CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC	
(X) PRÉ-PROJETO ( ) PROJETO	ANO/SEMESTRE: 2017.1

APLICAÇÃO PARA MONITORAMENTO VEICULAR E GEOLOCALIZAÇÃO EM TEMPO REAL

Maicon Machado Gerardi da Silva

Prof. Miguel Alexandre Wisintainer – Orientador

1 INTRODUÇÃO

Nos sete primeiros meses de 2016 foram roubados/furtados em Santa Catarina 3.164 carros e pick-ups nacionais e importados com seguro segundo Odega (2016). Conforme os dados da SUSEP (Superintendência de Seguros Privados), só no primeiro semestre de 2016 as seguradoras registraram 118 mil veículos roubados/furtados no Brasil (ODEGA, 2016).

Odega (2016) cita os veículos com mais índices de roubo, são eles:

a) Chevrolet Celta 1.0 com 6.055 ocorrências de um total de 170.524 veículos segurados;
b) VW Volkswagen Gol 1.0 com 4.514 ocorrências de 253.594 veículos com seguro;
c) Fiat Palio 1.0 com 4.127 ocorrências de um total de 219.654 com seguro.

No ano de 2017, o número de roubos a veículos aumentou 20% em Ribeirão Preto no estado de São Paulo. Segundo G1 Ribeirão e Franca (2017) "[...] 60 roubos de carros e motocicletas ocorreram durante janeiro de 2017. Ao todo, 50 casos ocorreram durante os primeiros trinta dias de 2016 [...]".

Entre janeiro e outubro de 2014 foram registrados pouco mais de meio milhão de veículos que ficaram parados nos mais de 6 mil quilômetros de rodovias do Programa de Concessões Rodoviárias do Estado de São Paulo por apresentarem problemas de manutenção, dentre os quais pneu furado e superaquecimento do motor (SOUZA, 2016). Essa estatística equivale à um pouco mais de 83 carros parados por quilômetro nesse período. Uma pesquisa realizada pelo Instituto Scaringella de Trânsito aponta que a falta de manutenção preventiva no automóvel é relacionada com 30% dos acidentes rodoviários e urbanos no Brasil (CZERWONKA, 2016). Czerwonka (2016) também aponta que a manutenção preventiva do veículo não só beneficia a segurança no trânsito, mas também o bolso. Cuidar do carro antes que alguma peça apresente defeito custa, em média 30% a menos do que fazer somente a checagem de rotina.

Diante desse cenário, Baumgarten (2016) desenvolveu um dispositivo que possibilitasse o rastreamento veicular através de geolocalização e uma imagem capturada através de uma câmera acoplada neste dispositivo. Paralelamente à Baumgarten (2016), Staroski (2016) desenvolveu um protótipo de software embarcado em uma placa Raspberry Pi

para capturar dados da porta OBD (*On-Board Diagnostic*) de um veículos e disponibilizá-los em uma página *web*.

Com base nesses argumentos, propõe-se integrar a aplicação desenvolvida por Baumgarten (2016) e o protótipo de Starosky (2016) em uma única plataforma através de um software embarcado em uma placa Raspberry Pi Zero W para capturar a geolocalização de um veículo, imagens do veículo e os dados de sua porta OBD. Será desenvolvida uma aplicação *mobile* para disponibilizar os dados do software embarcado.

1.1 OBJETIVOS

O objetivo deste trabalho é integrar a aplicação desenvolvida por Baumgarten (2016) e o protótipo de Starosky (2016) em uma única solução através de um software embarcado em uma placa Raspberry Pi Zero W para coletar a geolocalização, imagens e dados da porta OBD de um automóvel, bem como uma aplicação *mobile* para capturar as informações desse software embarcado.

Os objetivos específicos são:

a) desenvolver a integração entre Raspberry Pi Zero W e um módulo Global Positioning System (GPS), um módulo Bluetooth OBD e uma câmera;
b) desenvolver um sistema servidor, que irá receber os dados coletados pelo Raspberry Pi Zero W e armazenar os mesmos;
c) desenvolver uma aplicação *mobile* onde será possível verificar a localização atual, últimas localizações do veículo, capturar imagens e disponibilizar informações da porta OBD;
d) notificar o usuário sobre falhas no motor retornados pela porta OBD.

