectos de processamento, plataforma de desenvolvimento, robustez, tecnologias de comunicação, utilização de bibliotecas nativas do Linux (Morabito *et al.*, 2017; Atzori *et al.*, 2018).

Fazendo uma análise nos recursos tecnológicos dos trabalhos relacionados, foi elaborada uma lista com as ferramentas e tipos de comunicações mais comumente empregadas entre os trabalhos pesquisados. Para este sistema, pretende-se utilizar os seguintes recursos:

- Dispositivo para interface com o veículo: ELM 327 Wi-Fi;
- Dispositivo instalado no veículo: RaspBerry Pi 3
- *Gateway*: roteador da casa;
- Ferramentas: Python, Node.js;
- Protocolo de comunicação: HTTP;
- Tecnologia de comunicação: Wi-Fi;
- Tipos de dados: JSON.

As tecnologias listadas são semelhantes aos dos trabalhos que utilizaram os conceitos de AmI. São de fácil acesso, pois podem ser encontrados tutoriais, vídeo aulas e manuais para

o desenvolvimento dos *scripts* e funções. O *gateway* será o elo de conexão entre o veículo, o servidor na nuvem e o AmI da casa (servidor local).

3.4 Contribuição desta Pesquisa

A partir dos questionamentos feitos no início do capítulo nas Tabelas 3 e 4, as tabelas sínteses dos trabalhos relacionados e as análises que definiram os materiais e ferramentas, foram formuladas questões com o objetivo de montar uma tabela de comparação dos trabalhos relacionados para um melhor entendimento sobre a contribuição deste sistema e a responder a hipótese H1 citada no início deste capítulo. A resposta SIM será representada pelo sinal (+) e NÃO pelo sinal (-). As perguntas são:

- a. Conecta-se com algum ambiente?
- b. Notifica o usuário por algum dispositivo (*smartphone*, TV, aparelhos, computador)?
- c. Possui conexão com a internet?
- d. Utiliza *smartphone* como *gateway*?
- e. Utiliza outros dispositivos como *gateway*?
- f. Coleta dados de algum objeto (casa, equipamentos, veículo)?

A formulação das perguntas buscou unir os dois assuntos desta pesquisa e apresentar pela tabela de comparação um sistema capaz de mesclar as principais características de AmI e redes veiculares formando um sistema que integra o veículo com AmI de uma casa, e assim, criando uma base para novas aplicações e serviços.

A pergunta (a) tem relação com uma das principais características de AmI, pois é um pré-requisito que dispositivos estejam conectados à rede para que possam transmitir dados com o intuito de monitorar ou controlar o ambiente.

A pergunta (b) tem relação com uma das características que este sistema deve ter e que foi observado em trabalhos relacionados à AmI, especificamente em ambiente assistidos, os quais tinham o objetivo de notificar o usuário quanto a tomada do remédio no horário certo.

A pergunta (c) tem relação com um pré-requisito de AmI e também é um dos recursos que os trabalhos mais recentes em comunicação veicular estão empregando para que o veículo possa estar sempre conectado recebendo e transmitindo informações na internet.

A pergunta (d) e (e) são os dispositivos utilizados como ponte para coletar e transmitir os dados, nos quais foram observados dispositivos dedicados para essa finalidade e os multitarefas que são os *smartphones*.

A pergunta (f) tem relação com a coleta de dados dos dispositivos, equipamentos ou mesmo o veículo para serem transmitidos para outros dispositivos e serem armazenados ou analisados.

A Tabela 10 apresenta a tabela de comparação entre os trabalhos relacionados com as perguntas respondidas e o sistema proposto.

Tabela 10: Tabela de comparação dos trabalhos relacionados

Referências\ Questões	a	b	c	d	e	f	
Moseley, 2017	+	-	+	-	-	+	AmI
Obukata <i>et.al</i> , 2016	+	-	+	-	+	+	
Frey, 2013	+	-	+	-	+	+	
Montori <i>et al.</i> , 2017	+	-	+	-	+	+	
Putthapipat <i>et al.</i> , 2018	+	-	+	-	+	-	
Vivek and Sunil, 2016	+	-	+	-	+	+	
Seif e El-Saber, 2017	+	-	+	-	+	-	
Campbell <i>et al.</i> , 2016	+	-	+	-	+	-	
Rahmani and Kanter, 2017	-	-	-	-	-	+	
Yacchirema and C.Palau, 2016	+	+	+	-	+	+	
Silva <i>et al.</i> , 2016	+	+	+	-	+	+	
Seabra, <i>et al.</i> , 2016	+	-	+	-	+	+	
Soultanopoulos <i>et al.</i> , 2016	+	-	+	+	-	-	
Bhargava e Ivanov, 2018	+	-	+	+	-	-	
Silva et al., 2018	-	+	+	+	-	+	Redes veiculares
Atzori et al., 2018	-	+	+	-	+	+	
Shi <i>et al.</i> , 2016	-	-	+	+	+	-	
Amarasinghe <i>et al.</i> , 2015	-	+	+	+	-	-	
Magaña <i>et al.</i> , 2016	-	+	+	+	-	-	
Wallace <i>e.t al.</i> , 2015	-	-	-	-	+	+	
Siegel <i>et al.</i> , 2014	-	+	+	-	+	+	
Malekian <i>et al.</i> , 2017	-	+	-	-	+	+	
Meseguer <i>et al.</i> , 2017	-	+	+	+	-	+	
Martínez <i>et al.</i> , 2015	-	+	+	-	+	-	
Prytz, 2014	+	+	-	-	+	+	
Morabito <i>et al.</i> , 2017	-	-	-	+	-	-	
Sistema proposto	+	+	+	-	+	+	Contribuição

Os trabalhos relacionados apresentados na Tabela 10 diferem do presente trabalho, pois possibilitam que o veículo possa se conectar à internet utilizando dispositivo instalado no veículo e transmitindo os dados se conectando de forma automática a um ponto de internet sem fio para transmissão desses dados para servidor na nuvem.

Dessa forma o AmI da casa pode acessar os dados na nuvem para enfim distribuir informações de dados anormais de funcionamento do veículo por meio de notificação de

mensagens curtas para dispositivos conectados à rede do AmI da casa, como o *smartphone*.

3.5 Considerações finais sobre o capítulo

Neste capítulo, foi apresentada a metodologia de busca dos trabalhos relacionados, bem como os estudos sobre os trabalhos relacionados ao AmI e comunicação veicular. Foram analisadas as aplicações existentes de AmI avaliando as ferramentas e dispositivos utilizados nesses ambientes. Em relação à comunicação veicular, foram investigadas técnicas, abordagens e sistemas similares para desenvolver o presente sistema.

Deste modo, este capítulo teve a importância de apresentar os assuntos que embasam este sistema, bem como, o *gap* da pesquisa científica. Da mesma forma, foi apresentada a contribuição desta pesquisa que visa conceber um novo conceito de sistema que conecte veículo e ambiente formando uma base para novas aplicações e serviços.

4. Arquitetura do sistema

Neste capítulo é apresentada a segunda fase desta pesquisa que é a concepção da solução inspirada nos estudos dos trabalhos relacionados e a divisão em partes da arquitetura do sistema proposto.

Desse modo, o capítulo está organizado da seguinte forma: primeiramente é descrita a concepção da solução, em seguida é apresentada uma visão geral da arquitetura do sistema, finalizando com as considerações finais do capítulo.

4.1 Concepção da Solução

Buscou-se utilizar um dispositivo embarcado para coletar os dados do veículo, pois são dispositivos com baixo custo financeiro, com poder de processamento suficiente para a aplicação, fácil de programar e confiável. Além disso, tem acesso à internet por rede sem fio Wi-Fi ou 3G/4G para transmitir os dados diretamente para servidores na nuvem.

O envio desses dados para a nuvem possibilita o acesso de qualquer ponto remoto com internet, desde que o usuário seja habilitado. Além disso, foi observado nos trabalhos relacionados que a tendência é enviar os dados do veículo para a nuvem, a fim de gerar outros serviços, como rede social veicular, agendamento para manutenção e indicar autopeças mais próximas.

Outro ponto importante é a utilização dos recursos de computação que os AmI podem proporcionar. Um desses recursos é a infraestrutura de comunicação (conecta dispositivos do ambiente à internet), que pode ser usada para gerar outras aplicações que ajudem os usuários a identificar possíveis problemas nos veículos como se fosse um aparelho qualquer da casa.

Foi verificado nos trabalhos relacionados sobre AmI que havia um dispositivo dedicado para coletar e concentrar os dados, o qual tinha a função de enviar e receber as informações de servidores na nuvem.

4.2 Visão geral e arquitetura do sistema

A partir da concepção da solução foi possível fazer um esboço deste sistema dividindo em módulos que facilitarão no desenvolvimento, conforme a Figura 6.

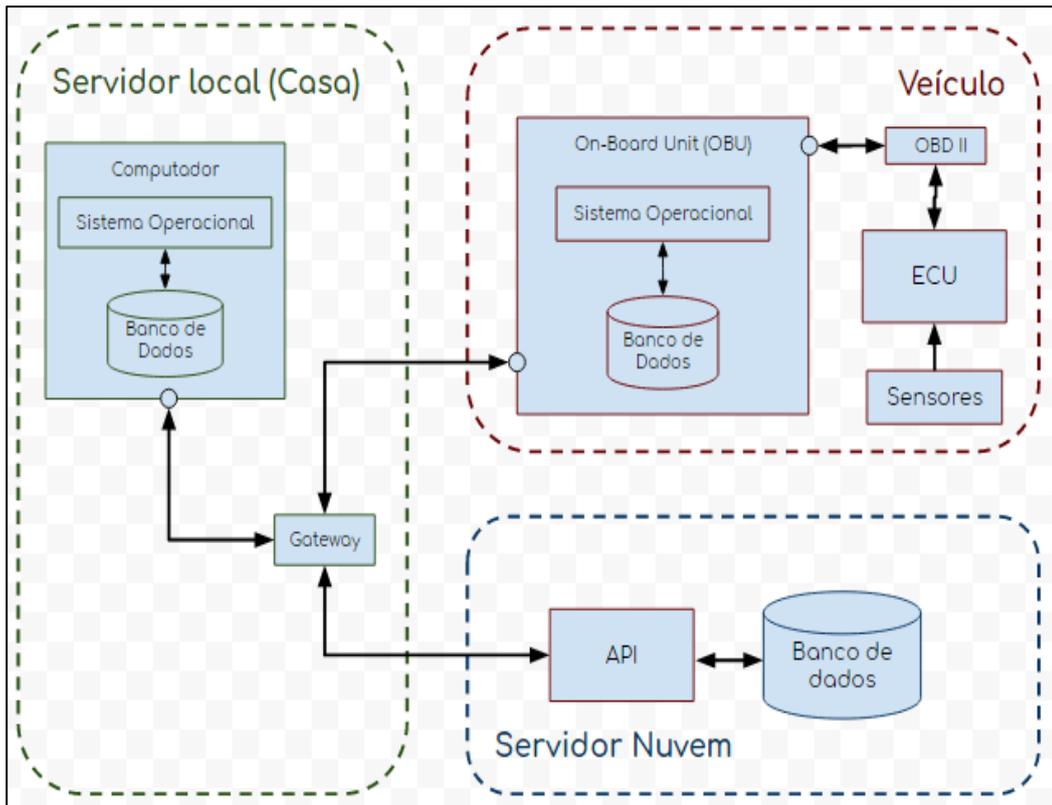


Figura 6: Esboço geral do sistema proposto

O sistema é formado por três módulos, denominadas de módulo veículo, módulo servidor na nuvem (SN) e módulo servidor local (SL). Conforme a Figura 7, o módulo veículo é formado por um carro fabricado a partir de 2010 com os seguintes dispositivos instalados:

- Adaptador ELM 327 Wi-Fi: dispositivo conectado a porta OBD-II, responsável por abstrair os protocolos de comunicação da rede e de sensores da ECU do veículo;
- Dispositivo embarcado: *On-Board Unit* (OBU) é formado pelo RaspBerry Pi 3 com *scripts* embarcados para coletar, armazenar e transmitir os dados dos sensores de funcionamento do veículo.



Figura 7: Módulo veículo

O módulo SN é um ambiente dedicado a armazenar os dados dos trabalhos produzidos pelos estudantes de graduação e pós-graduação do Laboratório de Ambientes Inteligentes (LabAmI) da Universidade Federal do Amazonas (UFAM).

Os dados são recebidos e filtrados na nuvem por uma API que retorna dados formatados de acordo com o intervalo de tempo informado. Para consumir as informações produzidas no módulo SN foi construído o módulo SL.

