

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO**

**FINDCAR: RASTREADOR VEICULAR UTILIZANDO**  
**OPENWRT**

**NYKOLAS EDUARDO ANTONIOLI BAUMGARTEN**

**BLUMENAU**  
**2016**

**NYKOLAS EDUARDO ANTONIOLI BAUMGARTEN**

**FINDCAR: RASTREADOR VEICULAR UTILIZANDO**

**OPENWRT**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Prof. Miguel Alexandre Wisintainer, Mestre - Orientador

**BLUMENAU  
2016**

# **FINDCAR: RASTREADOR VEICULAR UTILIZANDO OPENWRT**

Por

**NYKOLAS EDUARDO ANTONIOLI BAUMGARTEN**

Trabalho de Conclusão de Curso aprovado  
para obtenção dos créditos na disciplina de  
Trabalho de Conclusão de Curso II pela banca  
examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Miguel Alexandre Wisintainer, Mestre – Orientador, FURB

Membro: \_\_\_\_\_  
Profa. Luciana Pereira de Araújo, Mestre – FURB

Membro: \_\_\_\_\_  
Prof. Mauro Marcelo Mattos, Doutor – FURB

Blumenau, 06 de dezembro de 2016

Dedico este trabalho a minha família e a minha noiva que sempre me apoiaram e incentivaram para a conclusão do mesmo.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus por me capacitar e permitir que eu completasse mais esta etapa da minha vida.

À minha família pai Cláudio Eduardo Baumgarten, mãe Maralucia Antonioli Baumgarten e irmão João Victor Antonioli Baumgarten por compreenderem e me apoiarem sempre que precisei.

À minha noiva Rejane Luiza Lingner por me apoiar e sempre me animar para continuar o desenvolvimento do projeto.

Ao meu orientador Miguel Alexandre Wisintainer por me auxiliar no desenvolvimento do projeto, tirando dúvidas, orientando e sempre disposto a ajudar de alguma forma.

Aos demais amigos e familiares que apoiaram de alguma forma e compreenderam este período de afastamento para que o projeto pudesse ser concluído.

“Consagre ao Senhor tudo o que você faz, e os seus planos serão bem-sucedidos”.

Provérbios 16:3

## RESUMO

Este trabalho apresenta o desenvolvimento de um rastreador veicular independente, ou seja, sem demandar de um servidor externo. Para possibilitar esta autonomia em relação a servidor web, o rastreador utiliza o Weaved. Este rastreador foi desenvolvido utilizando uma distribuição embarcada de Linux chamada OpenWRT. Para a captura da localização foi utilizado um módulo de GPS e para a emissão das notificações e a conexão com o sistema FINDCAR foi utilizado um modem 3G Huawei. Além de fazer o rastreamento do veículo, este sistema permite que o usuário capture imagens do interior do veículo através de uma webcam integrada ao hardware. O sistema permite que o usuário realize a configuração de notificações que são enviadas via e-mail para destinatários em intervalos de tempo pré-definidos. Ao concluir o desenvolvimento do sistema, foram realizados alguns testes de campo para verificar a funcionalidade do rastreador. Uma das dificuldades encontradas foi o tempo de retorno das requisições realizadas no sistema, que aparentemente são causados pela oscilação da taxa de upload na rede 3G. Em questão de rastreamento e captura de imagens tanto a precisão do local quanto a qualidade das imagens capturadas pela câmera tiveram um bom resultado.

Palavras-chave: Rastreador veicular. Openwrt. Wrtnode. Sistema GPS.

## **ABSTRACT**

This work presents the development of an independent vehicular tracker, in other words, without the need of external server. To allow this autonomy in relation to the web server, the tracker uses Weaved. This tracker was developed using an embedded Linux distribution called OpenWRT. In order to capture the location, a GPS module was used and a Huawei 3G modem was used to issue the notifications and the connection to the FINDCAR system. In addition to tracking the vehicle, this system allows the user to capture images from the interior of the vehicle through a webcam integrated with the hardware. The system allows the user to configure the notifications that are sent by e-mail to recipients in predefined time intervals. At the conclusion of the development of the system, some field tests were performed to verify the functionality of the tracker. One of the difficulties was the time of return of the requests made in the system, which are apparently caused by the oscillation of the upload rate in the 3G network. In terms of tracking and capturing images, both the accuracy of the site and the quality of the images captured by the camera had a good result.

Key-words: Vehicle tracker. Openwrt. Wrtnode. GPS system.



## LISTA DE FIGURAS

Figura 1 – Tela inicial do OpenWRT .....	14
Figura 2 – Interface LUCI do OpenWRT.....	15
Figura 3– Diagrama Weaved.....	16
Figura 4 – Página de acesso do Weaved.....	17
Figura 5 – Tela de consulta.....	18
Figura 6 – Telas de consulta TCC Santos Júnior.....	19
Figura 7 – Tela de consulta Bustracker .....	20
Figura 8 – Tela de consulta no mapa Bustracker.....	20
Figura 9 – Diagrama de casos de uso .....	23
Figura 10 – Diagrama de atividades para o login.....	24
Figura 11 – Diagrama de atividades para o envio de notificações .....	25
Figura 12 – Diagrama esquemático de conexões .....	26
Figura 13 – WRTnode lado superior e inferior .....	27
Figura 14 – Standard Shield para WRTnode lado superior e inferior .....	28
Figura 15 – Módulo de GPS Ublox GY-NEO6MV2 .....	29
Figura 16 – Modem 3G/4G Huawei E3272.....	29
Figura 17 – Adaptadores USB/Serial PL-2303 e CH340G.....	30
Figura 18 – Webcam Logitech C270.....	31
Figura 19 – Menu para compilação dos pacotes.....	32
Figura 20 – Tela de login.....	38
Figura 21 – Tela de cadastro de usuário .....	39
Figura 22 –Tela de captura em tempo real .....	39
Figura 23 – Tela de consulta de notificações .....	40
Figura 24 – Tela de cadastro de notificações .....	41
Figura 25 – Tela de alteração de notificações .....	41
Figura 26 – Tela de consulta de últimas localizações .....	42
Figura 27 – Alertas do sistema ao usuário.....	43

## LISTA DE QUADROS

Quadro 1 – Exemplo de captura de imagem.....	33
Quadro 2 – Fonte para escutar porta serial .....	34
Quadro 3 – Captura de localização .....	35
Quadro 4 – Chamada da API Maps JavaScript .....	36
Quadro 5 – Script proc_notif.php .....	37
Quadro 6 – Comparação com correlatos .....	45
Quadro 7 – Pacotes para auxílio ao desenvolvimento.....	50
Quadro 8 – Pacotes para periféricos .....	50
Quadro 9 – Pacotes para base de dados e servidor web .....	51
Quadro 10 – Demais pacotes instalados .....	52
Quadro 11 – Custo do projeto .....	55