2 TRABALHOS CORRELATOS

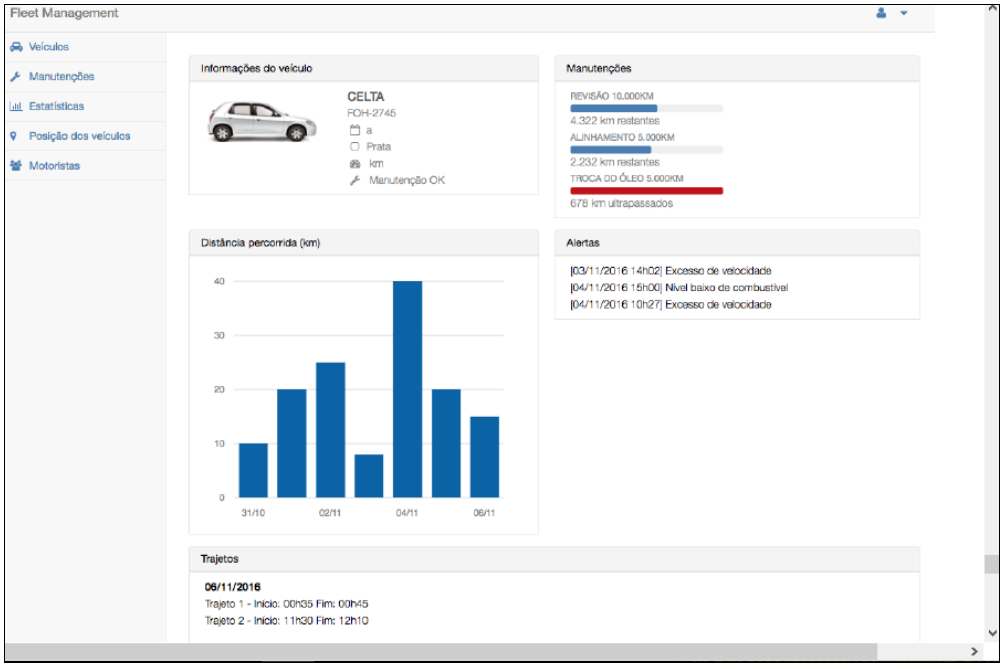
São apresentados três trabalhos correlatos que possuem características semelhantes à proposta deste trabalho. A seção 2.1 apresenta um trabalho de conclusão de curso da Universidade Federal de São Paulo denominado Monitoramento e Gestão de uma Frota de Veículos Utilizando Sistemas Embarcados desenvolvido por Pacheco (2016). A seção 2.2 trata de uma tese de mestrado desenvolvida no Instituto Superior de Engenharia do Porto por Pina (2015). Por fim, na seção 2.3 é apresentada uma aplicação denominada OnStar desenvolvida pela OnStar (2017) utilizada em automóveis da marca GM Chevrolet.

2.1 GESTÃO DE FROTA DE VEÍCULOS UTILIZANDO SISTEMAS EMBARCADOS

Segundo Pacheco (2016), o objetivo do trabalho é o desenvolvimento de um sistema embarcado para o monitoramento de veículos integrado à uma plataforma *web* para o gerenciamento de frota de veículos. Foi desenvolvido um sistema embarcado construído para uma placa Raspberry Pi, um adaptador OBD e um módulo GPS. A placa Raspberry Pi é um computador que utiliza o sistema operacional Linux que ocupa o papel central do sistema. Através dela é possível coletar os dados do adaptador OBD e do módulo GPS. Além disso, ela transmite os dados ao servidor que hospeda a uma plataforma *web*.

Pacheco (2016) desenvolveu um sistema embarcado tal que, dentro do contexto de gestão de frotas, pudesse coletar informações através da porta OBD tais como: velocidade do veículo, rotação do motor, carga do motor, distância percorrida e geolocalização. Essas informações foram coletadas para analisar como o veículo é conduzido (conforme Figura 1). O monitoramento de velocidade pode indicar se os limites de velocidade são respeitados e se há acelerações e frenagens bruscas. Valores de rotação muito elevados também seriam detectados. A carga do motor está associada ao consumo de combustível. A distância percorrida para o agendamento sem a necessidade de consultar o odômetro, a Figura 1 mostra na parte de manutenções essas informações prévias de quilometragem para manutenção, troca de óleo e alinhamento do veículo. Por fim, utiliza a geolocalização para registrar os trajetos realizados.

Figura 1 – detalhes sobre o veículo



fonte: Pacheco (2016, p. 70)

Estes dados são coletados e armazenados no cartão de memória da placa Raspberry Pi. Para isso foi considerado um intervalo de amostragem para essa coleta. Após a coleta dos dados, eles são enviados à um servidor que hospeda a plataforma de gestão de frotas. O envio de dados é feito através de uma conexão Wi-Fi no momento que o veículo retorna à garagem.

Pacheco (2016) utilizou como linguagem de programação Python orientado à objetos por ser uma linguagem bastante versátil. Ele cita que Python é uma linguagem de programação que tem uma quantidade considerável de bibliotecas disponíveis e podem ser desenvolvidas aplicações tais como: aplicações *web*, cálculo científico e numérico, gráficos, *Machine Learning*, *Data Science*, visualização de dados, interfaces gráficas, entre outras. Foram utilizadas neste trabalho bibliotecas que permitissem a criação de *threads*, o acesso ao banco de dados SQLite, o uso de expressões regulares e o *back-end* de um servidor. Para o banco de dados, foi utilizado o SQLite por ser possível executar comandos SQL, salvar e fazer alterações em registros.

Para comandar o sistema embarcado, foi escolhida a placa Raspberry Pi 3 Model B que funciona com o sistema operacional Linux. Possui *bluetooth* e uma porta serial, podendo assim, comunicar-se com o adaptador OBD e com o módulo GPS.

O adaptador OBD é o dispositivo utilizado para a comunicação com o computador de bordo do carro e é instalado diretamente na porta OBD do veículo. A troca de dados entre a Raspberry Pi é realizada através de *bluetooth*. O adaptador utilizado é baseado no circuito integrado ELM327. O ELM327 é um interpretador multiprotocolo projetado para funcionar com todos os protocolos automotivos de veículos previstos na especificação OBD-II. Este dispositivo custa aproximadamente 30 reais. Foi utilizado também o módulo GPS GY-NEO6VM2 para determinar a posição dos veículos utilizando a porta serial para fazer a comunicação dele com a placa Raspberry Pi.

Como pontos positivos a solução mostrou-se eficiente na coleta de dados, pois irá trabalhar com informações atualizadas lidas direto dos sensores do veículo, bem como elimina a necessidade da leitura constante do odômetro do veículo. O esforço das empresas que possuem um número elevado de veículos, diminui pelo fato do sistema executar a leitura dos sensores e também melhora a capacidade de detecção para manutenção da frota (PACHECO, 2016).