O módulo SL está localizado no ambiente e é formado por um computador portátil que simula o nó central de um AmI, esse computador está conectado à internet e tem a função de gerar o servidor local

Esse servidor local possui *scripts* com a função de requisitar os dados do módulo SN pelos parâmetros de intervalo de tempo e notificar o usuário enviando mensagens em aplicativo de mensagens do *smartphone* quando houver dados anormais de funcionamento do veículo.

O *gateway* utilizado para conectar o OBU e o SN é um roteador Wi-Fi no ambiente. Entre o SN e o SL pode haver outros tipos de comunicação, como a internet móvel 3G/4G. Para melhor entendimento será apresentada uma visão geral de funcionamento do sistema proposto, identificando e descrevendo cada bloco conforme a Figura 8.

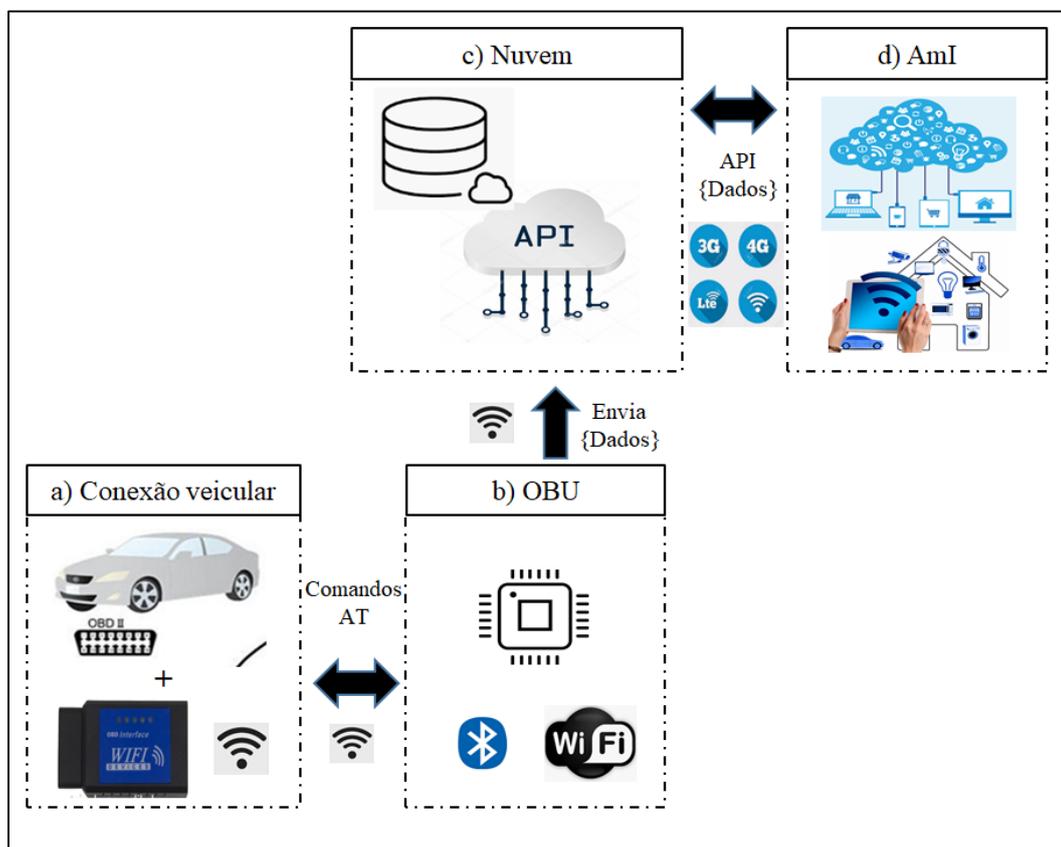


Figura 8: Visão geral de funcionamento do sistema

A comunicação do bloco Conexão veicular e OBU é realizada via rede sem fio, entre OBU e o bloco Nuvem é feita por comunicação Wi-Fi, entre os blocos Nuvem e AmI pode ser realizada por Wi-Fi ou internet móvel 3G/4G.

A arquitetura do sistema foi dividida em quatro blocos identificados como (a) Conexão veicular, (b) OBU, (c) Nuvem e (d) Servidor local, os quais são descritos quanto aos aspectos de materiais utilizados, comunicações e dados trocados. A seguir:

a) Conexão veicular:

É uma interface de comunicação em tempo real entre as unidades centrais (ECUs). O veículo possui o conector de interface do OBD-II, conhecido como *Data Link Connector* (DLC) ou simplesmente porta OBD-II. O dispositivo adaptador OBD-II (também chamado de “scanner OBD”) se conecta à porta OBD-II e atua como uma ponte entre o OBU e o veículo utilizando protocolos de comunicação, como: Rede de Área Controladora (CAN), ISO 9141-2 e ISO 14230. Esses protocolos são configurados de forma automática no adaptador OBD-II facilitando as requisições de dados que são feitas por comandos AT pelo OBU;

b) OBU:

É o dispositivo embarcado responsável por coletar os dados do veículo e enviar para o servidor na nuvem. O OBU se conecta ao adaptador OBD-II por meio do protocolo TCP/IP, no qual o OBU é o cliente e o adaptador é o servidor. As mensagens de requisição de dados do cliente são feitas por comandos AT e o conteúdo das mensagens de resposta do servidor são retornados no formato hexadecimal. O OBU faz as conversões de hexadecimal para decimal para armazenar no banco de dados interno. Além disso, os dados são formatados para JSON e enviados para o servidor na nuvem utilizando protocolos TCP/IP e HTTP sempre que consegue estabelecer conexão com rede sem fio. Para vislumbrar o formato do programa, bem como suas funções, foi concebido um diagrama de atividades apresentado na Figura 9.

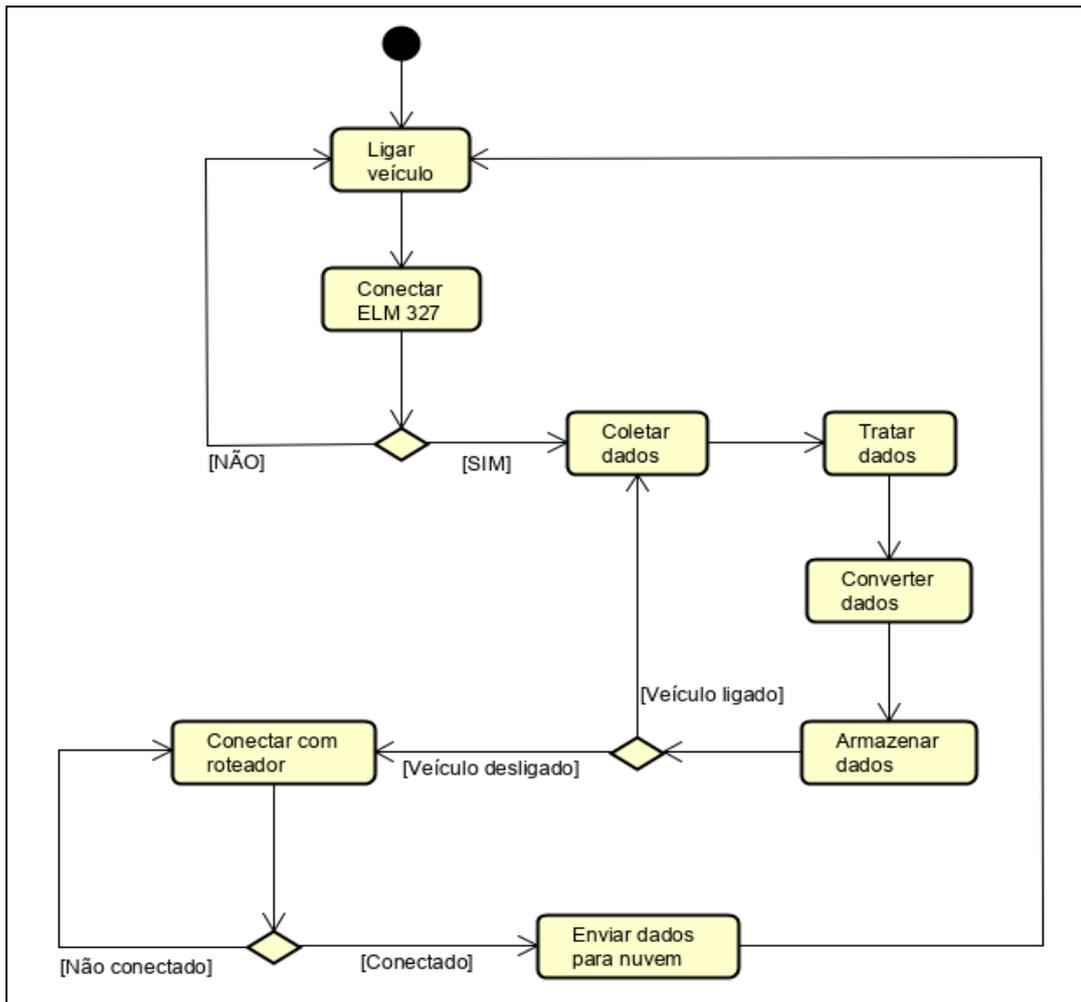


Figura 9: Diagrama de atividades

As funcionalidades foram mapeadas de acordo com o diagrama acima. Assim, foi possível organizar e facilitar o desenvolvimento do programa de modo que as funções fossem modularizadas e criados arquivos com funções mais específicas.

c) Nuvem:

É uma aplicação desenvolvida com um framework que utiliza PHP e banco de dados MySQL em um servidor na nuvem que é responsável pelo armazenamento e filtragem dos dados dos sensores do veículo. Há uma API que recebe os dados formatos JSON do OBU, armazena no banco de dados e filtra através de métodos parâmetros dos sensores considerados anormais (por exemplo: sensor de temperatura de arrefecimento do motor > 90). Esses métodos disponibilizam dados formatados em JSON com os sensores filtrados por valores pré-estabelecidos na API; O AmI, por meio dos métodos tem acesso aos dados que são baixados no formato JSON;

d) AmI:

É formado por um computador portátil que é o nó central que interliga os dispositivos do ambiente. Esse computador possui ferramentas que geram um servidor local e *scripts* que realizam consultas por meio do método POST que requisitam dados do servidor na nuvem. O AmI da casa também é responsável por coletar e armazenar esses dados no banco de dados local. Os programas do AmI possuem funções que realizam consultas periódicas no banco de dados local. Quando há dados, eles são coletados e utilizados para realizar as notificações no aplicativo Telegram por meio de mensagens (formato JSON) para *smartphone* sobre os sensores de funcionamento do veículo.

4.3 Considerações finais do capítulo

Este capítulo descreveu o processo da concepção da solução a partir dos trabalhos relacionados, o esboço e a arquitetura do sistema, dividindo-o em quatro partes a fim de facilitar o entendimento e o desenvolvimento. Cada parte foi descrita em detalhes mostrando os tipos de comunicação e os dados trocados entre as partes.

Este capítulo foi importante para o desenvolvimento da arquitetura e a divisão nos seguintes blocos: conexão veicular, OBU, nuvem e AmI, os quais facilitarão o planejamento e organização da implementação do sistema no próximo capítulo.

5. Implementação do sistema

Neste capítulo será apresentada a implementação de cada módulo da arquitetura e os recursos necessários para a construção deste sistema. Desse modo, o capítulo está organizado da seguinte forma: primeiramente na seção 5.1 é apresentado o dispositivo para extrair os dados do veículo. Em seguida, na seção 5.2 será mostrada a implementação do dispositivo OBU. A Seção 5.3 mostrará os filtros implementados no servidor na nuvem. Na seção 5.4 será apresentado os programas feitos no AmI da casa. Seguindo na seção 5.5 que apresentará a arquitetura de comunicação dando ênfase nos *hardwares* utilizados entre os módulos. Finalizando na seção 5.6 as considerações finais do capítulo.

5.1 Conexão veicular

Para este sistema foi utilizado o adaptador OBD-II ELM 327 com comunicação Wi-Fi, o qual possui um manual bem detalhado. Com o manual foi possível entender como funcionava a dinâmica de comunicação entre a rede de sensores do veículo e o adaptador ELM 327 que funciona como uma interface com outros dispositivos.

5.2 OBU

Nesta seção será apresentado o dispositivo responsável por coletar, processar, armazenar e transmitir dados para o servidor na nuvem, bem como as configurações de *hardware* e os *softwares* implementados para tal objetivo.

O conjunto poder de processamento, preço, sistema operacional e disponibilidade desse recurso no Laboratório de Ambientes Inteligentes (LabAmbInt) da UFAM foi determinante para a escolha do RaspBerry Pi 3 (RPi 3).

O RPi 3 possui três tipos de conexão, Wi-Fi, *Bluetooth* e rede Ethernet. A comunicação escolhida para a implementação do dispositivo foi o Wi-Fi, pois haviam diversos exemplos simples de socket no sistema operacional que simulavam as comunicações entre dispositivos sem fio. Vale ressaltar que o sistema operacional utilizado foi o Linux, o qual possui código aberto e várias bibliotecas com tutoriais e fóruns de apoio indicando como são utilizadas.