## **LISTA DE ABREVIATURAS E SIGLAS**

3G – 3rd Generation

API – Application Programming Interface

GPS – Global Positioning System

iOS – iPhone Operating System

Mac OS – Macintosh Operating System

Mbit – Mega-bit

Mbps – Mega-bits por segundo

RF – Requisito Funcional

RNF – Requisito não Funcional

SFTP – SSH File Transfer Protocol

TCP/IP – Transmission Control Protocol/Internet Protocol

USB – Universal Serial Bus

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>12</b>
1.1 OBJETIVOS.....	12
1.2 ESTRUTURA.....	13
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>14</b>
2.1 OPENWRT.....	14
2.2 WEAVED (REMOT3.IT) .....	15
2.3 TRABALHOS CORRELATOS .....	17
2.3.1 PROTÓTIPO DE UM SISTEMA DE RASTREAMENTO VEICULAR BASEADO NO MÓDULO TELIT .....	17
2.3.2 SISTEMA DE MONITORAMENTO ELETRÔNICO AUTOMOTIVO .....	18
2.3.3 BUSTRACKER: SISTEMA DE RASTREAMENTO PARA TRANSPORTE COLETIVO.....	19
<b>3 DESENVOLVIMENTO.....</b>	<b>21</b>
3.1 REQUISITOS.....	21
3.2 ESPECIFICAÇÃO .....	22
3.2.1 Diagrama de casos de uso .....	22
3.2.2 Diagramas de atividades .....	23
3.3 IMPLEMENTAÇÃO .....	26
3.3.1 Técnicas e ferramentas utilizadas.....	26
3.3.2 Hardware e periféricos do rastreador .....	27
3.3.3 Compilação de módulos e pacotes .....	31
3.3.4 Captura de imagem .....	32
3.3.5 Captura da localização .....	33
3.3.6 Google Maps JavaScript API.....	35
3.3.7 Processo de execução das notificações .....	36
3.3.8 Desenvolvimento <i>front end</i> .....	37
3.3.9 Operacionalidade da implementação .....	38
3.4 ANÁLISE DOS RESULTADOS .....	44
<b>4 CONCLUSÕES.....</b>	<b>46</b>
4.1 EXTENSÕES .....	46

## 1 INTRODUÇÃO

Há algum tempo a criminalidade ocupa uma grande parte dos noticiários. Esse problema vem ocorrendo não só no nosso país, mas também vem atingindo diversas outras nações. Vários fatores cooperam para um aumento da taxa de criminalidade, como por exemplo a miséria, a desigualdade social, a falta de amparo do governo, a cultura as más companhias entre outros (GARRIDO, 2007).

Em Santa Catarina tem-se alguns índices de criminalidade que se mantem em destaque com o passar dos anos, como por exemplo homicídios, furtos e roubos de estabelecimentos comerciais e veículos. Os roubos e furtos de veículos tiveram um aumento de 16,5% entre 2013 e 2014 no estado, sendo que, somente neste período foram registrados 7833 veículos furtados. As cidades que lideram o ranking deste tipo de delito são Itajaí, Florianópolis, Balneário Camboriú, Criciúma, São José, Chapecó, Blumenau, Brusque e Palhoça (DIÁRIO DO VALE, 2015).

Segundo pesquisas realizadas no estado, algumas cidades possuem taxas de recuperação de veículos entre 49% e 66%, como em Joinville situada no norte de Santa Catarina (BATISTA, 2015). O resultado dessa pesquisa expõe a dificuldade de recuperação de veículos furtados no estado, e o quão alarmante é este índice para os proprietários, que por sua vez, podem tentar aumentar as suas chances de recuperação com uso de rastreadores veiculares.

Tendo em vista a situação apresentada, a proposta deste trabalho é desenvolver um dispositivo que possibilite aos proprietários localizarem seus veículos a qualquer momento e que permita a captura de imagens. Desta forma o proprietário poderá ter mais controle sobre o seu veículo, mesmo não estando próximo. Esta ferramenta pode contribuir com o aumento da taxa de recuperação de veículos furtados, e cooperar com imagens para a identificação de suspeitos.

### 1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver um rastreador veicular.

Os objetivos específicos do trabalho são:

- a) realizar a integração entre OpenWRT e um modem *3rd Generation* (3G), um módulo Global Positioning System (GPS) e uma câmera;
- b) desenvolver um sistema para a plataforma Web, onde será possível verificar a localização atual, últimas localizações do veículo, capturar imagens e configurar envio de notificações por e-mail;

- c) tornar do rastreador o próprio servidor onde a aplicação será executada.

## 1.2 ESTRUTURA

Este trabalho está dividido em 4 capítulos para melhor entendimento e organização do conteúdo. O primeiro capítulo descreve a introdução do trabalho e os objetivos definidos para o mesmo. O segundo capítulo apresenta a fundamentação teórica para uma melhor compreensão sobre OpenWRT e Weaved (Remot3.it). No terceiro capítulo é demonstrado o desenvolvimento do trabalho com requisitos, especificação, implementação, operacionalidade do sistema e resultados obtidos. Por fim no quarto capítulo são descritas as conclusões e possíveis extensões para desenvolvimento e melhoria em trabalhos futuros.

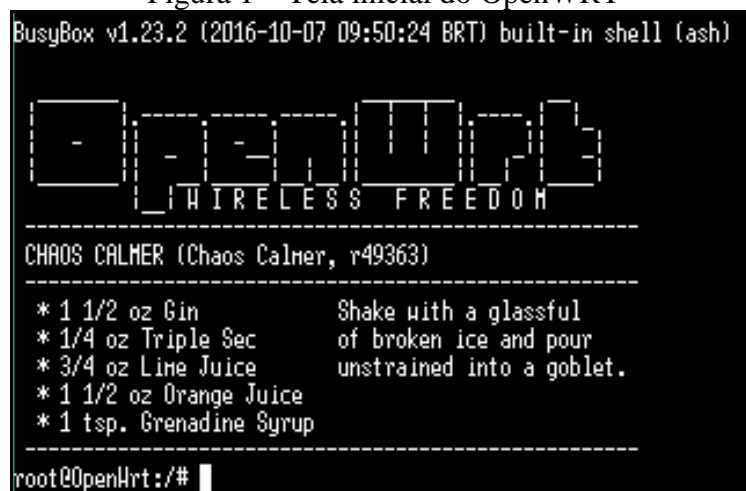
## 2 FUNDAMENTAÇÃO TEÓRICA

A seção 2.1 apresenta características do OpenWRT. A seção 2.2 faz uma explicação sobre o funcionamento da tecnologia GPS. A seção 2.3 discorre sobre a tecnologia 3G e modem 3G onde são expostos alguns dados técnicos e estatísticos. A seção 2.4 oferece um esclarecimento sobre Weaved (Remot3.it). E por fim a seção 2.5 traz um estudo de três trabalhos correlatos.

### 2.1 OPENWRT

OpenWRT é uma distribuição customizável do Linux para sistemas embarcados, utilizado em roteadores. A principal vantagem do OpenWRT é a liberdade de uso desta distribuição (Figura 1). O mesmo não foi desenvolvido com o objetivo de entregar um produto final que supra as necessidades do usuário, mas, disponibilizar um sistema operacional capaz de ser configurado/modificado conforme as mais diversas necessidades existentes (OPENWRT WIRELESS FREEDOM, 2015).

Figura 1 – Tela inicial do OpenWRT



```
BusyBox v1.23.2 (2016-10-07 09:50:24 BRT) built-in shell (ash)
-
WIRELESS FREEDOM
-----
CHAOS CALMER (Chaos Calmer, r49363)
-----
* 1 1/2 oz Gin          Shake with a glassful
* 1/4 oz Triple Sec     of broken ice and pour
* 3/4 oz Lime Juice     unstrained into a goblet.
* 1 1/2 oz Orange Juice
* 1 tsp. Grenadine Syrup
-----
root@OpenWrt:/#
```

Fonte: Elaborado pelo autor.