Pacheco (2016) sugere que sejam implementados soluções para veículos pesados, pois a solução abrange somente veículos de passeio, em razão de que a interface do sistema OBD é diferente da implementada por ele. Também destaca a necessidade da portabilidade da plataforma *web* para um sistema *mobile*. Sugere-se também, uma redução de custo da placa,

utilizando ao invés da placa Raspberry Pi 3 (que custa 35 dólares aproximadamente), uma placa Raspberry Pi Zero (que custa 5 dólares) que atende a especificação e ainda consome menos energia.

2.2 SISTEMA DE LOCALIZAÇÃO DE VEÍCULOS PARA SMARTPHONE ANDROID

Pina (2015) cita que o trabalho consiste no desenvolvimento de um sistema de localização de veículos para *smartphone* Android. Para isso, foram desenvolvidas duas aplicações: uma aplicação de localização Android e uma aplicação *web* para monitoramento.

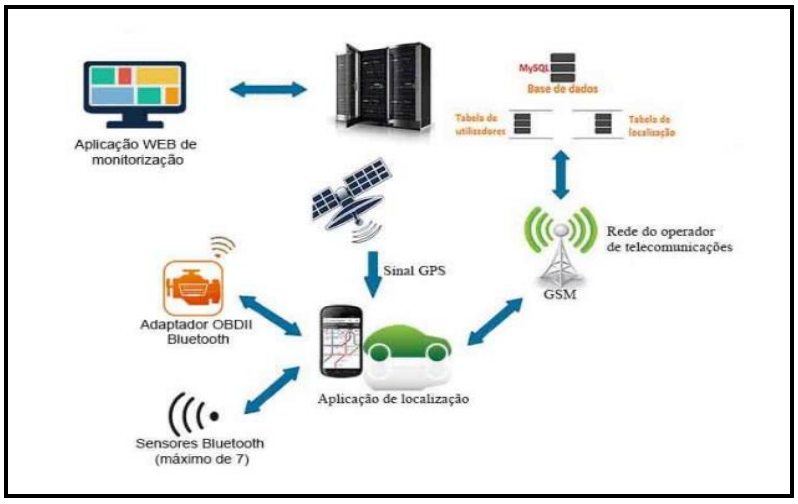
A aplicação de localização permite capturar dados de localização de GPS e estabelecer uma rede *piconet* Bluetooth, admitindo assim uma comunicação com uma unidade de controle do carro através de um adaptador OBDII e com até sete sensores/dispositivos Bluetooth que podem ser instalados no veículo. Os dados adquiridos pela aplicação Android são enviados periodicamente para um servidor *web*.

A aplicação *web* desenvolvida por Pina (2015) permite, ao gestor da frota, efetuar o monitoramento dos veículos em circulação registrados no sistema. É possível visualizar a posição geográfica dos veículos em um mapa, bem como, os dados do mesmo e sensores/dispositivos Bluetooth para cada localização enviada pela aplicação Android.

A Figura 2 exemplifica a arquitetura geral do sistema desenvolvido por Pina (2015), ele idealiza o sistema em quatro principais componentes:

a) aplicação Android de localização;
b) aplicação *web* de monitoramento;
c) adaptadores OBDII/Bluetooth;
d) dispositivos/sensores Bluetooth.

Figura 2 – Arquitetura geral do sistema

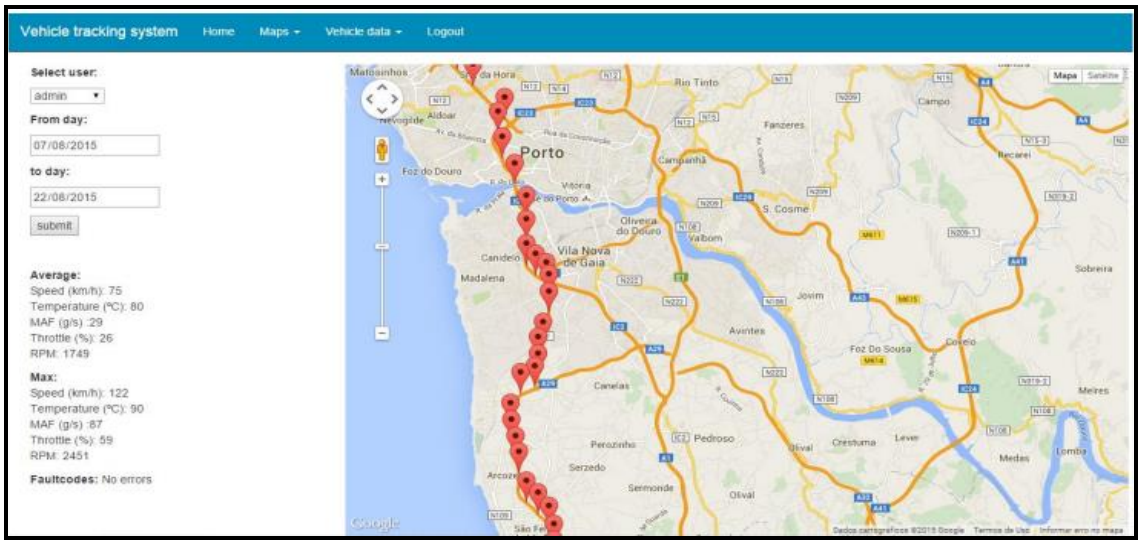


fonte: Pina (2015, p. 79)

A aplicação Android desenvolvida é responsável por gerir a comunicação com o adaptador OBDII/Bluetooth, possíveis sensores Bluetooth adicionais, posição geográfica e transmitir esses dados para o servidor *web*. Para essa aplicação, utilizou-se JAVA com Android e para a persistência de dados no aparelho, utilizou-se o banco de dados SQLite.