A linguagem de programação utilizada foi o Python, que é de código aberto e possui várias bibliotecas prontas e compartilhadas na internet. Há diferentes exemplos que

envolvem a coleta de dados por protocolo utilizando sockets TCP, que será utilizado na comunicação entre o adaptador ELM 327 Wi-Fi com o OBU.

Para realizar testes de comunicação entre o adaptador ELM 327 e o RPi 3, foi montado um protótipo, apresentado na Figura 10. Na Rpi 3 foi instalado o sistema operacional *Raspbian*, que é baseado em uma distribuição Linux com diversos pacotes pré-instalados, como o Python 3.

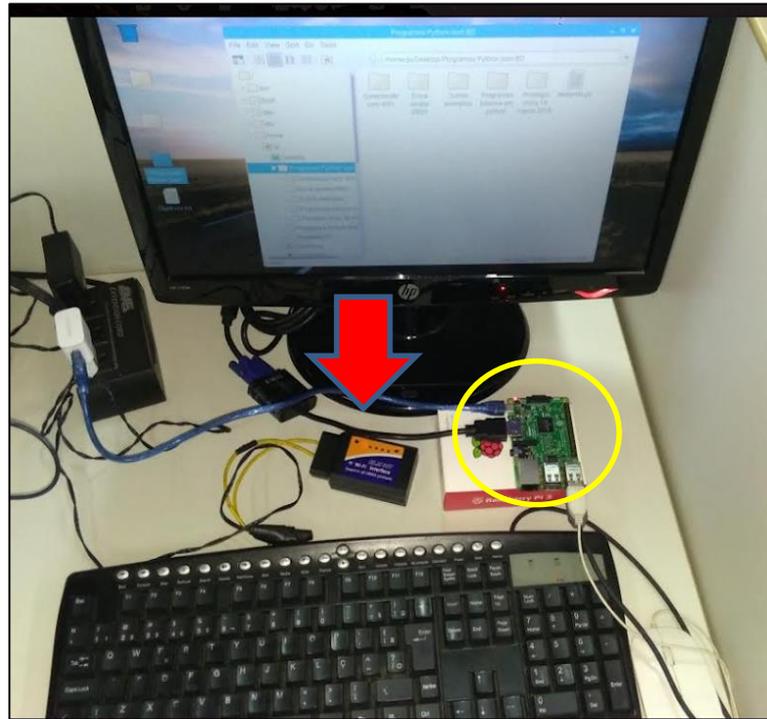


Figura 10: Protótipo do dispositivo OBU

Foi criado um programa para conectar RPi 3 (cliente) ao adaptador ELM 327 (servidor) utilizando os recursos das bibliotecas de comunicação de mensagens TCP, os sockets.

Foram criadas mais duas funções responsáveis respectivamente por enviar um comando de requisição de dados e coletar a resposta com o dado do sensor requisitado, conforme a Figura 11.

```

2  import time
3  from socket import *
4  #####
5  ##### Funcoes #####
6  #####
7
8  # Connect obd2
9  host='192.168.0.10' → IP do ELM 327
10 port=35000 → Porta
11 buffer_size = 1024 → Tamanho mensagem retornada
12 con = socket(AF_INET, SOCK_STREAM)
13 con.connect((host,port))
14 timewait = 0.8 # variavel global utilizada para
15                 #setar o tempo de espera
16 → Tempo de espera para receber mensagem
17 ##### Somente envia #####
18 #####
19 def sendMessage(message): → Envia comando
20     con.send(message)
21     time.sleep(timewait)
22     return 0
23 #####
24 #####
25
26 #####e recebe mensagem #####
27 #####
28 def receiveMessage(): → Recebe dado requisitado
29     messageReturned = con.recv(buffer_size)
30     return repr(messageReturned)
31 #####
32 #####

```

Figura 11: Script com funções para habilitar conexão e transmissão de mensagens

A Figura 11 apresenta o trecho do programa em Python que automatizou os processos de se conectar, enviar e receber os dados fornecidos pelo adaptador ELM 327. Porém, esse é um dos vários programas necessários para atingir as funcionalidades do OBU.

A Figura 12 apresenta os programas criados e implementados no OBU para atingir o objetivo de tornar a coleta e transmissão de dados automática sem a necessidade de intervenção do usuário.

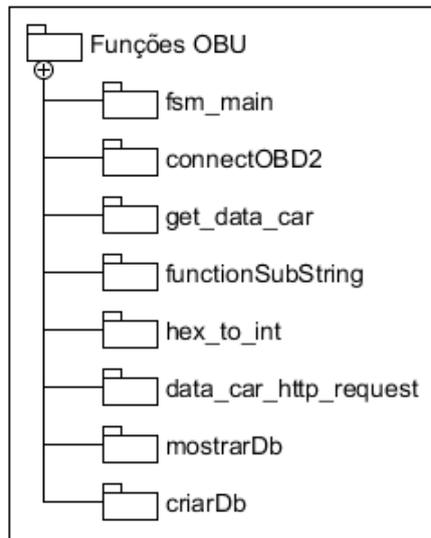


Figura 12: Módulos do OBU

Ao todo foram desenvolvidos oito módulos de funções para formar o dispositivo OBU, os quais serão descritos a seguir:

1 – Módulo fsm_main: é o módulo principal, na qual todas as funções são utilizadas para o OBU realizar seus procedimentos que inicia na coleta e finaliza enviando dados para nuvem no formato JSON através do protocolo HTTP. As funções podem ser visualizadas na Figura 13.

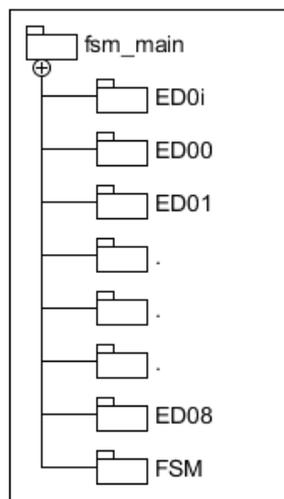


Figura 13: Funções do programa fsm_main

2 – Módulo connectOBD2: foram desenvolvidas funções responsáveis por conectar ao adaptador OBD-II utilizando protocolo TCP/IP, função que enviam e recebem mensagens via *socket* TCP/IP, o esboço do módulo com as funções pode ser visualizado na Figura 14.

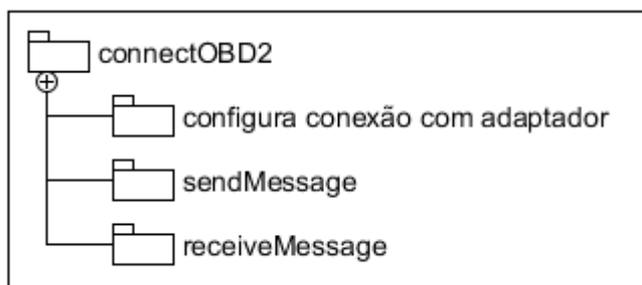


Figura 14: Funções do programa connectOBD2

3 – Módulo get_data_car: foram desenvolvidas funções que preparam o adaptador OBD-II para iniciar a coleta de dados da ECU. Os primeiros comandos para isso é o de resetar e configurar o protocolo para detectar de forma automática. Nessa função são realizadas as requisições dos dados, bem como o armazenamento no banco de dados do OBU. O esboço do módulo com as funções pode ser visualizado na Figura 15.

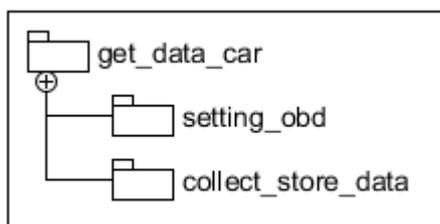


Figura 15: Funções do programa get_data_car

Os comandos necessários para os procedimentos de coleta de dados estão descritos conforme a Tabela 11.

Tabela 11: Exemplos de comandos utilizados

Comando	Resposta	Comentário
ATZ	ELM237 V1.5	Reset - versão do software
AT SP 0	OK	Escolha do protocolo utilizado - automático
01 04	41 04 00	Carga do motor
01 05	41 05 00	Temperatura do líquido de arrefecimento
01 0C	41 0C 00	Rotação atual do motor
01 0D	41 0D 00	Velocidade instantânea do veículo
01 0E	41 0E 00	Avanço do ângulo de ignição
01 0F	41 0F 00	Temperatura do Ar
03	43 00	DTC – Sem DTC (erros armazenados)

A Tabela 11 mostra os comandos enviados do OBU para o adaptador ELM 327, as respostas retornadas de cada comando e a descrição de cada um respectivamente. Vale

ressaltar que a tabela representa alguns exemplos de comandos. Contudo, há vários comandos relacionados aos sensores do veículo e funcionam em todos os veículos fabricados a partir de 2010 no Brasil, pois obedecem aos padrões internacionais de fabricação.

No banco de dados do OBU foi utilizado o SQLite. Foi criado um banco de dados com o nome 'CarMonitoringDataBase.db' e uma tabela com o nome 'sensores' de 16 colunas, dentre as quais, a primeira coluna é o identificador único (id), a segunda são dados de data e hora e as demais são os 14 sensores coletados do veículo. A estrutura da tabela é semelhante à estrutura apresentada na Figura 16.

id	Data	Status sistema comb.	Carga motor	Temp. motor	Regulador comb. baixo	Regulador comb. Alto
1	2019 08 19	103	104	105	106	107

Figura 16: Colunas da tabela criada para salvar os dados

A central do veículo disponibilizou 14 sensores de funcionamento, esses sensores são:

- Status do sistema de combustível (PID 103);
- Carga do motor (PID 0104);
- Temperatura de arrefecimento do motor (PID 0104);
- Regulador de combustível baixo (PID 0106);
- Regulador de combustível alto (PID 0107);
- Velocidade (PID 010D);
- RPM (010C);
- Ignição do cilindro (PID 010E);
- Temperatura do ar (PID 010F);
- Taxa de fluxo de ar (PID 0110);
- Posição do acelerador (PID 0111);
- Sensor de oxigênio 1 (PID 0114);
- Sensor de oxigênio 2 (PID 0115);
- Distância percorrida (PID 0121).

4 – Módulo functionSubString: foram desenvolvidas funções que processam o retorno das mensagens, verificando-se o cabeçalho, pois a partir disso é possível identificar quando há mensagens com dados (inicia com os caracteres 41) e quando não há retorno de dados (sem caracteres 41).

Esse módulo também realiza o tratamento da mensagem recebida que é readequada, tirando-se espaços e caracteres especiais para ser convertida de hexadecimal para decimal, conforme a Tabela 12. O esboço das funções pode ser visualizado na Figura 17.

Tabela 12: Processamento das mensagens

Ex:	Dado recebido	Funções	
		message_return	functionSubString
1	'010C\r41 0C 0D 13 \r\r>'	'010C\r41 0C 0D 13 \r\r>'	'010C410C0D13>'
2	'0105\r41 05 85 \r\r>'	'0105\r41 05 85 \r\r>'	'0105410585>'
3	'0110\rNO DATA\r\r>'	0	0

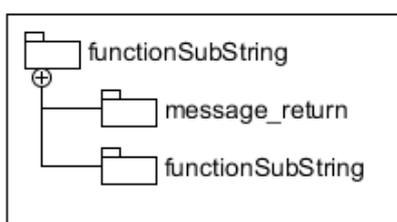


Figura 17: Funções do programa functionSubString

Pode-se observar nos exemplos 1 e 2 da tabela acima que a mensagem retornada tem os caracteres 4 e 1, logo, a mensagem possui dados dos sensores do motor. Em contrapartida, o exemplo 3 não retornou os caracteres que indicam que há dados, logo a função retorna valor 0 (zero). Como os programas são modularizados, essas funções serão usadas em outros módulos que vão utilizar os valores das variáveis de retorno.

5 – Módulo hex_to_int: foram desenvolvidas as funções de conversão de hexadecimal para decimal. Cada sensor tem uma fórmula diferente para realizar a conversão, conforme a Figura 18 que apresenta alguns exemplos.

Descrição	PID (hex)	Mínimo	Máximo	Unidade	Fórmula
Carga do motor (valor calculado)	04	0	100	%	$A*100/255$
Temperatura do fluido de arrefecimento	05	-40	215	°C	$A-40$
RPM	0C	0	16.383,75	RPM	$((A*256)+B)/4$
Velocidade do veículo	0D	0	255	km/h	A
Temperatura do ar de admissão	0F	-40	215	°C	$A-40$
Posição da borboleta	11	0	100	%	$A*100/255$

Figura 18: Fórmulas de conversão de alguns sensores

Um exemplo pode ser o dado coletado '0105410585>', o qual já passou pela função que converte o dado para que outra função possa realizar a leitura da mensagem. A função `hex_to_int` recebe o dado e coleta apenas os dois últimos caracteres antes do caractere (>), um trecho do *script* pode ser visualizado na Figura 19.

```
# =====  
def engineCoolantTemperature_hex(code):  
    data = functionSubString(code)  
    if data == NODATA:  
        return data  
    else:  
        data = int(data[2:],16)  
        return data - 40  
# =====
```

Figura 19: Script de conversão do sensor de temperatura do motor

No *script*, pode-se observar que há uma condição que verifica se o dado está correto, caso esteja correto, o dado coletado é convertido para decimal sendo selecionados apenas os caracteres de interesse (depende da fórmula aplicada). Os esboços do módulo com as funções podem ser visualizados na Figura 20.