Esta distribuição pode ser moldada para ser utilizada de diversas formas e cenários. Sua licença de código aberto disponibiliza uma maior abrangência de uso, porém para que possa ser modificado a este nível, é necessário um certo grau de conhecimento da estrutura e funcionamento do Linux (OPENWRT WIRELESS FREEDOM, 2015). Este firmware possui ainda uma interface de configuração muito simples de utilizar desenvolvida em LUA chamada LUCI (Figura 2). Esta interface é desenvolvida em LUA e permite que usuários que não tenham tanto conhecimento de Linux possam utilizar o OpenWRT.

Figura 2 – Interface LUCI do OpenWRT

The screenshot shows the OpenWRT LUCI interface. At the top, there is a navigation bar with 'OpenWrt' on the left and 'Status', 'System', 'Network', and 'Logout' on the right. A green 'AUTO REFRESH ON' button is in the top right corner. The main content area is titled 'Status' and is divided into two sections: 'System' and 'Memory'.

**System**

Hostname	OpenWrt
Model	WRTNODE
Firmware Version	OpenWrt Chaos Calmer 15.05.1 r49363 / LuCI for-15.05 branch (git-16.206.66468-6f5c466)
Kernel Version	3.18.29
Local Time	Sun Dec 11 10:29:41 2016
Uptime	0h 11m 31s
Load Average	0.29, 0.26, 0.23

**Memory**

Total Available	12820 kB / 61536 kB (20%)
Free	8992 kB / 61536 kB (14%)
Buffered	3828 kB / 61536 kB (6%)

Fonte: Elaborado pelo autor.

Nesta interface de configuração do OpenWRT é possível configurar uma conexão com um modem ou roteador. Também é possível realizar a manutenção de módulos e pacotes, instalando e desinstalando conforme a necessidade. Estes pacotes instalados através da interface LUCI já estão compilados e disponíveis em um diretório para instalação. É possível ainda fazer um backup da imagem do firmware instalado no hardware e atualizá-lo através desta interface (OPENWRT WIRELESS FREEDOM, 2015).

## 2.2 WEAVED (REMOT3.IT)

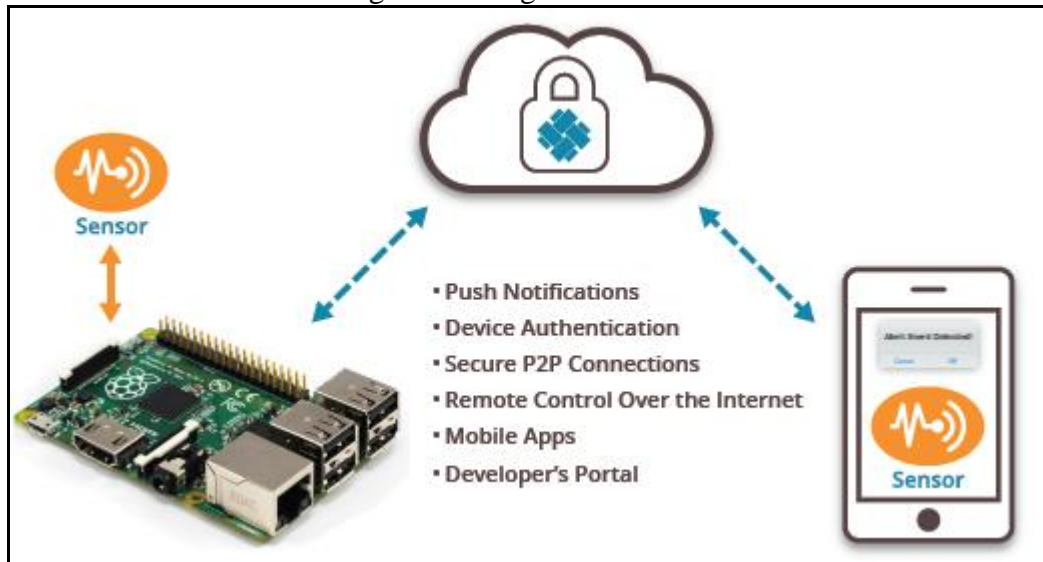
Weaved é um sistema online que torna possível o acesso de diversos dispositivos previamente cadastrados de qualquer lugar do mundo, como se estivessem interligados em uma Local Area Network (LAN). Com esta solução de Virtual Private Network (VPN) é possível acessar qualquer dispositivo que também utilize protocolo Transmission Control Protocol/Internet Protocol (TCP/IP) para se comunicar (WEAVED, 2016).

Este software é compatível com alguns dos sistemas operacionais disponíveis hoje no mercado como Linux, Windows, Macintosh Operating System (Mac OS), Android e iPhone Operating System (iOS). Também é compatível com algumas das principais plataformas embarcadas como o Raspberry Pi, Beagle Bone, Edison e outras demais. Após ser realizada a configuração de um dispositivo no Weaved, ele poderá ser acessado remotamente através de



um aplicativo que pode ser instalado em um dispositivo com Windows, Mac OS, Linux, Android ou iOS, vide Figura 3 (WEAVED, 2016).

Figura 3– Diagrama Weaved



Fonte: Weaved (2016).

O Weaved é um sistema pago, porém é possível utiliza-lo de forma gratuita com tempo limite de sessão de 30 minutos e um serviço concorrente, ou seja, é possível acessar dois dispositivos de forma simultânea (WEAVED, 2016). Na Figura 4 é possível visualizar como é o painel de acesso para o usuário do Weaved, nota-se que os dois dispositivos não estão disponíveis para conexão. Nesta tela é exibida uma lista de dispositivos cadastrados, estes dispositivos podem ser conectados via Hypertext Transfer Protocol (HTTP) ou Secure Shell (SSH) (WEAVED, 2016). No segundo semestre de 2016 Weaved tornou-se Remot3.it, porém ainda é possível acessar dispositivos via Weaved através do link <https://developer.weaved.com/portal/login.php>.

Figura 4 – Página de acesso do Weaved

The screenshot shows the Weaved user interface. At the top left is the Weaved logo. To the right are navigation links: My Services, My Account, Downloads, Forum, and Sign Out. Below the navigation is a welcome message: "Welcome nykbaum".

There are three main sections:

- New Service Plans Available:** A message stating "Go to [My Account](#) to choose a new service plan."
- Your current list of services:** A message stating "Click on service names to connect. Your account allows for 10 registered services and 30 minute connections on up to 1 concurrent service(s). You can [Upgrade Now](#) to get more services, longer connection times and more concurrent connections." Below this is a table of services:

Name	Type	Status	
HTTP_WRTnode	HTTP	offline	<a href="#">Share</a>   <a href="#">Settings</a>
SSH_WRTnode	SSH	offline	<a href="#">Share</a>   <a href="#">Settings</a>

- Adding Services:** A section with instructions: "Most services include setup instructions. If you're looking for our common setup instructions, use the [Downloads](#) page. If you want to pre-register a Docker container or similar service, use our [Pre-Registration](#) page."

Fonte: Weaved (2016).

## 2.3 TRABALHOS CORRELATOS

A seguir são relacionados três trabalhos correlatos. O item 2.5.1 descreve sobre Beszczynski (2008), Protótipo de um sistema de rastreamento veicular baseado no módulo Telit. O item 2.5.2 possui uma descrição sobre Santos Júnior (2009), Sistema de monitoramento eletrônico automotivo, onde ele faz uso de OBDII. E por fim o item 2.5.3 faz menção sobre Vicenzi (2015), Bustracker: sistema de rastreamento para transporte coletivo.