Já a aplicação *web* foi desenvolvida para verificar onde está cada veículo geograficamente e as informações capturadas das portas OBDII e sensores Bluetooth. Na Figura 3, pode-se observar a funcionalidade de visualizar as informações geográficas em um mapa. Essa aplicação *web* foi desenvolvida utilizando a tecnologia PHP juntamente com o banco de dados MySQL para persistência.

Figura 3 – Página Vehicle location history



fonte: Pina (2015, p. 115)

Pina (2015) conclui que o sistema desenvolvido é completamente funcional e teve alguma complexidade com o desenvolvimento na plataforma Android, mas que possui uma documentação bem completa. Cita também que teve facilidade em integrar os dispositivos devido à facilidade de instalação e portabilidade. A possibilidade de interação com o adaptador OBDII Bluetooth e com até 7 dispositivos simultaneamente, torna o sistema extremamente versátil, que pode ser utilizada à inúmeras aplicações, até mesmo fora do contexto de veículos como foi apresentado.

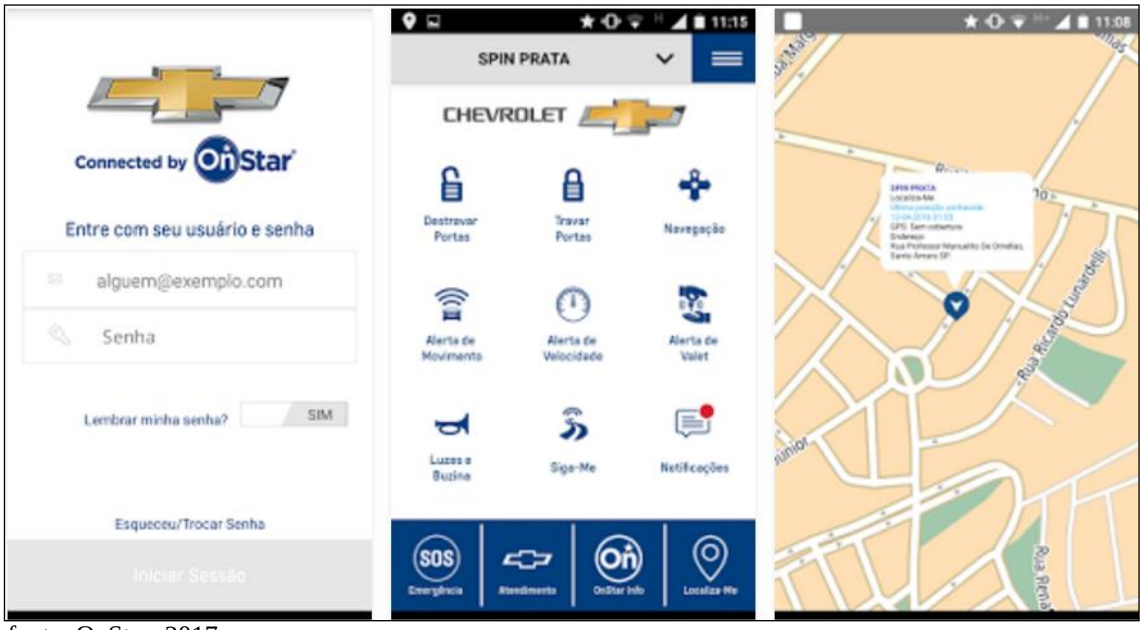
Pina (2015) diz que uma das desvantagens foi a fragilidade do adaptador OBDII/Bluetooth, sugere então, que seja utilizado outro de maior qualidade. O sistema pode aumentar a quantidade de funcionalidades disponíveis, tais como: a leitura de mais parâmetros da porta OBDII, melhoras no layout da aplicação e também um sistema de mensagens entre gestor e utilizador da aplicação.

2.3 ONSTAR

Onstar é um aplicativo exclusivo para clientes Chevrolet. Conforme Figura 4, o aplicativo mantem-se conectado remotamente permitindo comandar diversas funcionalidades do veículo à distância através do aplicativo instalado em um smartphone (ONSTAR, 2017). Este aplicativo é uma espécie de “assistente pessoal”, tal como a Siri nos aparelhos Apple. Nele, existem funcionalidades para monitorar o seu veículo à distância, travar ou destravar as portas, reservar restaurantes, marcar reuniões e até saber horóscopo (Paixão, 2015).

Este recurso é totalmente novo. O próprio OnStar existe a 21 anos nos Estados Unidos mas sem esse serviço de assistente pessoal. Outras marcas como Volvo e BMW oferecem serviços parecidos (Paixão, 2015).

Figura 4 – Aplicativo OnStar Para Android



fonte: OnStar, 2017

O princípio do funcionamento do OnStar é todo baseado no espelho interno do veículo. Existem três botões: um serve para atender ligações da central, outro serve para solicitar serviços de assistência e outro é para emergência. De acordo com a GM, há um chip instalado na base do espelho, porém, não há custos extras com ligações para os usuários. Além do espelho, o motorista pode acessar as funcionalidades por um aplicativo (Figura 4) ou através de um website (Paixão, 2015).

Paixão (2015) destaca os pontos fortes do aplicativo, são eles:

a) a solução é a mais completa que a dos concorrentes;
b) chega com modelos generalistas e bem mais baratos do que qualquer modelo BMW ou Volvo.