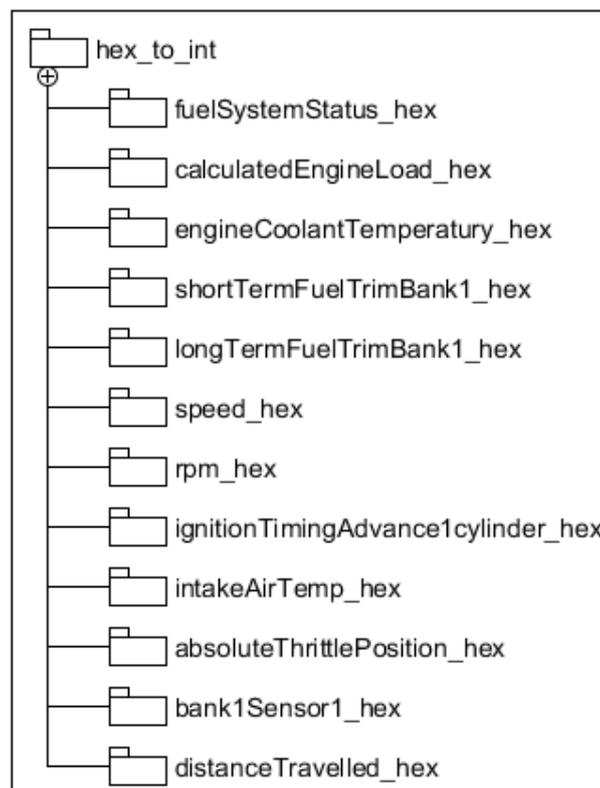


Figura 20: Funções do programa hex_to_int

6 – Módulo `data_car_http_request`: foram desenvolvidas funções que preparam os dados salvos no banco de dados do OBU, os quais serão convertidos em dados no formato JSON a fim de serem enviados para o servidor na nuvem. As funções são descritas conforme a Tabela 13.

Tabela 13: Descrição das funções

Funções	Descrição
getSensoresFromSqlite	Seleciona todos os dados do banco de dados do OBU
convertToJson	Coleta as linhas do banco de dados e monta um conjunto de dados no formato JSON
uploadData	Envia os dados no formato JSON para o servidor na nuvem
deleteSensoresFromSqlite	Apaga os dados que já foram enviados para nuvem

Os dados JSON tem uma formatação definida na implementação da função do lado OBU. Essa mesma formatação deve ser implementada no servidor da nuvem, pois caso contrário, os dados não são salvos. O esboço do módulo com as funções que foram implementadas pode ser visualizado na Figura 21.

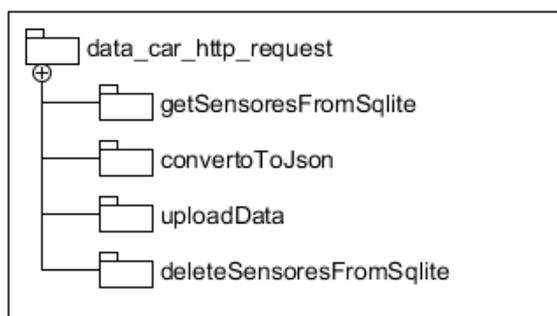


Figura 21: Funções do programa `data_car_http_request`

7 – Módulo `mostrarDb`: é a função responsável para realizar testes e mostrar os dados salvos no banco de dados interno do OBU.

8 – Módulo `criarDb`: é a função responsável por criar o banco de dados no OBU, essa função deve ser executada apenas uma vez.

5.3 Nuvem

O servidor na nuvem é um espaço de hospedagem web utilizando o framework *Yii2* para implementar os serviços, os quais serão descritos no decorrer da seção. O serviço de hospedagem web, chamado ufam-automation, é composto por banco de dados *MYSQL*, gerenciamento de banco de dados *PHPMYADMIN*, servidor web *APACHE2* e servidor FTP.

O acesso ao site é livre, pois são apresentados os trabalhos de pesquisa do grupo de estudos do laboratório. Contudo, para esta aplicação é necessário o login e senha para ter acesso aos dados de funcionamento do veículo.

O servidor na nuvem tem o objetivo de armazenar todos os dados dos sensores coletados pelo OBU. É utilizado uma API de leitura e gravação de dados com filtros que identificam dados anormais segundo definições prévias. Para ter acesso aos dados foram criados métodos em API que os retornam de acordo com o intervalo de tempo informado.

O ufam-automation é destinado para repositórios, arquivos dos membros do LabAmbInt, bem como projetos de graduação e pós-graduação. Esses projetos geralmente utilizam o ufam-automation para servir de interface do sistema com o usuário e armazenamento dos dados coletados em nuvem.

A ferramenta de programação utilizada para manipular os dados no servidor é o *Yii2*, o qual é um framework *PHP* baseado em componentes de alto desempenho destinado a desenvolvimento rápido de modernas aplicações web. Com esse framework pode ser criado desde um simples site até aplicações mais sofisticadas utilizando APIs.

Na Figura 22 é mostrado o site do ufam-automation (<http://ufam-automation.net/walmiracioli/carmonitoring>), no qual é possível visualizar os dados coletados pelo OBU na aba histórico dos sensores, assim como os alertas de temperatura, os alertas de velocidade, os alertas de rotação e erros da centrado do veículo. Esses alertas são valores de grandeza dos sensores que ultrapassaram parâmetros estabelecidos logo a seguir.

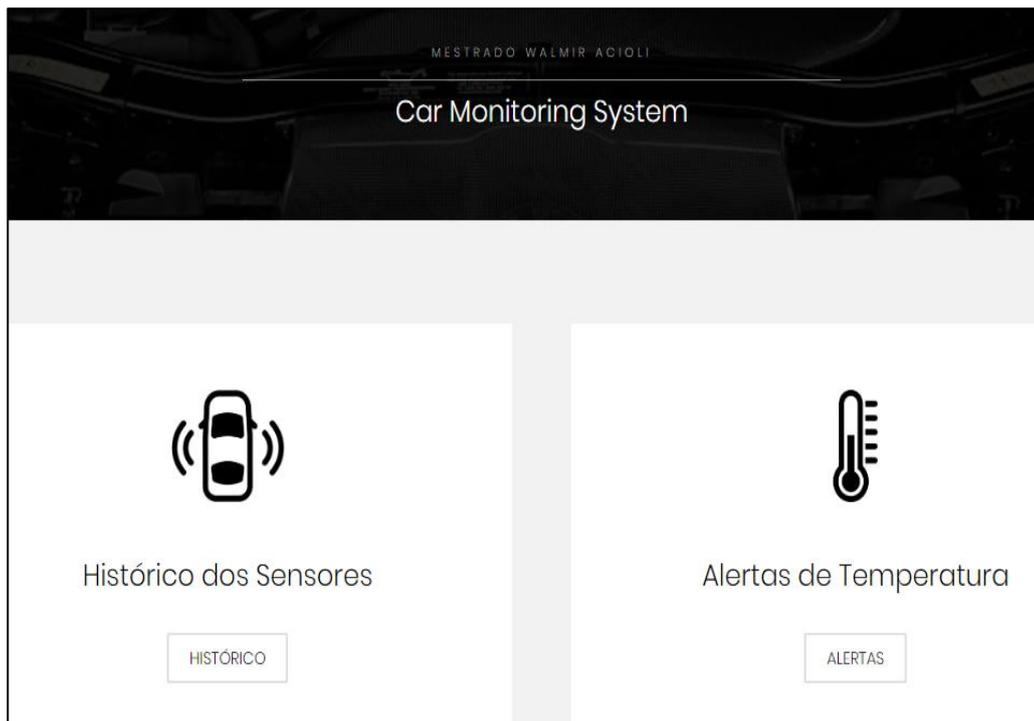


Figura 22: Tela inicial da aplicação na nuvem

Na nuvem foi implementada uma API com quatro métodos responsáveis por detectar os parâmetros anormais dos seguintes sensores:

- Temperatura: o sistema de arrefecimento é responsável por manter o veículo a uma temperatura ideal de 90 graus Celsius variando para mais ou para menos de acordo com o local (Nemati e Seyedtabaii, 2014). No método foi configurado o parâmetro para coletar valores iguais ou maiores que 100 graus Celsius;
- Velocidade: esse parâmetro é um forte indicador de condução do motorista, uma vez que geralmente a velocidade na cidade é de no máximo 60 km/h (Srinivasan, 2018). No método foi configurado para obter velocidades acima de 60 km/h;
- Rotação do motor: esse parâmetro é um indicador importante, pois é possível detectar a utilização inadequada do veículo quando a rotação é muito alta, acima de 2500 rotações (Khanapuri et al., 2015). O método utilizou parâmetro de 2000 rotações por segundo.
- Erros da central: o modo de emergência é acionado quando a ECU por algum motivo recebe sinais anormais dos sensores do motor, ao acontecer isso, a luz-espia do painel do veículo liga indicando problemas. O método responsável por detectar os erros é acionado quando o valor é diferente de 0 (zero).

A Figura 23 mostra os procedimentos entre a transmissão dos dados do OBU para o site de hospedagem ufam-automation e o download dos dados por meio de métodos da API que retornam dados anormais do veículo.

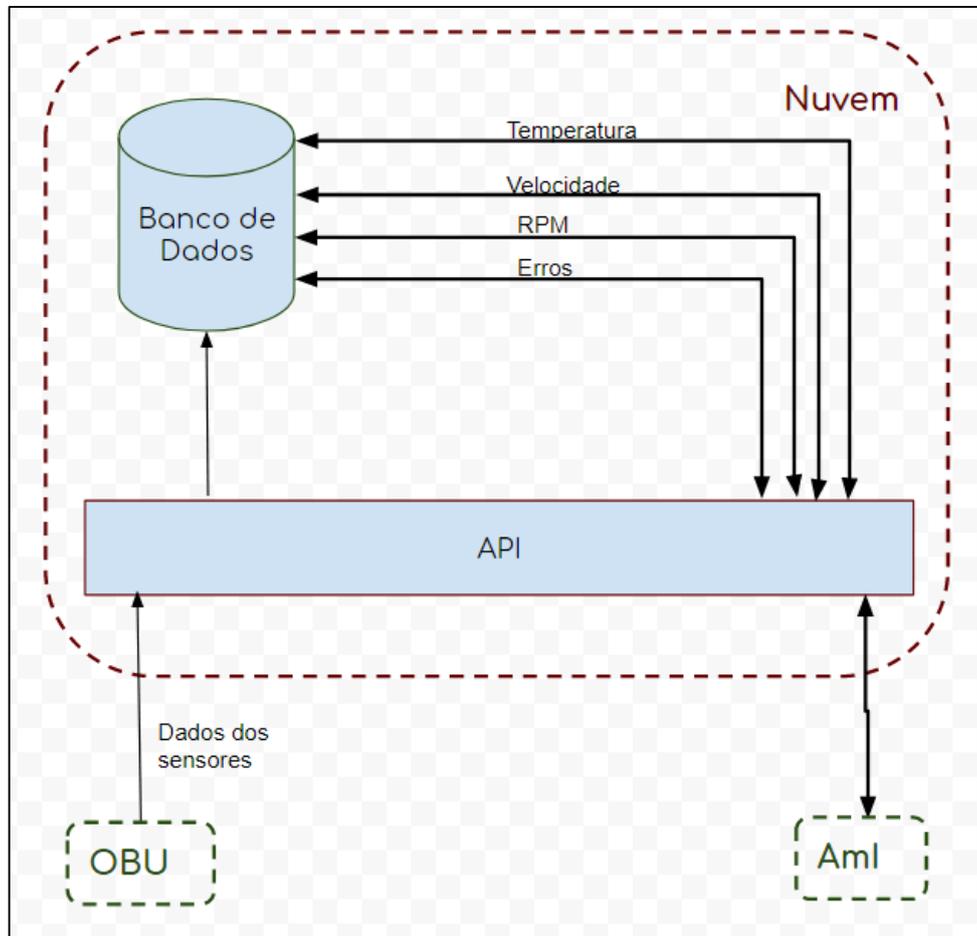


Figura 23: Esboço de entrada e saída dos dados

A dinâmica inicia quando o OBU envia dados para a nuvem. Na nuvem há uma API que recebe os dados no formato JSON e armazena em uma tabela do banco de dados todas as informações de funcionamento dos 14 sensores do veículo. Na tabela a mesma API que armazena possui 4 (quatro) métodos responsáveis por detectar os parâmetros configurados anteriormente (temperatura de arrefecimento, velocidade, RPM e erros da ECU). A API separa os dados com parâmetros anormais e os envia separadamente quando solicitados por meio dos métodos API. A Figura 24 apresenta um trecho em destaque do método que requisita um dos parâmetros selecionados para monitorar o veículo.

```

def getTempAlerts(date_min, date_max):
    temp_alerts_url = 'http://ufam-automation.net/walmiracioli/carmonitoring/web/index.php?r=api-
v001/get-temp-alerts'
    data_range = json.dumps([date_min, date_max])

    r = requests.post(url = temp_alerts_url, data = data_range, headers={'Content-Type': 'application/json',
'Accept': 'application/json'})

    return json.loads(r.text)

```

Figura 24: Script que solicita dados da nuvem com API

Nesse *script* é retornado os dados acima do normal da temperatura de arrefecimento do motor entre as datas de início e fim disponibilizada pelo usuário. Essa função foi implementada na linguagem Python e serviu para realizar testes com a API que retorna um conjunto de dados no formato JSON.