### 2.3.1 PROTÓTIPO DE UM SISTEMA DE RASTREAMENTO VEICULAR BASEADO NO MÓDULO TELIT

Beszczynski (2008) tem como objetivo disponibilizar uma forma de rastrear um veículo e capturar imagens do interior do mesmo. O protótipo desenvolvido faz uso de GPS para recuperação da localização do veículo, possui uma câmera para captura de imagens e outros componentes que compunham o hardware. Para a parte de software foram utilizados PHP, Python e Delphi.

Esta aplicação faz a captura da localização e de uma imagem do interior do veículo, ambas as informações são armazenadas em um banco de dados em um servidor, para que possam ser utilizadas de alguma forma posteriormente. A aplicação desenvolvida por Beszczynski (2008) também permite que o automóvel seja bloqueado ou desbloqueado a distância via página Web. Como pode ser visto na Figura 5, na interface criada para consulta

das tomadas de localização realizadas pelo rastreador além de registrar a localização o sistema exibe uma imagem capturada a partir do interior do veículo e a velocidade no momento da captura.

Figura 5 – Tela de consulta

Data/Hora	Latitude	Longitude	Velocidade	Mapa	Foto
Jul 07 2008 09:23AM	-26°54.2804'	-049°04.8976'	0.36Km/h		
Jul 06 2008 03:52PM	-26°50.4626'	-048°37.8631'	38.26Km/h		
Jul 06 2008 03:30PM	-26°49.7529'	-048°37.4788'	32.22Km/h		
Jul 06 2008 03:26PM	-26°49.9709'	-048°37.6438'	37.98Km/h		
Jul 06 2008 03:15PM	-26°53.0389'	-048°38.5445'	36.68Km/h		
Jul 06 2008 03:12PM	-26°53.7768'	-048°39.1455'	34.70Km/h		
Jul 06 2008 02:54PM	-26°52.1453'	-048°38.3233'	46.69Km/h		

Fonte: Beszczynski (2008).

### 2.3.2 SISTEMA DE MONITORAMENTO ELETRÔNICO AUTOMOTIVO

Santos Júnior (2009) tem como objetivo permitir que o usuário possa acessar os dados referentes ao seu veículo através de um dispositivo móvel com Windows Mobile 6 como pode ser visto na Figura 6. Nesta aplicação foi utilizada a linguagem C#. Com esta linguagem foram desenvolvidas as rotinas de integração e o software que é executado no dispositivo móvel.

Figura 6 – Telas de consulta TCC Santos Júnior



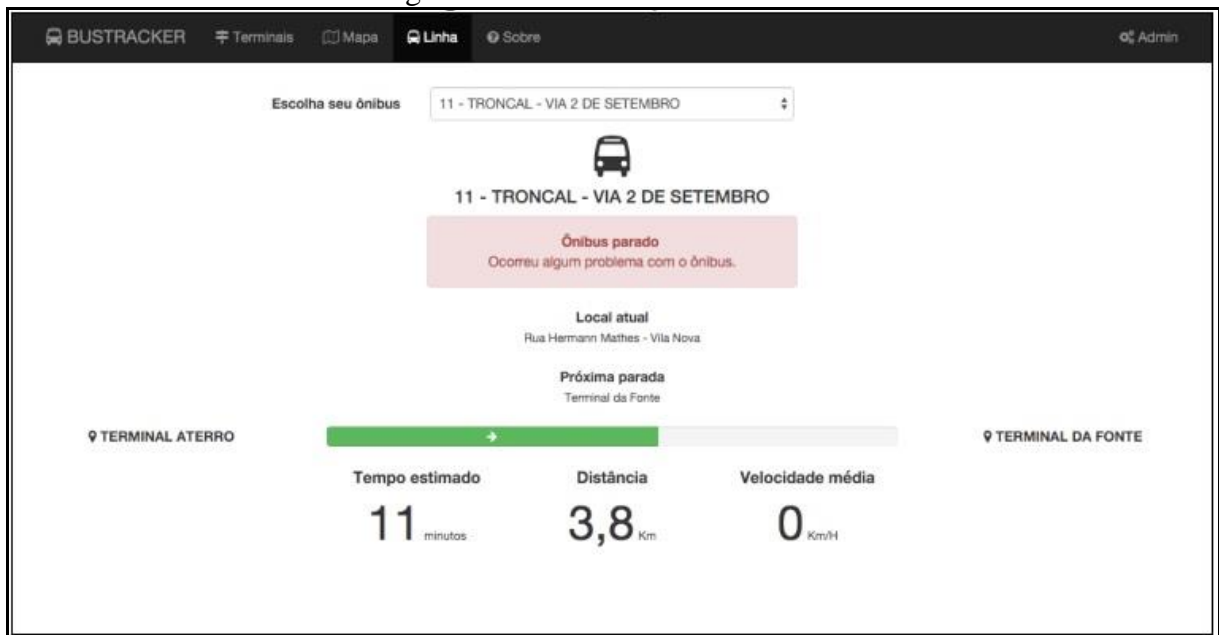
Fonte: Santos Júnior (2009).

Por ter sido desenvolvido em 2009, este sistema possui uma integração junto ao hardware que captura os dados do veículo via OBDII não muito viável. Na ocasião Santos Júnior (2009) utilizou um notebook para requisitar as informações do veículo e transmitir para o dispositivo móvel (SANTOS JÚNIOR, 2009). Porém a ideia de disponibilizar esse tipo de informação ao alcance das mãos foi tão bem aceita que hoje temos diversos aplicativos e softwares que integram diretamente com pequenos dispositivos via Bluetooth.

### 2.3.3 BUSTRACKER: SISTEMA DE RASTREAMENTO PARA TRANSPORTE COLETIVO

O sistema desenvolvido por Vicenzi (2015) tem como objetivo principal disponibilizar uma plataforma onde o usuário possa consultar a localização de ônibus em tempo real. Este projeto foi desenvolvido utilizando duas linguagens de programação sendo, Lua para o microcontrolador ESP8266 e Python para receber informações e realizar o armazenamento dos dados no MongoDB e Django responsável por disponibilizar as informações para os usuários (VICENZI, 2015). O sistema é capaz de calcular o tempo estimado que o veículo chegará ao ponto de destino, mostrar a trajetória e a distância percorrida, e outras informações como podem ser vistos na Figura 7 e Figura 8.

Figura 7 – Tela de consulta Bustracker



Fonte: Vicenzi (2015).

Figura 8 – Tela de consulta no mapa Bustracker



Fonte: Vicenzi (2015).

### 3 DESENVOLVIMENTO

Nesta seção são abordadas as etapas do desenvolvimento do rastreador veicular utilizando OpenWRT. A seção 3.1 exibe os requisitos levantados para o desenvolvimento do projeto. A segunda seção 3.2 demonstra através de diagramas de atividades, diagrama casos de uso e um modelo de entidade e relacionamento como foi desenvolvido este trabalho. Na terceira seção 3.3 está disposta uma descrição detalhada sobre os principais pontos de desenvolvimento, componentes deste trabalho e interface de relação com o usuário. Por fim na seção 3.4 são descritos os resultados obtidos neste trabalho e apresentadas algumas dificuldades encontradas no decorrer do desenvolvimento.