3 FERRAMENTAS ATUAIS

Nesta seção serão apresentados dois trabalhos, ambos desenvolvidos pelo curso de Ciência da Computação na Universidade Regional de Blumenau. A seção 3.1 trata de um trabalho denominado Findcar desenvolvido por Baumgarten (2016). Por fim, a seção 3.2 apresenta o trabalho OBD-JRP, desenvolvido por Starosky (2016).

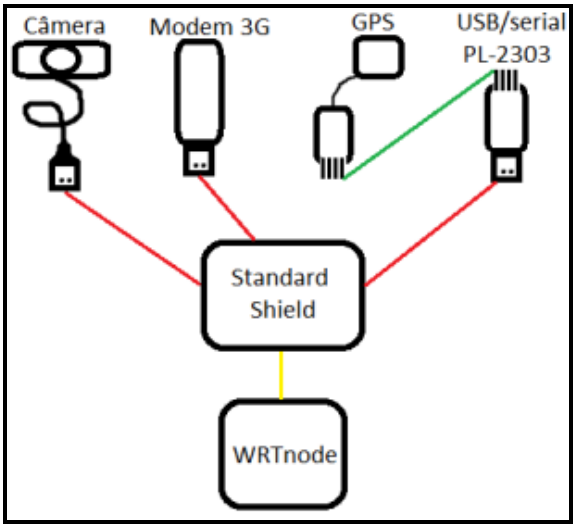
3.1 FINDCAR

Baumgarten (2016) desenvolveu um dispositivo que possibilitasse o rastreamento veicular através de geolocalização e uma imagem capturada através de uma câmera acoplada neste dispositivo. Os objetivos cumpridos no trabalho foram:

a) realizou-se a integração do OpenWRT com: um modem 3rd Generation (3G), um módulo Global Positioning System (GPS) e uma câmera;
b) desenvolveu uma plataforma- *web* para verificar a localização atual, ultimas localizações do veículo, capturar imagens e configurar o envio de notificações por e-mail;
c) tornou o rastreador o próprio servidor onde a aplicação é executada.

Para este rastreador veicular, Baumgarten (2016) utilizou uma placa WRTnode de modelo MT7620 com OpenWRT que é uma distribuição customizável do Linux para sistemas embarcados. Utilizou também, um módulo de GPS Ublox GY-NEO6MV2 para capturar a geolocalização e uma webcam Logitech C270 para capturar as imagens do veículo. Todas as informações são disponibilizadas através de um modem 3G/4G Huawei E3272. O esquema de conexões pode ser observado na Figura 5.

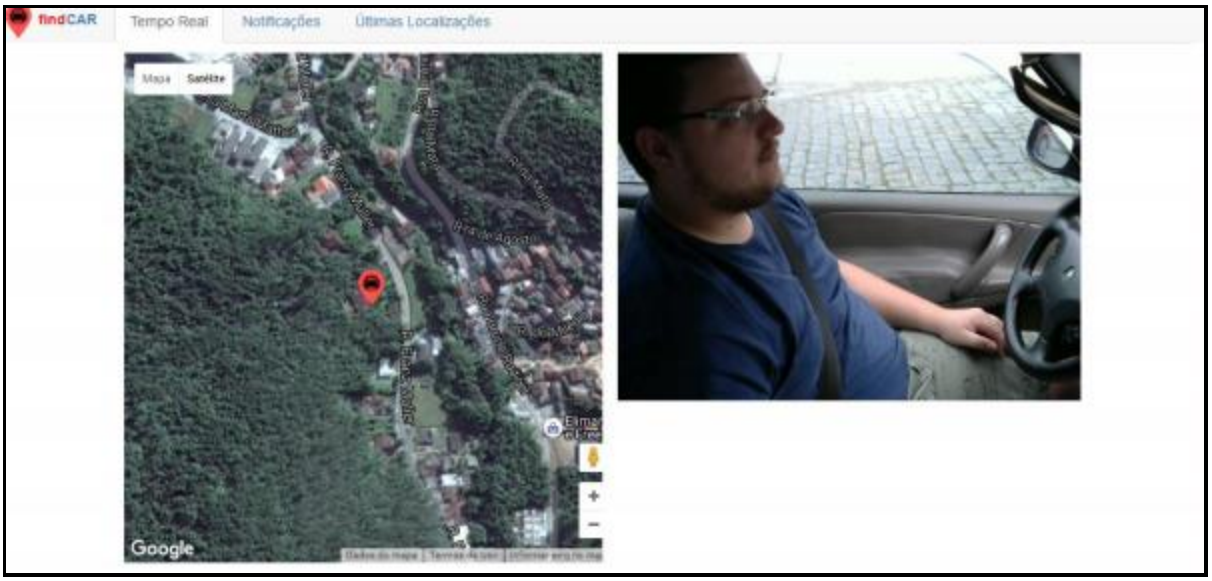
Figura 5 – Diagrama esquemático de conexões



fonte: Baumgarten (2016, p. 26)

Para desenvolver a aplicação *web*, Baumgarten (2016) utilizou para interface gráfica HTML5, CSS3 e Bootstrap3. Com relação à comunicação com o servidor, foi utilizado PHP e a persistência de dados foi feita através do banco de dados MySQL. Utilizou também Google Maps Javascript Api para mostrar as coordenadas capturadas pelo GPS no mapa, conforme Figura 6.

Figura 6 – Tela de captura em tempo real



fonte: Baumgarten (2016, p. 39)

Baumgarten (2016) ressalta que o objetivo do trabalho foi atingido adequadamente. O uso da linguagem PHP supriu as necessidades do sistema. Ele enfatiza o uso do banco de dados MySQL que foi facilmente integrado e manipulado com o OpenWRT.