5.4 AmI

O AmI da casa é responsável por coletar os dados da nuvem, processar e os distribuir nos dispositivos conectados em sua rede. Nessa versão prototipada o usuário receberá informações do veículo por meio de notificações no aplicativo de mensagens no *smartphone*. Essas notificações são geradas no servidor local que envia para o aplicativo. No protótipo foram desenvolvidos no servidor local (AmI da casa) os seguintes módulos:

- Módulo Web Server;
- Módulo de Integração;
- Módulo *Smartphone*.

Esses módulos são capazes de requisitar os dados da nuvem por meio de API e enviar em um formato correto por notificações de mensagens no aplicativo. O *hardware* utilizado para o servidor local foi um *laptop* com as configurações, conforme a Tabela 14.

Tabela 14: Configuração servidor local

Marca	Lenovo Z400
Sistema operacional	Windows 10
Processador	Intel core i5-3320M
CPU	2.60GHz
Memória RAM	8Gb
Tipo de sistema	64 bits

O *smartphone* utilizado foi o ASUS Zenfone 4 MAX com sistema operacional Android. Foi necessário baixar da loja virtual Google Play o aplicativo Telegram que é um aplicativo gratuito de mensagens instantâneas que possibilita criar bots (robôs virtuais) que notificam via mensagens.

O módulo Web Server é responsável por gerar um servidor local no AmI da casa. O *software* utilizado foi o WampServer, o qual instala um conjunto de softwares capaz de gerar um servidor com banco de dados no qual são salvos os dados requisitados da nuvem.

O módulo de Integração é responsável por fazer as requisições dos alertas da nuvem. Esses alertas são coletados por meio da API implementada na nuvem, a qual é acionada quando esse módulo é ativado realizando os procedimentos para coletar e salvar no banco de dados local gerado pelo módulo Web Server.

O módulo de Integração também é responsável por realizar as notificações por meio do aplicativo Telegram, enviando mensagens através do bot configurado para enviar mensagens com o conteúdo dos parâmetros anormais dos sensores selecionados de funcionamento do veículo.

O módulo *smartphone* fecha o ciclo deste sistema, pois possibilita que as mensagens de notificação cheguem ao usuário final. Para que isso aconteça, foi necessário configurar o aplicativo Telegram e criar um bot relacionado ao módulo de Integração que ao coletar os dados de alertas do banco de dados local aciona o bot que automaticamente envia mensagem para o usuário.

Os recursos necessários para o AmI da casa são listados na tabela com as ferramentas utilizadas para implementar cada módulo descrito acima, conforme a Tabela 15.

Tabela 15: Módulos do sistema

AmI da casa (servidor local)		
Módulo	Ferramentas	Funções
Web Server	Wamp Server Node JS Visual Studio Code	Servidor local Banco de dados
Integração	Python Sublime Text	Coleta dados da nuvem Salva no banco de dados local Coleta dados do banco local Notifica pelo Telegram
Smartphone	Telegram	Notifica por meio de bots

Para apresentar melhor o que acontecem no AmI da casa, a Figura 25 mostra a dinâmica dos dados entre o servidor da Nuvem e as iterações no AmI da casa.

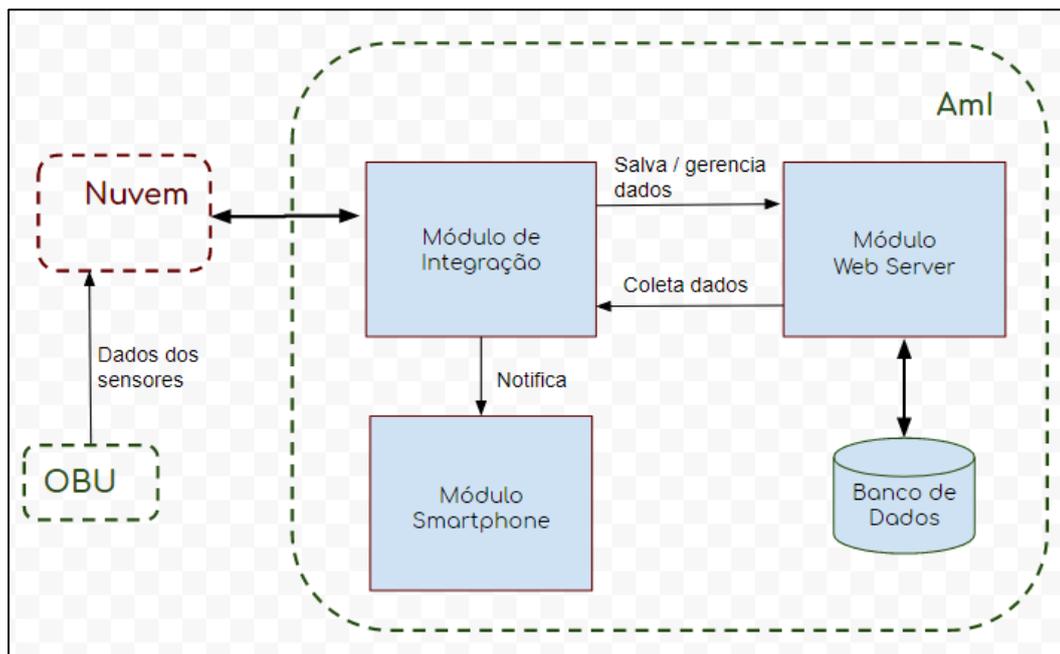


Figura 25: Esboço do fluxo de dados no AmI

A dinâmica inicia quando o módulo de integração requisita dados através da API fornecida pela nuvem (os quatro métodos descritos na seção anterior). Esses dados são retornados em formato JSON e encaminhados para o Web Server, no qual são salvos no banco de dados local e gerenciados no controle de mensagens, ou seja, as mensagens enviadas são marcadas para que não sejam reenviadas.

Uma vez salvos e devidamente marcados como lido e não lido, o módulo de Integração realiza periodicamente consultas ao banco de dados local, caso haja dados de alertas, esse módulo marca o alerta como lido e notifica o módulo *Smartphone*, que envia mensagem de notificação para o Telegram por meio do bot.

5.5 Arquitetura de comunicação

Nessa seção, será apresentada a arquitetura do sistema com ênfase nos dados trocados entre os módulos, bem como o *hardware* utilizado. A figura do usuário será inserida nesse contexto para fechar o ciclo do sistema, de acordo com a Figura 26.

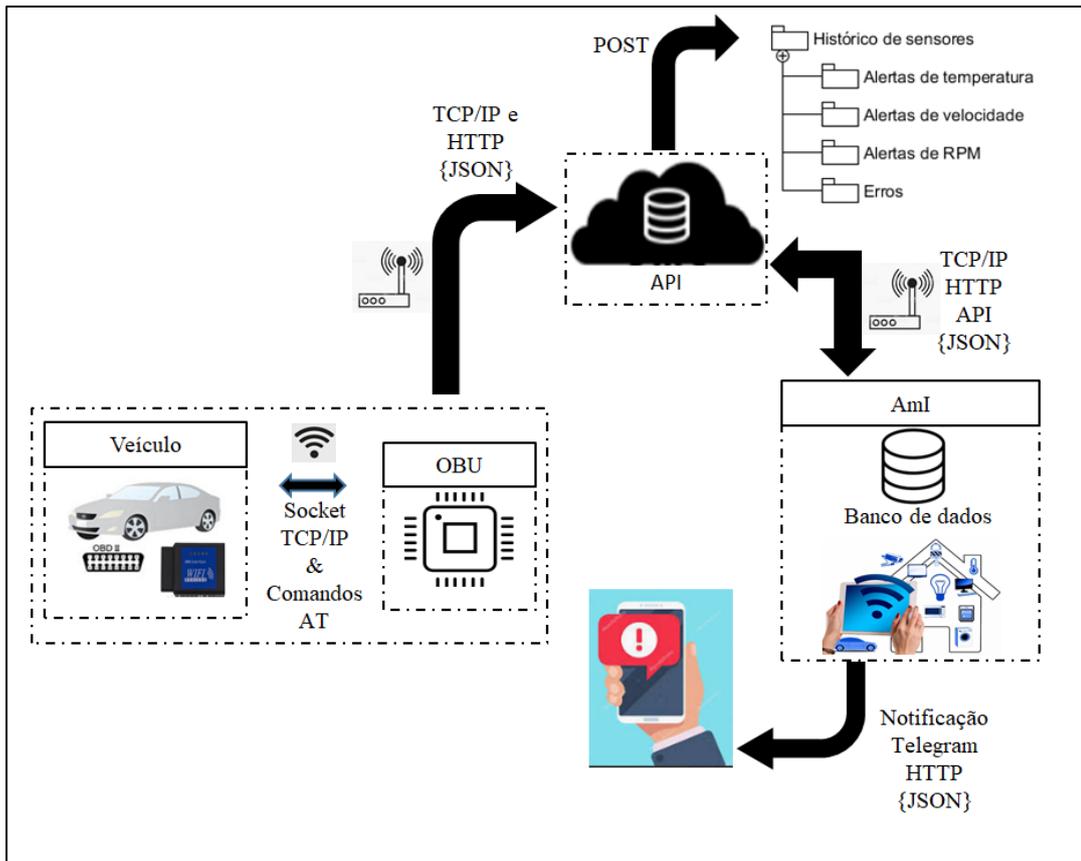


Figura 26: Arquitetura de comunicação e fluxo de dados

A comunicação entre veículo e OBU é feita através do adaptador ELM 327 Wi-Fi que cria uma rede sem fio com o nome WiFi_OBDII. O OBU, formado pelo RPi 3 e os programas implementados e embarcados, conecta-se a essa rede via protocolo TCP/IP, na qual o adaptador é o servidor e o OBU é cliente. O OBU requisita dados dos sensores do veículo enviando comandos AT e recebendo as respostas no formato hexadecimal.

O OBU envia os dados para a nuvem ao se conectar a um roteador (casa, hotel ou *hotspot*), isso acontece de forma automática, ou seja, quando detecta a rede Wi-Fi tenta se conectar e envia os dados para a nuvem utilizando os protocolos TCP/IP e HTTP. Nessa comunicação o OBU formata os dados para JSON.

Na nuvem a API faz a leitura dos dados no formato JSON e armazena no banco de dados. A API filtra os dados dos sensores, rotulando como alertas de temperatura, velocidade, rotação do motor e erros. Esses alertas são disponibilizados por métodos que podem ser acessados por HTTP por requisições que retornam dados no formato JSON dos parâmetros filtrados.

O AmI da casa possui o dispositivo central responsável por gerenciar e distribuir as informações do veículo para que o usuário seja notificado no ambiente. O dispositivo é o

computador portátil conectado à internet com acesso aos dados via protocolo TCP/IP quando envia as requisições em HTTP dos alertas. Os dados são recebidos em formato JSON, são readequados no AmI e salvos no banco de dados local.

Como prova de conceito o AmI da casa envia as notificações sobre os sensores para o aplicativo Telegram pela internet via protocolo HTTP no formato de dados JSON. No aplicativo esses dados são convertidos e podem ser visualizados pelo usuário. Contudo, essas notificações poderão ser visualizadas em outros dispositivos da casa ou até mesmo no ambiente de trabalho do usuário.

5.6 Considerações finais do capítulo

Este capítulo descreveu a implementação de cada módulo da arquitetura, bem como os *softwares*, *hardwares* e a arquitetura de comunicação. Em cada módulo foi apresentada uma visão geral sobre o funcionamento e o detalhamento de cada etapa da implementação.