#### 3.1 REQUISITOS

Os requisitos do sistema a ser desenvolvidos são:

- a) o sistema deverá permitir o cadastro de usuário (Requisito Funcional - RF);
- b) o sistema deverá permitir a manutenção do cadastro de usuário (RF);
- c) o sistema deverá permitir que o usuário capture a localização atual do veículo a qualquer momento (RF);
- d) o sistema deverá permitir que o usuário configure o envio da localização do veículo com um intervalo de tempo para e-mails previamente cadastrados (RF);
- e) o sistema deverá permitir recuperação de senha através de e-mail (RF);
- f) o sistema deverá permitir que o usuário capture imagens do interior do veículo a qualquer momento (RF);
- g) o sistema deverá permitir que o usuário configure o envio de imagens do veículo com um intervalo de tempo para e-mails previamente cadastrados (RF);
- h) o sistema deverá utilizar a linguagem PHP (Requisito não Funcional - RNF);
- i) o sistema deverá possuir um usuário *default* para a primeira inicialização, podendo ser alterado após o cadastro (RNF);
- j) o sistema deverá ser desenvolvido para plataforma Web (RNF);
- k) o sistema deverá ter a parte *front end* desenvolvida com Bootstrap, HTML5, JavaScript e JQuery (RNF);
- l) o sistema deverá ser acessível a partir do navegador Google Chrome (RNF);
- m) o sistema deverá ser integrado com o sistema de acesso online Weaved (RNF);
- n) o sistema deverá fazer uso de um modem 3G Huawei E3272 para realizar as transmissões dos dados via e-mail (RNF);
- o) o sistema deverá fazer uso de uma webcam Logitech C270 para capturar as

- imagens do interior do veículo (RNF);
- p) o sistema deverá fazer uso de um módulo de GPS Ublox GY-NEO6MV2 para capturar a localização do veículo (RNF);
  - q) o módulo de GPS deve ser conectado ao Standard Shield através de um adaptador USB/Serial PL-2303 (RNF);
  - r) o sistema deverá ser hospedado no WRTnode (RNF);
  - s) os periféricos deverão ser conectados ao WRTnode através do Standard Shield (RNF);
  - t) o sistema deverá fazer uso da Google Maps JavaScript API para exibir as localizações para o usuário (RNF).

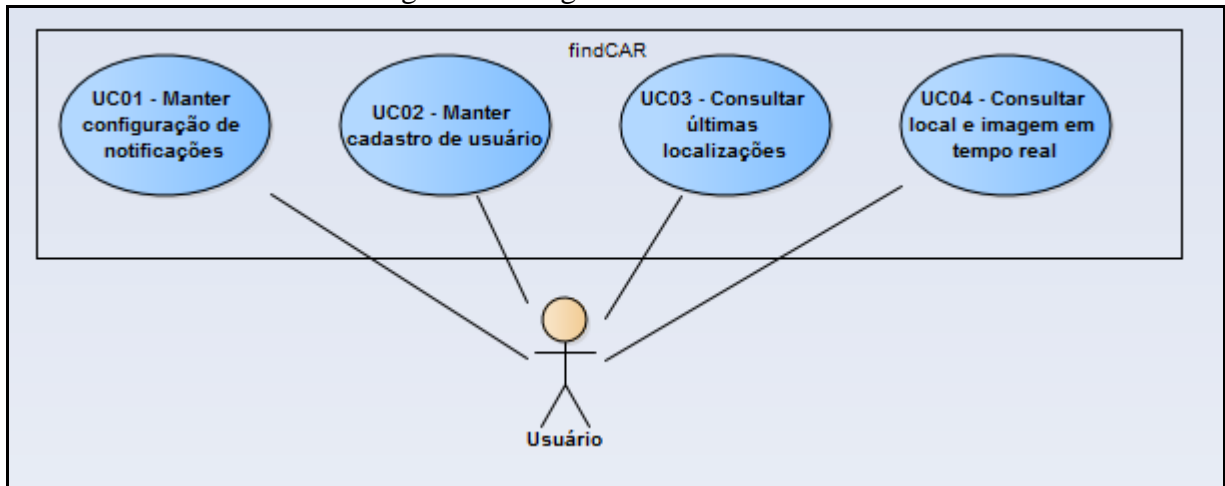
## 3.2 ESPECIFICAÇÃO

A especificação deste trabalho foi desenvolvida na ferramenta de modelagem Enterprise Architect 12. Para a especificação foram montados os diagramas de casos de uso, diagrama de atividades e diagrama de topologia.

### 3.2.1 Diagrama de casos de uso

A Figura 9 exibe os casos de uso do rastreador implementados neste trabalho. O caso de uso UC01 refere-se à ação do usuário em relação ao cadastro de notificações, onde o mesmo pode cadastrar, alterar ou excluir notificações. O caso de uso UC02 refere-se a ação do usuário em relação ao cadastros de usuário, onde o mesmo pode cadastrar ou alterar informações de usuário. O caso de uso UC03 refere-se a ação do usuário em relação a consulta de últimas localizações, onde o mesmo pode consultar a data e a localização onde o veículo esteve através de um mapa. Por fim o caso de uso UC04 refere-se a ação do usuário em relação a consulta da localização e imagem em tempo real, onde o mesmo recebe imagem e local através do mapa atualizados com um intervalo de tempo.

Figura 9 – Diagrama de casos de uso



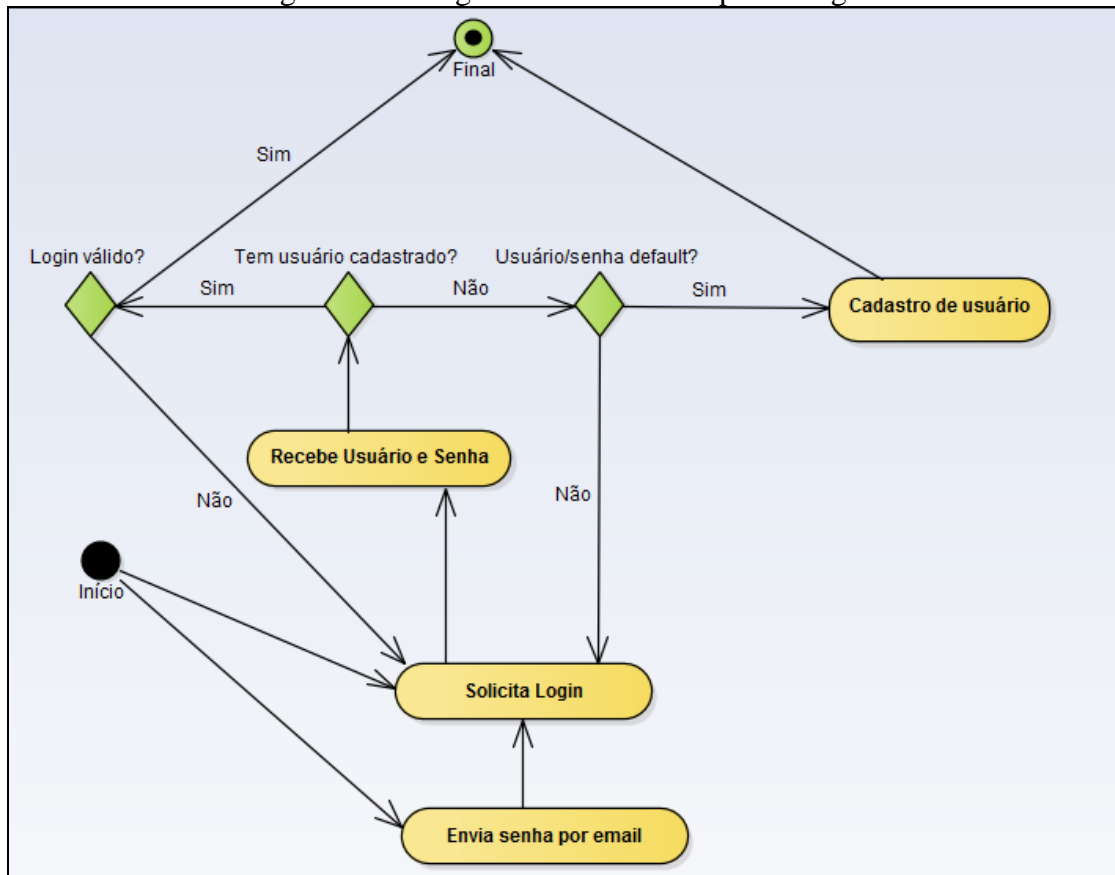
Fonte: Elaborado pelo autor.