3.2 OBD-JRP

Staroski (2016) desenvolveu um protótipo de software embarcado em uma placa Raspberry Pi. Esta placa foi utilizada para capturar dados da porta OBD de um veículos e disponibilizá-los em uma página *web*. Ele enumerou e concluiu alguns objetivos específicos que foram atendidos, são eles:

a) desenvolver o firmware, que irá monitorar a porta OBD2 do carro, coletar dados e os enviar para um servidor;
b) desenvolver o software servidor, que irá receber os dados coletados pelo firmware e armazenar os mesmos;
c) desenvolver uma página *web* para consultar o histórico dos dados.

Para a elaboração do trabalho, Starosky (2016) utilizou o ambiente de desenvolvimento Java com a biblioteca BlueCove para realizar a comunicação com a

interface ELM327 Bluetooth. No desenvolvimento do servidor, foi utilizado a biblioteca Google Charts para criar gráficos com a linguagem Javascript. Foi utilizado a placa Raspberry Pi 3 Model B com o sistema operacional Raspian GNU/Linux 8 que é disponibilizada com a versão 1.8 do Java. Os dispositivos citados e utilizados podem ser visualizados na Figura 7. Além desses dispositivos, para concluir a comunicação com o servidor, foi utilizado um modem 3G/4G da marca Huawei.

Figura 7 - Instalação no Volkswagen SpaceFox 2009

fonte: Starosky (2016, p. 73)

A aplicação foi testada em três veículos, sendo eles: GM Corsa Sedan 2003, Volkswagen Gol 2010 e um Volkswagen SpaceFox 2009. Todos possuíam o conector OBD2, porém o Corsa Sedan 2003 não implementava nenhum protocolo OBD2 apesar de possuir a porta. O protótipo atendeu os objetivos propostos e o Raspberry Pi atendeu as exigências computacionais desenvolvidas.

4 PROPOSTA DA APLICAÇÃO

Este capítulo tem como objetivo apresentar a justificativa para elaboração deste trabalho, assim como os requisitos e metodologia de desenvolvimento.

4.1 JUSTIFICATIVA

No Quadro 1, as principais diferenças entre os trabalhos correlatos é apresentada de forma comparativa. Observa-se primeiramente, que nenhum trabalho apresentado desenvolveu uma notificação quando ocorrer alguma falha no veículo. Também, Pacheco (2016) implementou uma ferramenta que não é acessível através de dispositivos móveis.

Quadro 1 – Comparativo entre os trabalhos correlatos

Correlatos / Características	Pacheco (2016)	Pina (2015)	OnStar (2017)
Hardware do sistema embarcado	Raspberry Pi 3	*Smartphone* Android	Próprio
Notificação de falhas no veículo	Não	Não	Não
Leitura da porta OBD	Sim	Sim	Não informado
Utilização em dispositivos móveis	Não	Sim	Sim
Linguagens	Python	PHP, Java e Android	Não informada
Marcas de carro suportadas	Todas	Todas	Somente Chevrolet

fonte: elaborado pelo autor.

O instituto de pesquisa Gartner divulgou um *ranking* de sistemas operacionais móveis mais utilizados em 2016, quem lidera esse *ranking* é o sistema operacional Android (82,2%) e o IOS (12,9%) (PAVÃO, 2016). Segundo o Quadro 1, Pina (2016) e OnStar (2017) construíram as aplicações que são também acessáveis por dispositivos móveis. Porém, Pina (2015) implementou o recurso como um software embarcado no aparelho Android, sendo assim, longe do veículo a aplicação no dispositivo móvel ficaria inoperante, além disso, ela funciona somente com aparelhos que tenham o sistema operacional Android. Já a aplicação OnStar (2017), funciona com dispositivos móveis, porém, opera somente com carros da marca Chevrolet. O trabalho implementará a funcionalidade para dispositivos móveis com o sistema operacional Android e IOS independentes do software embarcado e também funcionará para todas as marcas de automóveis que disponibilizarem portas OBD e implementam os protocolos suportados.

Foi levado em consideração o baixo custo da placa para o desenvolvimento deste projeto. Pacheco (2016) fez uso da placa Raspberry Pi 3 que custa 35 dólares americanos, ele sugere como melhoria, a utilização da Raspberry Pi Zero que custa 5 dólares americanos. Para este trabalho, será necessário a integração da Raspberry Pi com a interface OBD *bluetooth*, porém, a Raspberry Pi Zero não possui *bluetooth* integrada. Por isso, será utilizada a placa Raspberry Pi Zero W que já disponibiliza *bluetooth* e *wireless* integrado conforme Social Compare (2017). Segundo Social Compare (2017), esta placa Raspberry Pi Zero W custa 10 dólares americanos, 25 dólares a menos do que a Raspberry Pi 3. Levando em consideração o consumo de energia, a placa Raspberry Pi Zero W consome 180 mA, enquanto que a Raspberry Pi 3 consome 800 mA (SOCIAL COMPARE, 2017).

No meio social, a aplicação pode auxiliar na prevenção aos danos causados pela emissão de gases, resultado na melhor qualidade do ar (RESOLUÇÃO CONAMA Nº 354,

2004). Além, de auxiliar e alertar os condutores de veículos sobre falhas em seus automóveis. Na área profissional o trabalho é relevante por propor aspectos da especificação OBD2, bem como por utilizar GPS que podem servir de base para possíveis soluções comerciais.