Os resultados alcançados neste capítulo foram essenciais para realizar os ajustes necessários para melhorar a implementação do sistema e a realização dos experimentos que serão mostrados no capítulo seguinte.

As contribuições deste capítulo para a dissertação foram as integrações e testes seguindo a arquitetura proposta no capítulo anterior e, assim, mostrando que o sistema funciona de acordo com o que foi especificado.

6. Resultados Experimentais e Discussão

Este capítulo tem como objetivo realizar os experimentos como prova de conceito do funcionamento deste sistema, seguindo as especificações e implementações apresentadas nos capítulos 4 e 5. São apresentados os experimentos com o caso de uso para testar o sistema, os resultados obtidos e as discussões.

6.1 Experimentos

Os estudos de caso começaram a ser realizados na bancada do Laboratório de Termociências da Faculdade de Tecnologia (FT) da Universidade Federal do Amazonas (UFAM), a qual simula integralmente os principais dados de funcionamento do veículo, conforme a Figura 27.

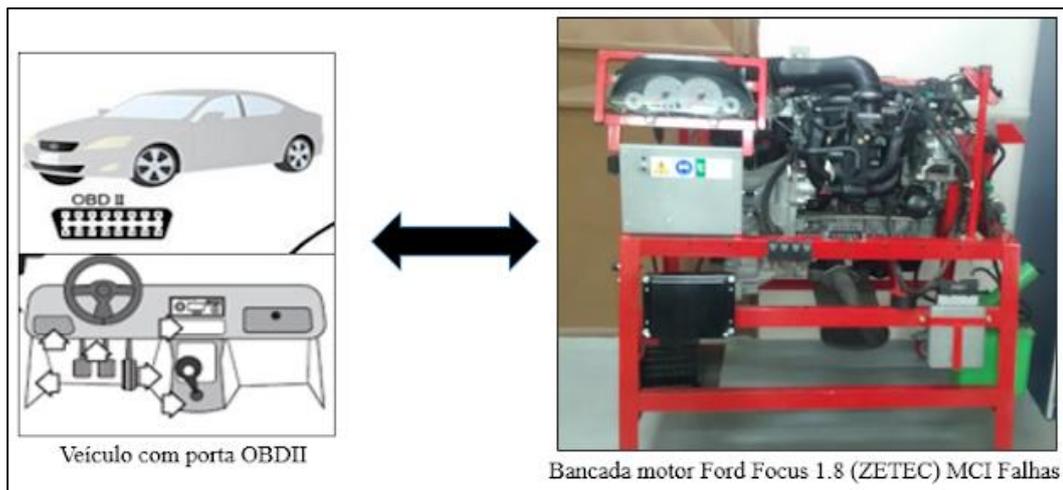


Figura 27: Bancada do motor do laboratório de mecânica

A Bancada Ford Focus 1.8 ZETEC é um motor de veículo de pequeno porte com todos os sistemas essenciais de funcionamento como os sensores e a ECU. É capaz também de produzir 10 tipos comuns de falhas.

Essas falhas são causadas em sensores e atuadores que mantêm o motor funcionando adequadamente. São inseridas manualmente por meio do quadro de falhas e causam o mal funcionamento do motor. O quadro de falhas pode ser visto na Figura 28.



Figura 28: Quadro de inserção de falhas

Conforme a Figura 29 a bancada possui a porta OBD-II, possibilitando a coleta de dados de sensores da ECU do motor.



Figura 29: Porta OBD-II da bancada

Assim foi possível realizar vários ensaios com a bancada do motor de forma segura e com a possibilidade de inserir os erros dos sensores ampliando os testes com o protótipo OBU.

6.2 Resultados Obtidos

O experimento realizado será ilustrado em um cenário representado pela Figura 30. O cenário é dividido em quatro partes que são apresentadas detalhadamente a seguir com os dados do sistema em execução.

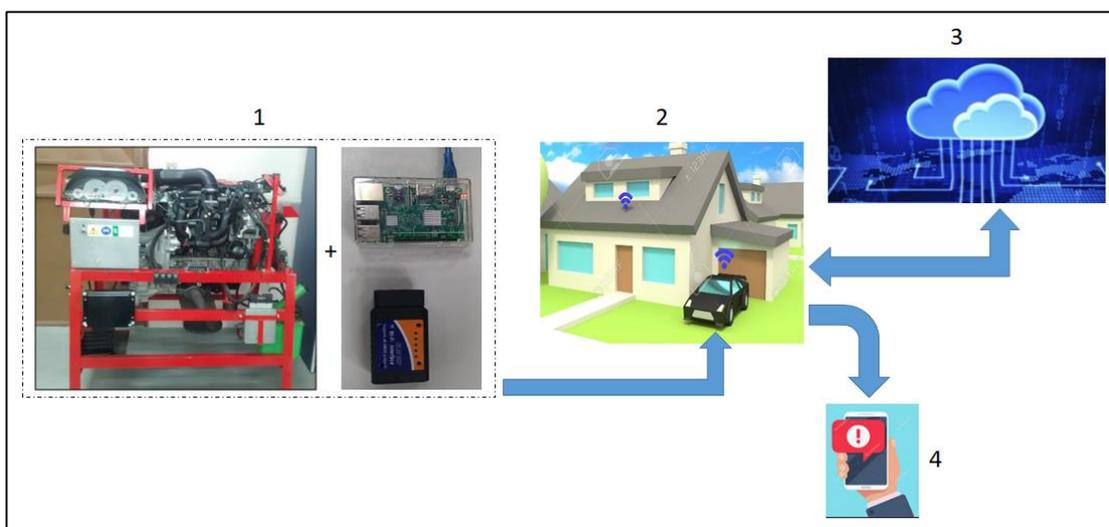


Figura 30: Cenário de utilização do sistema

Parte 1: Bancada do motor + OBU: o OBU coleta dados dos sensores da bancada do motor, sabe-se pelos capítulos anteriores que ao mesmo tempo em que são coletados, os dados dos sensores são convertidos e salvos no banco de dados interno. Esse processo acontece até que o veículo seja desligado e o OBU inicia a procura por uma rede Wi-Fi habilitada para transmitir os dados salvos no seu banco de dados interno.

Parte 2: OBU + Roteador AmI (Gateway) + Nuvem: Ao chegar na casa com AmI ou local com Wi-Fi que o OBU está habilitado em se conectar, a conexão entre veículo e roteador é estabelecida, logo em seguida, os dados são transmitidos para a nuvem (parte 3). É importante salientar que os locais que o OBU pode se conectar pode ser em hotéis, o local de trabalho e até mesmo o *hotspot* gerado pelo *smartphone*, desde que o OBU esteja habilitado a se conectar.

Parte 3: Nuvem + AmI casa: na nuvem os dados são armazenados e classificados em alertas (temperatura de arrefecimento do motor, velocidade, rotação do motor e erros da central), os quais podem ser acessados por meio de APIs e salvos no banco de dados local do AmI da casa, conforme a Figura 31.

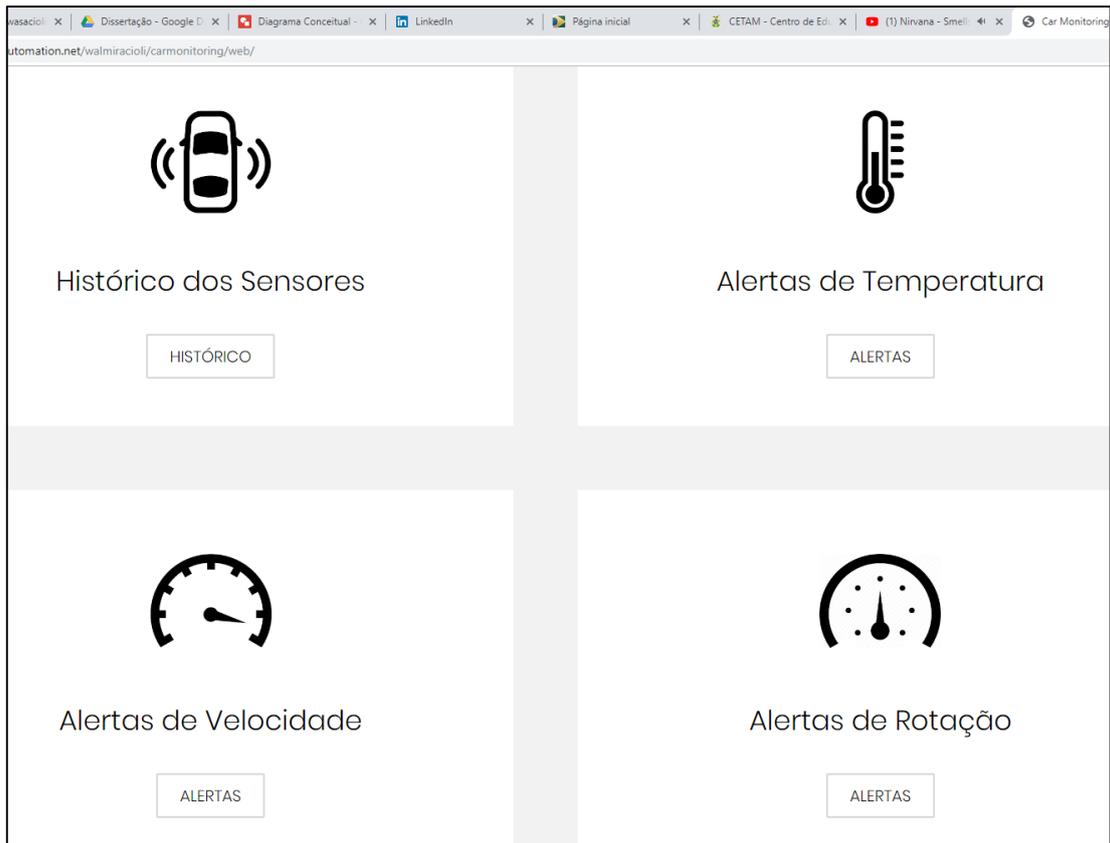


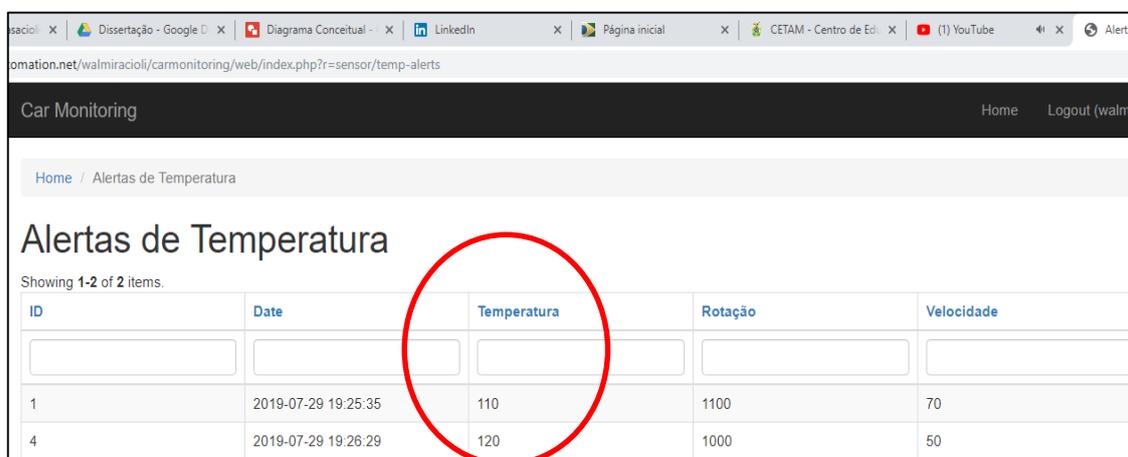
Figura 31: Aplicação web que armazena dados do OBU no servidor na nuvem

A Figura 32 apresenta parte da página web, na qual possui o histórico de sensores e os alertas que são filtrados a partir do histórico. No histórico de sensores são armazenados todos os 14 sensores da bancada do motor, porém são visualizados apenas os quatro parâmetros.

ID	Date	Temperatura	Rotação	Velocidade	
<input type="text"/>					
1	2019-07-29 19:25:35	110	1100	70	
2	2019-07-29 19:25:35	80	2000	100	
3	2019-07-29 19:26:29	60	3500	60	
4	2019-07-29 19:26:29	120	1000	50	

Figura 32: Histórico dos sensores

A Figura 33 mostra os Alertas de Temperatura, no qual são coletados os registros em que a temperatura é superior a 100 graus Celsius.