### 3.2.2 Diagramas de atividades

Na Figura 10 está demonstrado o diagrama de atividades para o login no sistema. Estão demonstradas as validações para o primeiro acesso do usuário com informações de usuário e senha padrão, cadastros e validação de usuário já cadastrado. Ao receber o input com os dados de usuário e senha o sistema fará a validação verificando se já existe usuário cadastrado. Caso não exista o usuário e senha digitados devem ser os padrões.



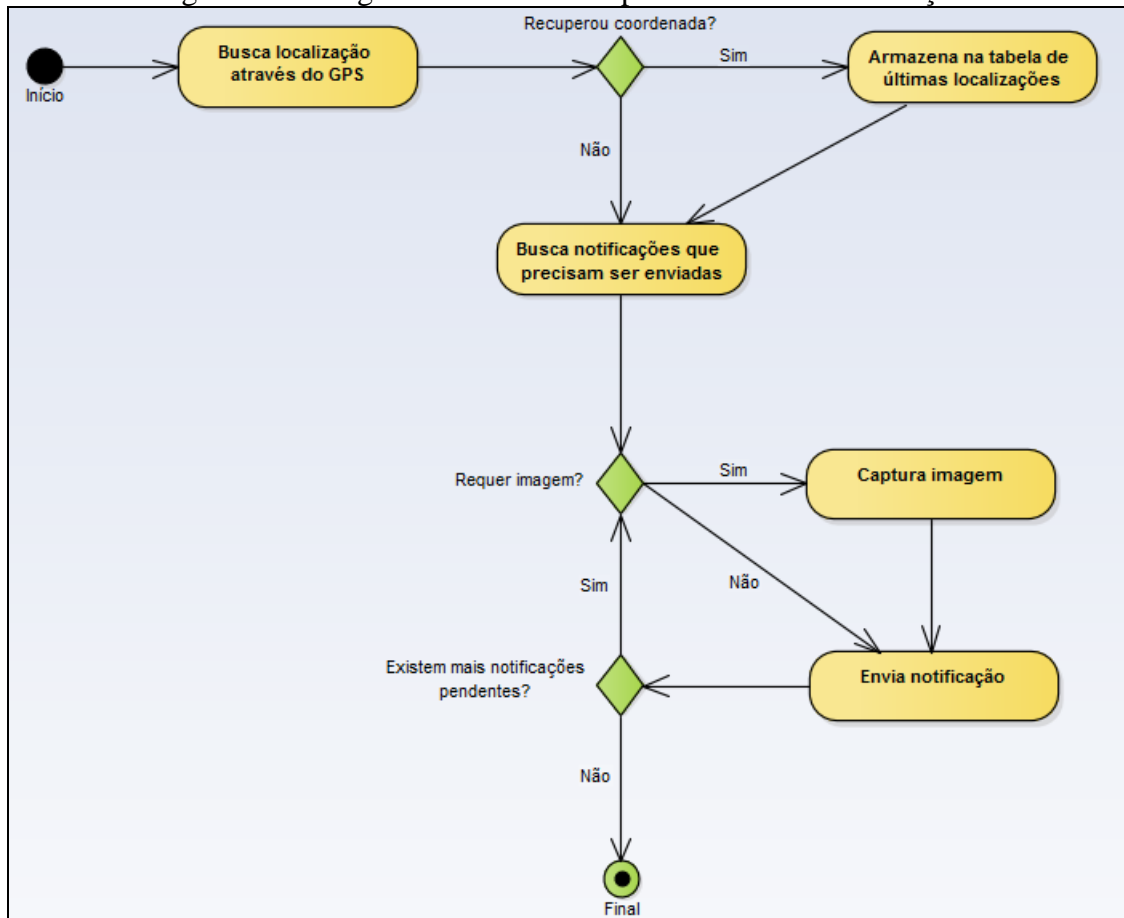
Figura 10 – Diagrama de atividades para o login



Fonte: Elaborado pelo autor.

Na Figura 11 está demonstrado o funcionamento da atividade de envio das notificações, onde também são realizadas as capturas de localização para armazenar na tabela de últimas localizações do veículo. Este processo está sendo descrito com mais detalhes na seção 3.3.7.

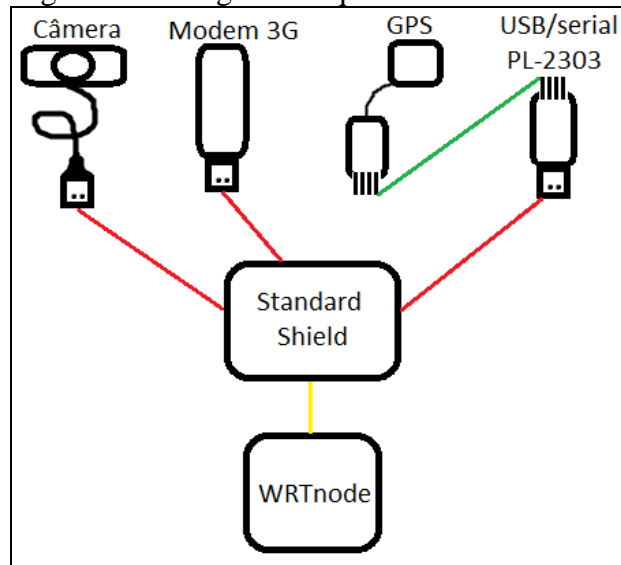
Figura 11 – Diagrama de atividades para o envio de notificações



Fonte: Elaborado pelo autor.

Na Figura 12 está demonstrado um diagrama com a topologia de conexões entre os periféricos e o hardware base do projeto. As conexões destacadas em vermelho são realizadas via USB, a conexão entre o módulo GPS e o adaptador USB/serial em verde é realizada de forma serial e a conexão em amarelo entre o WRTnode e o Standard Shield é realizada diretamente entre as placas via barra pinos.

Figura 12 – Diagrama esquemático de conexões



Fonte: Elaborado pelo autor.