4.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A aplicação descrita neste trabalho deverá:

a) disponibilizar informações da geolocalização do veículo (Requisito Funcional - RF);
b) capturar uma foto do veículo através de uma *webcam* (RF);
c) disponibilizar dados dos sensores do automóvel através da porta OBD (RF);
d) permitir receber notificação caso ocorram falhas na porta OBD do veículo (RF);
e) utilizar a placa Raspberry PI Zero W (Requisito Não Funcional - RNF);
f) integrar placa Raspberry PI Zero W com a porta OBD do veículo via *bluetooth* (RNF);
g) integrar a placa Raspberry PI Zero W com um módulo GPS (RNF);
h) utilizar a linguagem de programação JAVA (RNF);
i) utilizar o banco de dados PostgreSQL para persistir dados (RNF);
j) utilizar a biblioteca Ionic Framework para desenvolver a aplicação *mobile* (RNF).

4.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

a) levantamento bibliográfico: fazer levantamentos bibliográficos sobre OBD2, Raspberry Pi e Ionic Framework.
b) elicitação de requisitos: baseando-se no levantamento bibliográfico, reavaliar os requisitos e, se caso necessário, elaborar mais requisitos;
c) especificação: utilizar a ferramenta Enterprise Architect (EA) para elaborar os diagramas de casos de uso, diagramas de classes, diagrama de pacotes e fluxograma para explicitar a integração entre os sistemas mobile e embarcado;
d) implementação: à partir da especificação, desenvolver o software embarcado utilizando a linguagem de programação JAVA, desenvolver o sistema *mobile* com a Ionic Framework e persistir os dados utilizando PostgreSQL;
e) testes: paralelamente à implementação, realizar testes de comunicação entre os hardwares da placa embarcada Raspberry PI Zero W, o módulo GPS, o módulo *bluetooth* OBD, realizar testes de integração entre os sistemas embarcados e *mobile*

para verificar se a comunicação se dá de forma adequada e testar o envio de *sms* e e-mail.

As etapas serão realizadas nos períodos relacionados no Quadro 2.

Quadro 2 - Cronograma

	2017									
	ago.		set.		out.		nov.		dez.	
etapas / quinzenas	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico										
elicitação dos requisitos										
especificação										
implementação										
testes										

Fonte: elaborado pelo autor.

5 REVISÃO BIBLIOGRÁFICA

Este capítulo descreve brevemente os assuntos que fundamentarão o estudo a ser realizado: OBD2, Raspberry Pi e Ionic Framework.

Segundo Santos (2016) OBD é uma sigla do inglês *On-Board Diagnostic* e designa um sistema de autodiagnostico disponível na maioria dos veículos e a ligação ao sistema ocorre por meio de um conector padronizado que foi sancionado como obrigatório na Europa e nos Estados Unidos à partir de 1996. No Brasil foi sancionado como obrigatório somente à partir de 2010 com o padrão de segunda geração OBD2. Santos (2016) cita que “A medida tem a finalidade de fiscalizar a emissão de gases poluentes na atmosfera, dado que, alguns países possuem acordos mundiais em que se comprometem com a preservação ambiental, como o protocolo de Kyoto.”.

Raspberry Pi é um computador do tamanho de um cartão de crédito que se conecta a um monitor ou uma TV, usa um teclado e mouse padrão, ele foi desenvolvido no Reino Unido pela Fundação Raspberry Pi. Todo o hardware é integrado à uma única placa. O principal objetivo é promover o ensino da ciência da computação básica em escolas (RASPBERRY PI FOUNDATION, 2017, tradução nossa).

Ionic é uma biblioteca *free* e *open source* para criar aplicativos híbridos com HTML5, CSS e Javascript para as versões iOS 6 ou superiores e Android 4.0 ou superiores. O Ionic foi criado com base no AngularJS e possui vários componentes e ferramentas que facilitam o desenvolvimento e não prejudicam a performance do seu aplicativo (FRANCO, 2016).

REFERÊNCIAS

BAUMGARTEN, Nykolas E. A., **FINDCAR**: RASTREADOR VEICULAR UTILIZANDO OPENWRT. 2016, 56 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

CZERWONKA, Mariana, **Falta de manutenção triplica risco de acidentes**. [Goiás?], 2016. Disponível em <http://portaldotransito.com.br/noticias/falta-de-manutencao-triplica-risco-de-acidentes>. Acesso em: 20 mar 2017.

FRANCO, Felipe. **Como criar aplicativos com Ionic Framework**. 2016. Disponível em: <http://www.fabricadecodigo.com/como-criar-aplicativos-com-ionic-framework/>. Acesso em: 01 abr. 2017.

G1 Ribeirão e Franca, **Estatística divulgada pela SSP mostra aumento da violência em Ribeirão**, [São Paulo], 2017. Disponível em: <http://g1.globo.com/sp/ribeirao-preto-franca/noticia/2017/02/estatistica-divulgada-pela-ssp-mostra-aumento-da-violencia-em-ribeirao.html>. Acesso em: 20 mar 2017.

ODEGA, Alessandra, **Confira os modelos de veículos mais roubados e quanto custa o seguro de cada um deles**. Santa Catarina, 2016. Disponível em: <http://ndonline.com.br/florianopolis/coluna/alessandra-ogeda/confira-os-modelos-de-veiculos-mais-roubados-e-quanto-custa-o-seguro-de-cada-um-deles>. Acesso em: 20 mar 2017.

ONSTAR, **OnStar Br**. [São Paulo?], 2017. Disponível em <https://play.google.com/store/apps/details?id=com.roadtrack.onstar&hl=pt_BR>. Acesso em: 31 mar 2017.