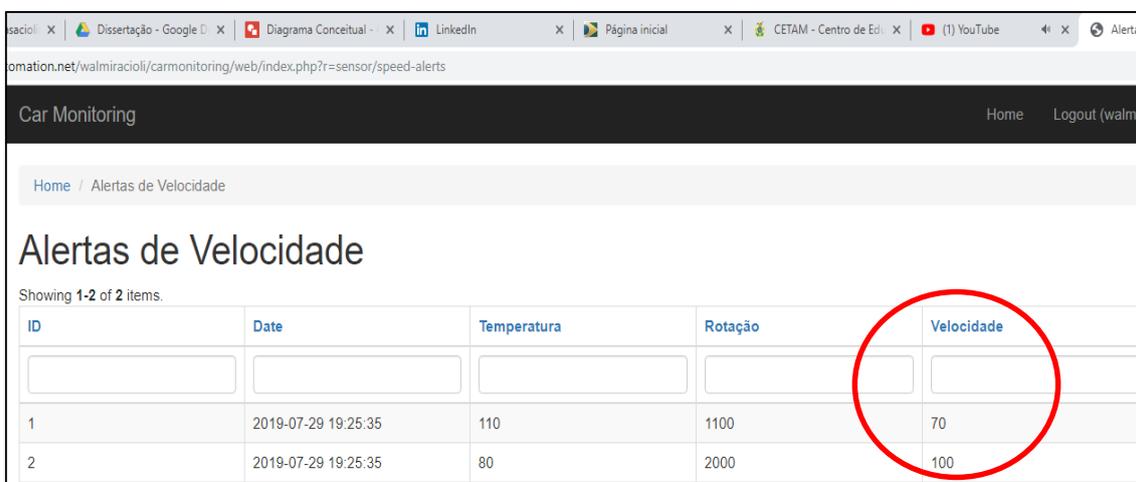


The screenshot shows a web browser window with the URL 'omation.net/walmiracioli/carmonitoring/web/index.php?r=sensor/temp-alerts'. The page title is 'Car Monitoring' and the breadcrumb is 'Home / Alertas de Temperatura'. The main heading is 'Alertas de Temperatura' with a sub-heading 'Showing 1-2 of 2 items.' Below this is a table with the following data:

ID	Date	Temperatura	Rotação	Velocidade
1	2019-07-29 19:25:35	110	1100	70
4	2019-07-29 19:26:29	120	1000	50

Figura 33: Alertas de temperatura

A Figura 34 mostra os Alertas de Velocidade, no qual são coletados registros em que a velocidade é superior a 60 km/h.



The screenshot shows a web browser window with the URL 'omation.net/walmiracioli/carmonitoring/web/index.php?r=sensor/speed-alerts'. The page title is 'Car Monitoring' and the breadcrumb is 'Home / Alertas de Velocidade'. The main heading is 'Alertas de Velocidade' with a sub-heading 'Showing 1-2 of 2 items.' Below this is a table with the following data:

ID	Date	Temperatura	Rotação	Velocidade
1	2019-07-29 19:25:35	110	1100	70
2	2019-07-29 19:25:35	80	2000	100

Figura 34: Alertas de velocidade

A Figura 35 mostra os Alertas de Rotação, no qual são coletados registros em que a rotação do motor é superior ou igual a 2000.

The screenshot shows a web browser window with several tabs open. The active page is titled 'Car Monitoring' and displays a table of rotation alerts. The table has five columns: ID, Date, Temperatura, Rotação, and Velocidade. The 'Rotação' column is circled in red. The table contains two rows of data.

ID	Date	Temperatura	Rotação	Velocidade
2	2019-07-29 19:25:35	80	2000	100
3	2019-07-29 19:26:29	60	3500	60

Figura 35: Alertas de Rotação

A página web que mostra os dados dos sensores é um ambiente de teste e experimentos com acesso restrito com login e senha, logo não será visualizada pelo usuário. A página web foi utilizada no desenvolvimento para realizar testes com as APIs implementadas na nuvem.

Parte 4 – AmI casa + *smartphone* + usuário: no AmI da casa há um banco de dados local que coleta dados da nuvem. Esse banco de dados local é consultado com frequência por um *script* responsável por detectar as alterações dos dados e enviar mensagens no AmI da casa por meio do aplicativo do *smartphone*. A Figura 36 apresenta a tela do aplicativo Telegram com os usuários (a) e as notificações (b).

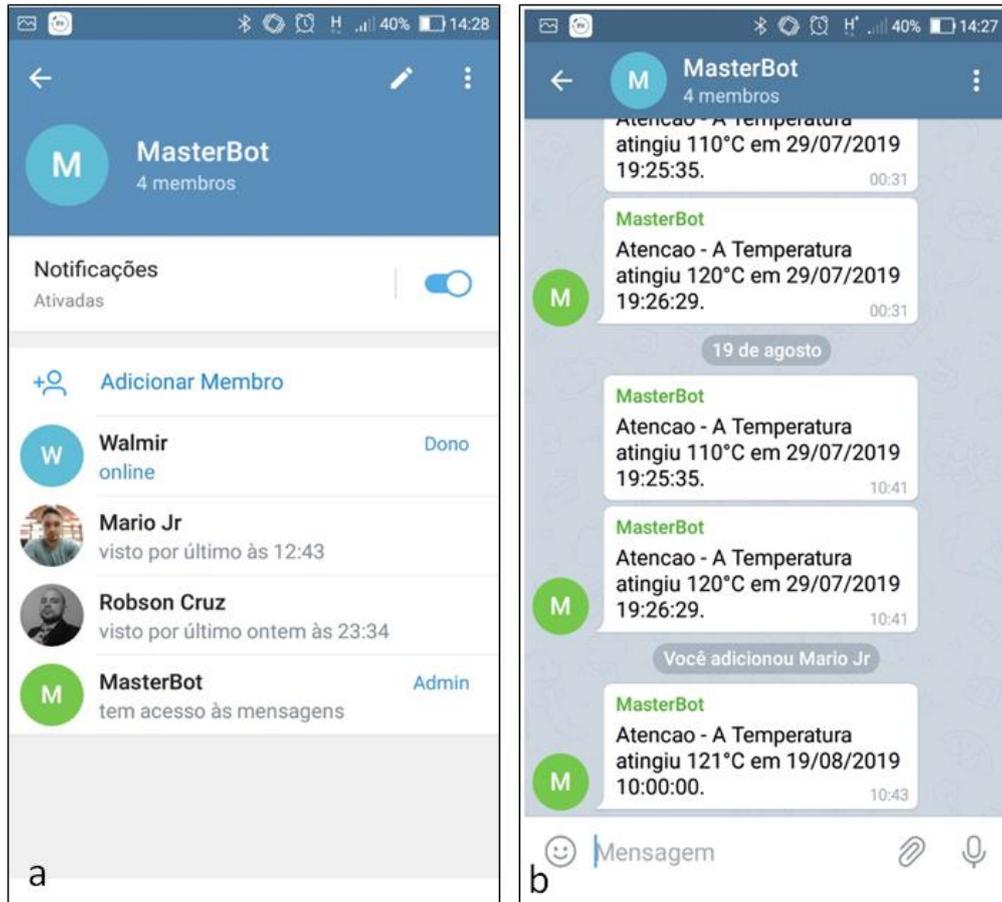


Figura 36: Notificações via aplicativo

Na Figura 36(a) os membros do grupo do aplicativo Telegram podem ser notificados pelo *MasterBot* que recebe do AmI da casa dados anormais de funcionamento do veículo. A temperatura de arrefecimento, velocidades e rotações acima dos limites recomendados que podem indicar mal-uso do veículo.

A Figura 36(b) mostra as notificações que contêm a identificação do sensor, o valor, a data e hora do parâmetro coletado. Portanto é uma ferramenta útil no monitoramento do uso do veículo dependendo dos sensores coletados.

6.3 Discussão

Os trabalhos relacionados aos AmIs enfatizaram a automação de tarefas do ambiente da casa. Outros criticaram que um AmI não é simplesmente automatizar tarefas, e sim utilizar técnicas computacionais para detectar padrões e se adequar ao usuário. Em ambos os casos, o ambiente era cercado de sensores, atuadores, com poder computacional de processamento e redes de comunicação entre todos os dispositivos.

Os trabalhos relacionados à comunicação veicular focavam nas tecnologias mais atuais de comunicação na qual o veículo para tudo (v2x) dá a ideia de que o veículo pode se conectar a qualquer objeto com internet. Mas na prática esses objetos eram equipamentos dedicados das rodovias e cidades.

No entanto, foi verificado que o surgimento de conceitos novos, como a Internet dos Veículos (IoV), os veículos de fato estão conectados à internet. Ainda nos trabalhos relacionados, a maioria apresentou uma arquitetura em que o *smartphone* foi o principal mediador entre o veículo e as aplicações propostas.

Nos poucos trabalhos que utilizavam dispositivos dedicados para coletar dados nos veículos, nenhum tinha o objetivo de enviar dados de forma automática para algum ambiente. Este sistema utiliza um dispositivo dedicado para coletar, armazenar e estabelece conexão à internet de forma automática.

O sistema possui um servidor que recebe os dados do veículo, o qual é acessado periodicamente pela aplicação do AmI da casa que ao detectar novas mensagens notifica por meio de mensagens direta ao usuário por meio do aplicativo de mensagens do *smartphone*.

Os resultados dos experimentos mostraram que o sistema funciona e contribui para uma nova abordagem que conecta o veículo com o AmI da casa de forma automática e transparente ao usuário que recebe mensagens de notificação pelo aplicativo de mensagens.

Este sistema alcançou os resultados esperados dentro do escopo definido. Contudo, há alguns pontos que foram mapeados, mas que não foi possível adicionar no presente protótipo, pois não haveria tempo hábil para finalizar esta dissertação e são situações que demandariam mais tempo de estudos. Esses pontos serão abordados mais detalhadamente no capítulo a seguir.

7. Conclusão

O presente trabalho teve como objetivo desenvolver um sistema capaz de integrar um veículo a um Ambiente Inteligente que se conecta de forma automática a um ponto de internet sem fio e envia dados à nuvem permitindo que o usuário seja notificado sobre os parâmetros de funcionamento do veículo por meio de dispositivos conectados a este ambiente.

A concepção da solução foi formada a partir da análise dos trabalhos relacionados, buscando-se sempre usar tecnologias relacionadas com base no objetivo desta pesquisa e aplicando a ideia de que o veículo faz parte do AmI da casa, realizando os procedimentos de coleta, transmissão, processamento e envio de notificações de forma automática. Foi construída uma arquitetura composta pelo OBU, servidor na nuvem e servidor local.

Na arquitetura deste sistema foram utilizadas diferentes técnicas, ferramentas, metodologias e procedimentos, como: comunicação de diferentes plataformas, implementação de *softwares* de integração, processamento de dados, *web services*, conceitos de redes de computadores, aplicações web e sistemas embarcados.

A construção do OBU e a integração dos blocos foram os desafios que tiveram mais destaque neste sistema. Pois, para o OBU foram necessários tempo de estudo para verificar os métodos mais adequados de coleta e transmissão dos dados do veículo, uma vez que o dispositivo fica instalado no veículo e deve realizar os procedimentos de forma autônoma. Portanto, os códigos implementados devem ser tolerantes a falhas e robustos.

Na integração dos blocos do sistema foi necessário que cada bloco tivesse o mesmo formato de mensagens de comunicação, do contrário, o OBU não conseguiria enviar os dados para o servidor na nuvem e conseqüentemente o servidor local no AmI da casa não conseguiria coletar as informações e distribuir nos dispositivos conectados a rede da casa.

O sistema proposto neste trabalho testou a hipótese H1 apresentada no Capítulo 1, “é possível integrar um veículo a um AmI da casa que por meio de dispositivos conectados à rede desse ambiente, seja capaz de notificar o usuário sobre parâmetros anormais de funcionamento do veículo”, foi testada e avaliada no decorrer do trabalho”.

Durante a construção da arquitetura do sistema e dos protótipos, foi possível responder parte do problema de pesquisa que tratava sobre a comunicação entre veículo e ambiente necessária no desenvolvimento para integrar esses dois.

Na validação foi verificado empiricamente que o conceito proposto de integrar o veículo ao AmI da casa foi atendido de acordo com os objetivos da pesquisa e notificou o

usuário sobre os parâmetros incomuns de funcionamento do motor por meio de mensagens no aplicativo Telegram do *smartphone*, e assim, contemplando a última parte da hipótese H1.

7.1 Contribuições

Como contribuições deste trabalho citam-se: um sistema modularizado formado por três partes que são o OBU, Servidor na nuvem e AmI da casa;

- Construção do OBU, dispositivo instalado no veículo responsável por coletar e transmitir de forma automática os dados ao se conectar com a rede sem fio;
- Mescla de características dos trabalhos de redes veiculares e AmI;
- Concepção de uma arquitetura capaz de integrar veículo, servidor na nuvem e ambiente apresentado no Capítulo 4;
- Sistema capaz de notificar o usuário por meio de mensagens de aplicativo do *smartphone* ligado ao AmI da casa;
- Sistema ubíquo, ou seja, sistema transparente ao usuário;
- Veículo conectado com a internet;
- A integração do sistema por meio de módulos planejados na arquitetura;
- Sistema que integra veículo, nuvem, ambiente e usuário por meio de dispositivos;

7.2 Limitações e Trabalhos Futuros

A principal limitação deste trabalho é a segurança dos módulos do sistema uma vez que há transmissão de dados entre veículo, servidor na nuvem e o servidor local do ambiente da casa. A segurança dos dados é um assunto crítico, pois ataques de terceiros são comuns nos dias atuais com a disseminação da internet das coisas.