### 3.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas utilizadas e a operacionalidade da implementação.

#### 3.3.1 Técnicas e ferramentas utilizadas

Para o desenvolvimento da aplicação web foi utilizado o editor de código fonte Sublime Text build 3126. Para a parte de *front end* foram utilizados HTML5, Bootstrap, JavaScript e JQuery. Para a parte de *back end* foi utilizada a linguagem PHP5 e a base de dados foi armazenada em MySQL. A compilação de pacotes de drivers e módulos de funções para o sistema operacional do WRTnode foi realizada em uma máquina virtual Oracle VM Virtual Box com sistema operacional Ubuntu 14.0.4 de 64bits. Foram utilizados também o emulador de terminal Tera Term 4.91, o SSH File Transfer Protocol (SFTP) e WinSCP para realizar o acesso e a transmissão dos fontes e arquivos utilizados no desenvolvimento.

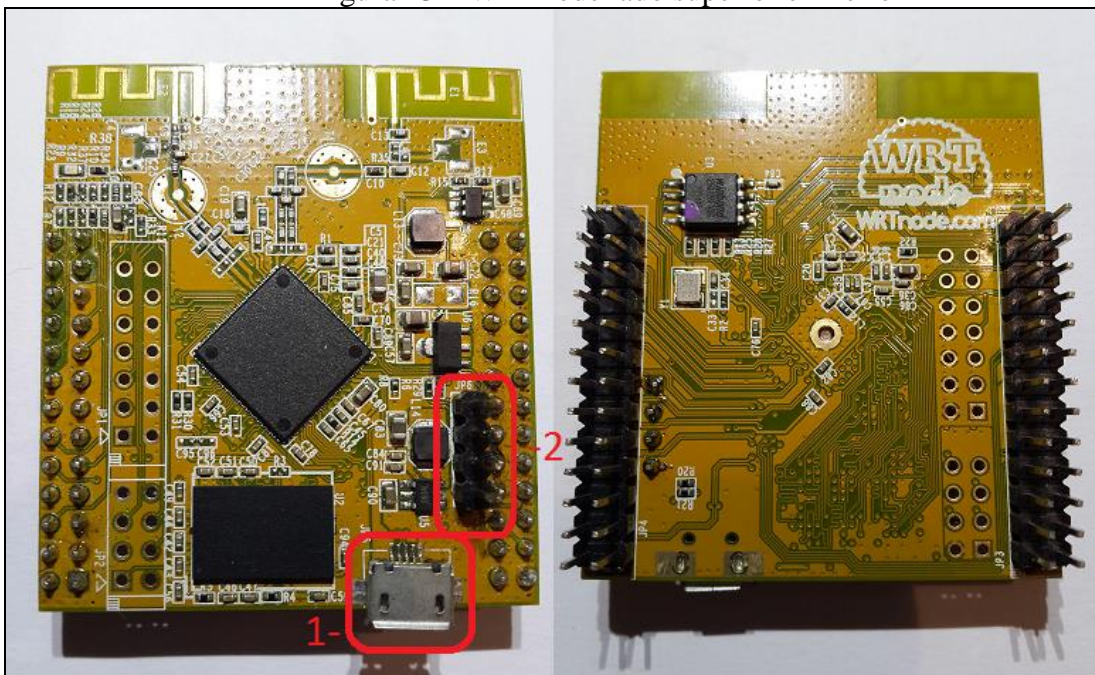
Para a exibição da localização do veículo foi utilizada a Application Programming Interface (API) Maps JavaScript API da Google. As coordenadas são recuperadas com o uso de um módulo de GPS GY-NEO6MV2 conectado ao WRTnode utilizando um adaptador USB/Serial PL2302. Compõem o rastreador ainda uma webcam Logitech C270 para fazer a captura das imagens do interior do veículo, um modem 3G Huawei 3272 e um Shield WRTnode para permitir a conexão de periféricos USB e permitir conexão a uma rede via RJ45.

### 3.3.2 Hardware e periféricos do rastreador

Nesta seção serão expostos os periféricos e o hardware que servem de base para o rastreador. O custo das peças e periféricos foi disponibilizado no apêndice B através do Quadro 11.

O WRTnode foi o hardware adotado como base para o desenvolvimento do rastreador, pois o seu firmware tem como base o Linux o que permite sua utilização e configuração para inúmeras aplicações. Seu firmware possui vários pacotes e módulos já compilados, além de vários que ainda podem ser compilados e desenvolvidos para o mesmo. Este dispositivo é bem compacto o que o torna muito útil para o desenvolvimento de projetos que exigem certa portabilidade (Figura 13). Ainda na Figura 13 pode-se observar destacado em vermelho indicado pelo número 1 a porta USB 2.0 e destacado pelo número 2 a porta serial que é utilizada para realizar o acesso ao WRTnode através de um computador para realizar as configurações necessárias.

Figura 13 – WRTnode lado superior e inferior



Fonte: Elaborado pelo autor.

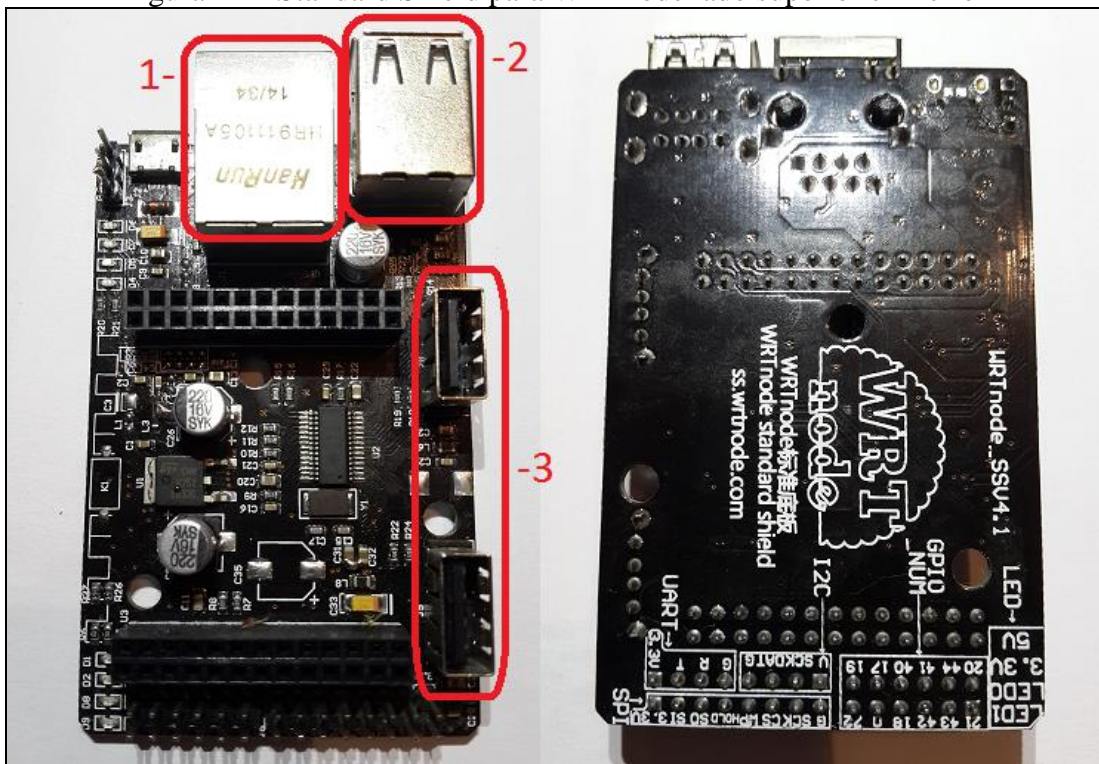
Este WRTnode de modelo MT7620 possui as seguintes configurações:

- Central Processing Unit (CPU) de 580 Mega-hertz (MHz);
- 512 Mega-bit (Mbit) Random Access Memory (RAM) Double Data Rate 2 (DDR2);
- 128 Mbit Serial Peripheral Interface (SPI) Flash Read-only Memory (ROM);
- 300 Mbit Wireless Fidelity (Wi-Fi) 2 Transmitter 2 Receiver (2T2R) 802.11n 2.4 Giga-hertz (GHz);

- e) 23 General Purpose Input/Output (GPIOs);
- f) Joint Test Access Group (JTAG);
- g) Universal Asynchronous Receiver/Transmitter (UART) Lite;
- h) USB 2.0;
- i) OpenWRT no kernel do Linux 3.10.44;
- j) Alimentação via MicroUSB 5 Volts (V).