PACHECO, Lucas V., **Monitoramento e gestão de uma frota de veículos utilizando sistemas embarcados**. 2016, 76 f. Trabalho de Conclusão de Curso (Curso de Engenharia Elétrica) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Paulo.

PAIXÃO, André, **Veja como funciona o OnStar, 'assistente pessoal' da Chevrolet**. São Paulo, 2016. Disponível em: < http://g1.globo.com/carros/noticia/2015/10/veja-como-funciona-o-onstar-assistente-pessoal-da-chevrolet.html>. Acesso em: 31 mai 2017.

PAVÃO, Felipe. **Os 5 sistemas operacionais mobile mais vendidos de 2016**. 2016. Disponível em: <https://www.tecmundo.com.br/mercado/108748-5-sistemas-operacionais-mobile-vendidos-2016.htm>. Acesso em: 31 mar. 2017.

PINA, Afonso L. P., **SISTEMA DE LOCALIZAÇÃO DE VEÍCULOS PARA SMARTPHONE ANDROID**. 2015, 136p, Tese/Dissertação de Mestrado (Engenharia Eletrotécnica e de Computadores) – Departamento de Engenharia Eletrotécnica, Instituto Superior de Engenharia do Porto, Porto - Portugal.

RASPBERRY PI FOUNDATION, **FAQS.** [2017?]. Disponível em: <https://www.raspberrypi.org/help/faqs/>. Acesso em: 01 abr. 2017.

RESOLUÇÃO CONAMA nº 354, de 13 de dezembro de 2004. Publicada no D.O.U. nº 239, de 14 de dezembro de 2004, Seção 1, p. 62-63. Disponível em: <http://www.mma.gov.br/port/conama/legislacao/CONAMA_RES_CONS_2004_354.pdf>. Acesso em: 01 abr. 2017.

SANTOS, Arthur Luis V., **Economia de combustível com o uso de telemetria para veículos de passeio**. 2016, 82 f. Trabalho de Conclusão de Curso (Engenharia de Computação) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Rio Grande do Sul.

SOCIAL COMPARE. **RaspberryPI models comparison**. 2017. Disponível em: <http://socialcompare.com/en/comparison/raspberrypi-models-comparison>. Acesso em: 01 abr. 2017

SOUZA, Beatriz, **Quatro em cada dez veículos de carga apresentam falha mecânica**. [Santa Catarina?], 2016. Disponível em: <http://www.perkons.com.br/noticia/1694/quatro-em-cada-dez-veiculos-de-carga-apresentam-falha-mecanica>. Acesso em: 20 mar 2017.

STAROSKY, Ricardo A., **OBD-JRP**: monitoramento veicular com java e raspberry pi. 2016, 87 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

ASSINATURAS

(Atenção: todas as folhas devem estar rubricadas)

Assinatura do(a) Aluno(a): _____

Assinatura do(a) Orientador(a): _____

Assinatura do(a) Coorientador(a) (se houver): _____

Observações do orientador em relação a itens não atendidos do pré-projeto (se houver):

FORMULÁRIO DE AVALIAÇÃO – PROFESSOR TCC I

Acadêmico(a): _____

Avaliador(a): _____

	ASPECTOS AVALIADOS[1]	atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?			
	O problema está claramente formulado?			
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?			
	Os objetivos específicos são coerentes com o objetivo principal?			
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?			
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?			
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?			
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?			
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?			
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?			
	7. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?			
	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?			
ASPECTOS METODOLÓGICOS	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?			
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?			
	9. ORGANIZAÇÃO E APRESENTAÇÃO GRÁFICA DO TEXTO A organização e apresentação dos capítulos, seções, subseções e parágrafos estão de acordo com o modelo estabelecido?			
	10. ILUSTRAÇÕES (figuras, quadros, tabelas) As ilustrações são legíveis e obedecem às normas da ABNT?			
	11. REFERÊNCIAS E CITAÇÕES As referências obedecem às normas da ABNT?			
	As citações obedecem às normas da ABNT?			
	Todos os documentos citados foram referenciados e vice-versa, isto é, as citações e referências são consistentes?			

PARECER – PROFESSOR DE TCC I OU COORDENADOR DE TCC
(PREENCHER APENAS NO PROJETO):

O projeto de TCC será reprovado se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos **4 (quatro)** itens dos **ASPECTOS TÉCNICOS** tiverem resposta ATENDE PARCIALMENTE; ou
- pelo menos **4 (quatro)** itens dos **ASPECTOS METODOLÓGICOS** tiverem resposta ATENDE PARCIALMENTE.

PARECER: ( ) APROVADO ( ) REPROVADO

Assinatura: _____ Data: _____

[1] Quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.

FORMULÁRIO DE AVALIAÇÃO – PROFESSOR AVALIADOR

Acadêmico(a): _____

Avaliador(a): _____

	ASPECTOS AVALIADOS[1]	atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?			
	O problema está claramente formulado?			
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?			
	Os objetivos específicos são coerentes com o objetivo principal?			
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?			
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?			
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?			
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?			
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?			
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?			
	7. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?			
	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?			
ASPECTOS METODOLÓGICOS	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?			
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?			

PARECER – PROFESSOR AVALIADOR:
(PREENCHER APENAS NO PROJETO)

O projeto de TCC ser deverá ser revisado, isto é, necessita de complementação, se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos **5 (cinco)** tiverem resposta ATENDE PARCIALMENTE.

PARECER: ( ) APROVADO ( ) REPROVADO

Assinatura: _____ Data: _____

[1] Quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.