Outra limitação é quanto à infraestrutura do ambiente. Apesar do crescimento do acesso a internet em áreas urbanas a infraestrutura computacional não está totalmente presente em todos os lares, o que gera um fator limitante para a aplicação deste trabalho.

Para este trabalho foi utilizado um computador portátil para ser o nó central do ambiente e fazer a ponte entre o servidor na nuvem e os robôs virtuais configurados no aplicativo de mensagens do Telegram no *smartphone*.

Como trabalhos futuros, propõe-se desenvolver mecanismos de segurança na comunicação, possibilitar que outros dispositivos da casa possam notificar o usuário e inserir outros parâmetros de sensores para serem notificados.

Outros pontos para trabalhos futuros é o início das pesquisas para utilizar manutenção preditiva utilizando a análise de um número maior de dados dos sensores, gerar relatórios de desempenho semanais, mensais ou anuais e outros serviços visando unir as fabricantes dos veículos, oficinas e autopeças de uma forma que o sistema possa indicar os possíveis locais de conserto com preços fixados e agendados.

Referências

- Amaral, G. L., Yazbek, C. L., Olenike, J. E. Instituto Brasileiro de Planejamento e Tributação e Empresômetro Tecnologias LTDA. São Paulo. 2018. Disponível em < http://www.automotivbusiness.com.br/abinteligencia/pdf/estudo_frota_completo.pdf>. Acesso em: 27 ago. 2019.
- Amarasinghe, M. *et al.* (2015) “Cloud-based driver monitoring and vehicle diagnostic with OBD2 telematics”, *IEEE International Conference on Electro Information Technology*, 2015-June, p. 505–510. doi: 10.1109/EIT.2015.7293433.
- Ang, L., Member, S. e Seng, K. A. H. P. (2019) “Deployment of IoV for Smart Cities : Applications , Architecture , and Challenges”, *IEEE Access*. IEEE, 7, p. 6473–6492. doi: 10.1109/ACCESS.2018.2887076.
- Atzori, L. *et al.* (2018) “Towards the implementation of the Social Internet of Vehicles”, *Computer Networks*. Elsevier B.V., 147, p. 132–145. doi: 10.1016/j.comnet.2018.10.001.
- Bhargava, K., Mcmanus, G. e Ivanov, S. (2017) “Fog-centric Localization for Ambient Assisted Living”, p. 1424–1430.
- Chen, S. H., Pan, J. S. J. e Lu, K. (2015) “Driving behavior analysis based on vehicle OBD information and adaboost algorithms”, *Lecture Notes in Engineering and Computer Science*, 1, p. 102–106. Available at: http://www.scopus.com/inward/record.url?eid=2-s2.0-84938068441&partnerID=tZOtx3y1%5Cnhttp://www.iaeng.org/publication/IMECS2015/IMECS2015_pp102-106.pdf.
- Coulouris, G., Dollimore, J. e Kindberg, T. (2012) *Distributed Systems: Concepts and Design, Computer*. Available at: <http://www.amazon.com/dp/0321263545>.
- ELM327 Datasheet. Allcomponent.com. United State of America (USA). Disponível em: < <https://www.alldatasheet.com/datasheet-pdf/pdf/542978/ELM/ELM327.html>>. Acesso em: 28 de ago. de 2019.
- Frey, J. (2013) “AdAPT - A dynamic Approach for Activity Prediction and Tracking for Ambient Intelligence”, *2013 9th International Conference on Intelligent Environments*. IEEE, (i), p. 254–257. doi: 10.1109/IE.2013.38.
- Greening, P. (2001) “European Vehicle Emission Legislation—Present and Future”, *Topics in Catalysis*, 16–17(1), p. 5–13. Available at: <http://dx.doi.org/10.1023/A:1016629326634>.
- Husniah, H., Pasaribu, U. S. e Iskandar, B. P. (2015) “Dimensional Lease Contract”, *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, p. 1067–1071. doi: 10.1109/IEEM.2015.7385812.
- Jezzini, A., Ayache, M. e Elkhansa, L. (2013) “Effects of Predictive maintenance (PdM), Proactive Maintenance (PoM) & Preventive Maintenance (PM) on minimizing the faults in medical instruments .”, *2013 2nd International Conference on Advances in Biomedical Engineering*. IEEE, p. 53–56. doi: 10.1109/ICABME.2013.6648845.
- Kiening, A. *et al.* (2013) “Trust Assurance Levels of Cybercars in V2X Communication”, p. 49–60.
- Khanapuri, A. V., Shastri, A., D'souza, G. and D'souza, S. "On road: A car assistant application," *2015 International Conference on Technologies for Sustainable Development (ICTSD)*, Mumbai, 2015, pp. 1-7.
- Krishnan, L. A. e Kathpalia, S. S. (2002) “Literature Reviews in Student Project Reports”, *IEEE Transactions on Professional Communication*. IEEE, 45(3), p. 187–197. doi:

10.1109/TPC.2002.801637.

Lalli, F. Tecnologia para diagnóstico. São Paulo. 2015. O mecânico. Disponível em: <<http://omecanico.com.br/tecnologia-para-diagnosticos/>>. Acesso em: 24 set. 2019.

Magana, V. C. e Munoz-Organero, M. (2016) “Artemisa: A Personal Driving Assistant for Fuel Saving”, *IEEE Transactions on Mobile Computing*, 15(10), p. 2437–2451. doi: 10.1109/TMC.2015.2504976.

Manavella, H. J. HM Autotrônica. [S.l.]. Disponível em: <<http://www.hmautotron.eng.br/zip/cap19-hm004web.pdf>>. Acesso em: 24 jun. 2019.

Malekian, R. *et al.* (2017) “Design and Implementation of a Wireless OBD II Fleet Management System”, 17(4), p. 1154–1164.

Marin, M. *et al.* (2017) “Hardware and Software Approach for Teaching Automotive Networks”, *Electric Vehicles International Conference & Show*, p. 7–10. doi: 10.1109/EV.2017.8242097.

Meseguer, J. E. *et al.* (2017) “Drivingstyles: A mobile platform for driving styles and fuel consumption characterization”, *Journal of Communications and Networks*, 19(2), p. 162–168. doi: 10.1109/JCN.2017.000025.

Montori, F. *et al.* (2018) “Achieving IoT interoperability through a service oriented in-home appliance”, *2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings*, 2018-Janua, p. 1–6. doi: 10.1109/GLOCOM.2017.8255044.

Morabito, R. *et al.* (2017) “Lightweight Virtualization as Enabling Technology for Future Smart Cars Lightweight Virtualization as Enabling Technology for Future Smart Cars”, (March). doi: 10.23919/INM.2017.7987466.

Moseley, R. "Creating an ambient intelligence network using insight and merged reality technologies," *2017 Computing Conference*, London, 2017, pp. 469-474.

Naik, G., Member, S. e Choudhury, B. (2019) “IEEE 802 . 11bd & 5G NR V2X : Evolution of Radio Access Technologies for V2X Communications”, *IEEE Access*. IEEE, 7, p. 70169–70184. doi: 10.1109/ACCESS.2019.2919489.

Nemati, M. e Seyedtabaai, S, "On-board diagnosis of vehicles cooling system," *2014 22nd Iranian Conference on Electrical Engineering (ICEE)*, Tehran, 2014, pp. 1317-1322.

Obdii, W. H. Y. *et al.* (1996) “OBDII: PAST , PRESENT & FUTURE”, (Mil), p. 1–5. Available at: http://www.autotap.com/Articles/OBDII_Past_Present_Future.pdf.

Obukata, R., Oda, T. e Barolli, L. (2016) “Design of an Ambient Intelligence Testbed for Improving Quality of Life”, *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE, p. 714–719. doi: 10.1109/WAINA.2016.148.

Prytz, R. (2014) *Machine learning methods for vehicle predictive maintenance using off-board and on-board data*.

Putthapipat, P., Woralert, C. e Sirinimnuankul, P. (2018) “Speech recognition gateway for home automation on open platform”, *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, p. 1–4. doi: 10.23919/ELINFOCOM.2018.8330715.

Ramos, C., Juan, P. e Augusto, C. (2008) “Ambient Intelligence — the Next Step for Artificial Intelligence”, p. 15–18.

- Sa'adah, U., Akhmad, J. and Hisyam, M. "Implementing Singleton method in design of MVC-based PHP framework," *2015 International Electronics Symposium (IES)*, Surabaya, 2015, pp. 212-217.
- Sadri, F. (2011) "Ambient intelligence", *ACM Computing Surveys*, 43(4), p. 1–66. doi: 10.1145/1978802.1978815.
- Safronov, M.; WINESETT, J. *Web Application Development with Yii 2 and PHP*.
- Sampaio, D., Reis, L. P. e Rodrigues, R. (2012) "A survey on Ambient Intelligence projects", *Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on*, p. 1–6..
- Sawant, R. (2018) "(OBD) Device for Cars", *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE, p. 1–4.
- Seabra, J. C., Costa, M. A. e Lucena, M. M. (2016) "IoT based intelligent system for fault detection and diagnosis in domestic appliances", *2016 IEEE 6th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, p. 205–208. doi: 10.1109/ICCE-Berlin.2016.7684756.
- Seif, A. A. e El-Saber, N. (2017) "Scalable Distributed-Computing IoT Applied Architecture with Semantic Interoperable Gateway", *Proceedings of the 3rd Africa and Middle East Conference on Software Engineering - AMECSE '17*, p. 43–44. doi: 10.1145/3178298.3178306.
- Shi, G. *et al.* (2016) "A vehicle electric control unit over-the-air reprogramming system", *2015 International Conference on Connected Vehicles and Expo, ICCVE 2015 - Proceedings*, p. 48–51. doi: 10.1109/ICCVE.2015.21.
- Siegel, J. *et al.* (2014) "Vehicular engine oil service life characterization using on-board diagnostic (OBD) sensor data", *Proceedings of IEEE Sensors, 2014-Decem(December)*, p. 1722–1725. doi: 10.1109/ICSENS.2014.6985355.
- Silva, M. *et al.* (2019) "A Customer Feedback Platform for Vehicle Manufacturing Compliant with Industry 4 . 0 Vision", 18(10), p. 1–24. doi: 10.3390/s18103298.
- Silva, V. J. *et al.* (2016) "Intelligent System for Medication Management in Residential Environments", *IFAC-PapersOnLine*. Elsevier B.V., 49(30), p. 171–174. doi: 10.1016/j.ifacol.2016.11.162.
- Srinivasan, A. "IoT Cloud Based Real Time Automobile Monitoring System," *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, Singapore, 2018, pp. 231-235.
- Soultanopoulos, T. *et al.* (2016) "Internet of Things data management in the cloud for Bluetooth Low Energy (BLE) devices", *Proceedings of the Third International Workshop on Adaptive Resource Management and Scheduling for Cloud Computing - ARMS-CC'16*, p. 35–39. doi: 10.1145/2962564.2962568.
- Vivek, G. V. e Sunil, M. P. (2016) "Enabling IOT services using WIFI - ZigBee gateway for a home automation system", *Proceedings of 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks, ICRCICN 2015*, p. 77–80. doi: 10.1109/ICRCICN.2015.7434213.
- Wallace, B. *et al.* (2015) "Automation of the Validation, Anonymization, and Augmentation of Big Data from a Multi-year Driving Study", *Proceedings - 2015 IEEE International Congress on Big Data, BigData Congress 2015*, 1, p. 608–614. doi: 10.1109/BigDataCongress.2015.93.

Wang, R., Ma, H. e Fu, J. (2019) “Research on Joint Optimization of Domestic Garbage Collection and Transportation Vehicle Dispatching and Preventive Maintenance”, *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*. IEEE, p. 295–299.

Yacchirema, D. C. e Member, S. (2016) “Smart IoT Gateway For Heterogeneous Devices Interoperability”, 14(8), p. 3900–3906. doi: 10.1109/TLA.2016.7786378.

Yachir, A. *et al.* (2016) “Event-Aware Framework for Dynamic Services Discovery and Selection in the Context of Ambient Intelligence and Internet of Things”, *IEEE Transactions on Automation Science and Engineering*. IEEE, 13(1), p. 85–102. doi: 10.1109/TASE.2015.2499792.

Zhang, P., *Advanced Industrial Control Techology*. 1st Edition. USA: ELSEVIER , 2010, pp. 455-460.