O Standard Shield V4.1 escolhido para compor o hardware do rastreador possui 4 portas USB e uma porta RJ45. Esta placa de expansão permite que o WRTnode se conecte aos demais periféricos que constituem o rastreador.

Figura 14 – Standard Shield para WRTnode lado superior e inferior

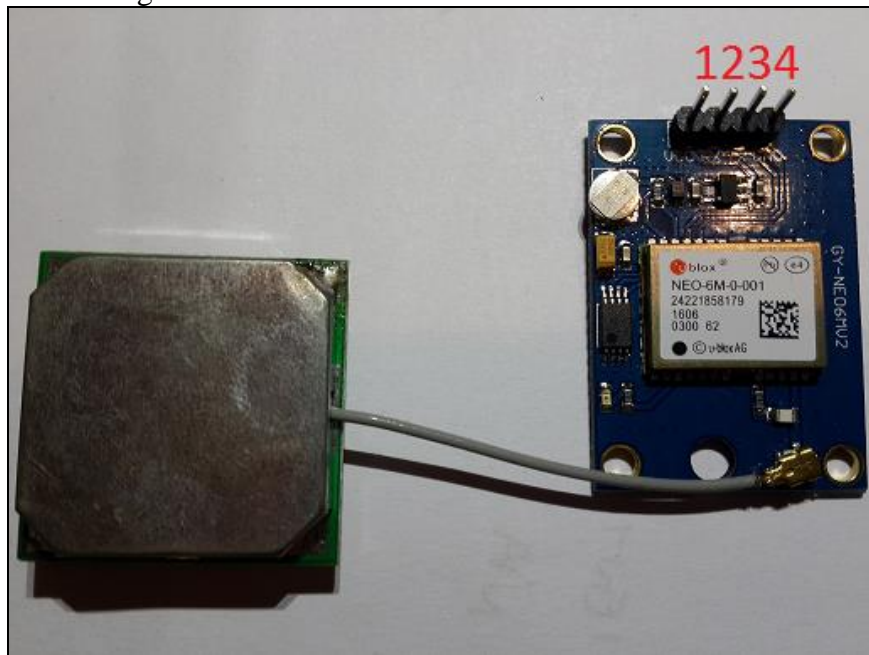


Fonte: Elaborado pelo autor.

Na Figura 14 estão identificados em vermelho, com o número 1 a porta *ethernet* RJ45 para conexão via rede cabeada, com o número 2 estão destacadas duas portas USB na orientação horizontal e por fim com o número 3 são destacadas duas outras portas USB, porém com orientação vertical.

O módulo GPS utilizado no projeto foi o Ublox GY-NEO6MV2 com antena de cerâmica. Este módulo possui uma interface de comunicação serial de 4 pinos Voltage Current Clamp (VCC), Receiver (RX), Transmitter (TX) e Ground (GND), representados na Figura 15 através dos números 1,2,3 e 4 respectivamente. Este módulo é bem compacto o que o torna uma ótima opção para aplicação em dispositivos portáteis.

Figura 15 – Módulo de GPS Ublox GY-NEO6MV2



Fonte: Elaborado pelo autor.

Para realizar a conexão do rastreador com a internet, foi utilizado o modem 3G/4G Huawei E3272 (Figura 16). Este modem permite uma conexão com taxa de transferência de até 21,6 Mbps e possui a tecnologia HiLink que emula uma porta ethernet quando plugado via porta USB. Esta tecnologia simplifica a conexão, pois para estar conectado à internet basta plugá-lo à porta USB.

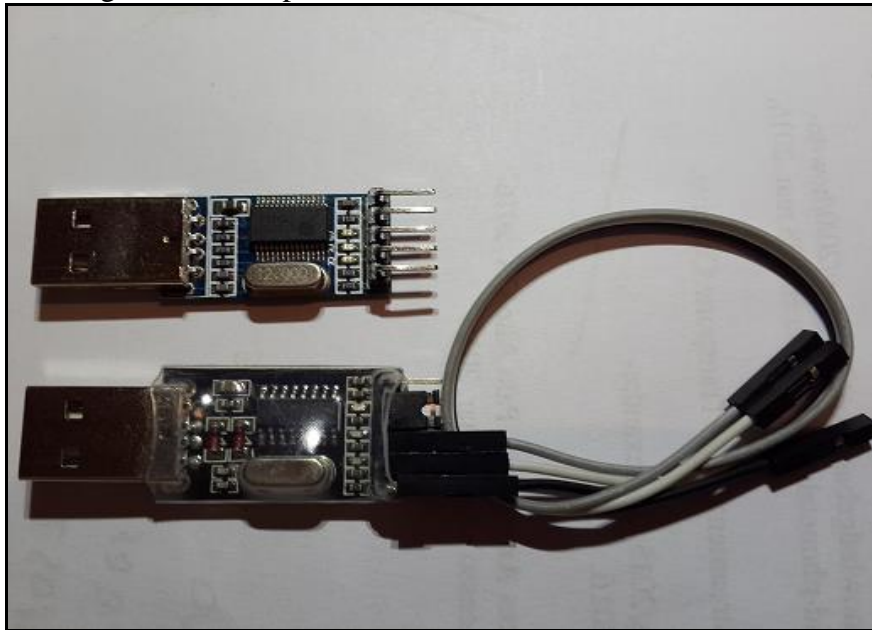
Figura 16 – Modem 3G/4G Huawei E3272



Fonte: Elaborado pelo autor.

Para realizar a conexão do módulo GPS com o Shield do WRTnode foi utilizado um adaptador USB/Serial com Circuito Integrado (CI) PL-2303 com protocolo de transmissão de dados RS232. Também foi utilizado um adaptador USB/Serial, desta vez com CI CH340G para conectar o WRTnode ao notebook para que fosse possível realizar as configurações necessárias via Tera Term que é um emulador de terminal. Na Figura 17 está sendo exibida na parte superior o adaptador com CI PL-2303 e na parte inferior da imagem o adaptador com CI CH340G.

Figura 17 – Adaptadores USB/Serial PL-2303 e CH340G



Fonte: Elaborado pelo autor.

A webcam escolhida para compor o conjunto de periféricos do rastreador, foi uma Logitech C270 (Figura 18). A escolha por esta câmera foi definida devido a boa qualidade da imagem capturada, com resolução de 720 pixels. Além da boa qualidade de imagem o que cooperou para a escolha deste hardware foi o resultado positivo que alguns usuários de WRTnode compartilharam no fórum específico para este hardware embarcado.

Figura 18 – Webcam Logitech C270



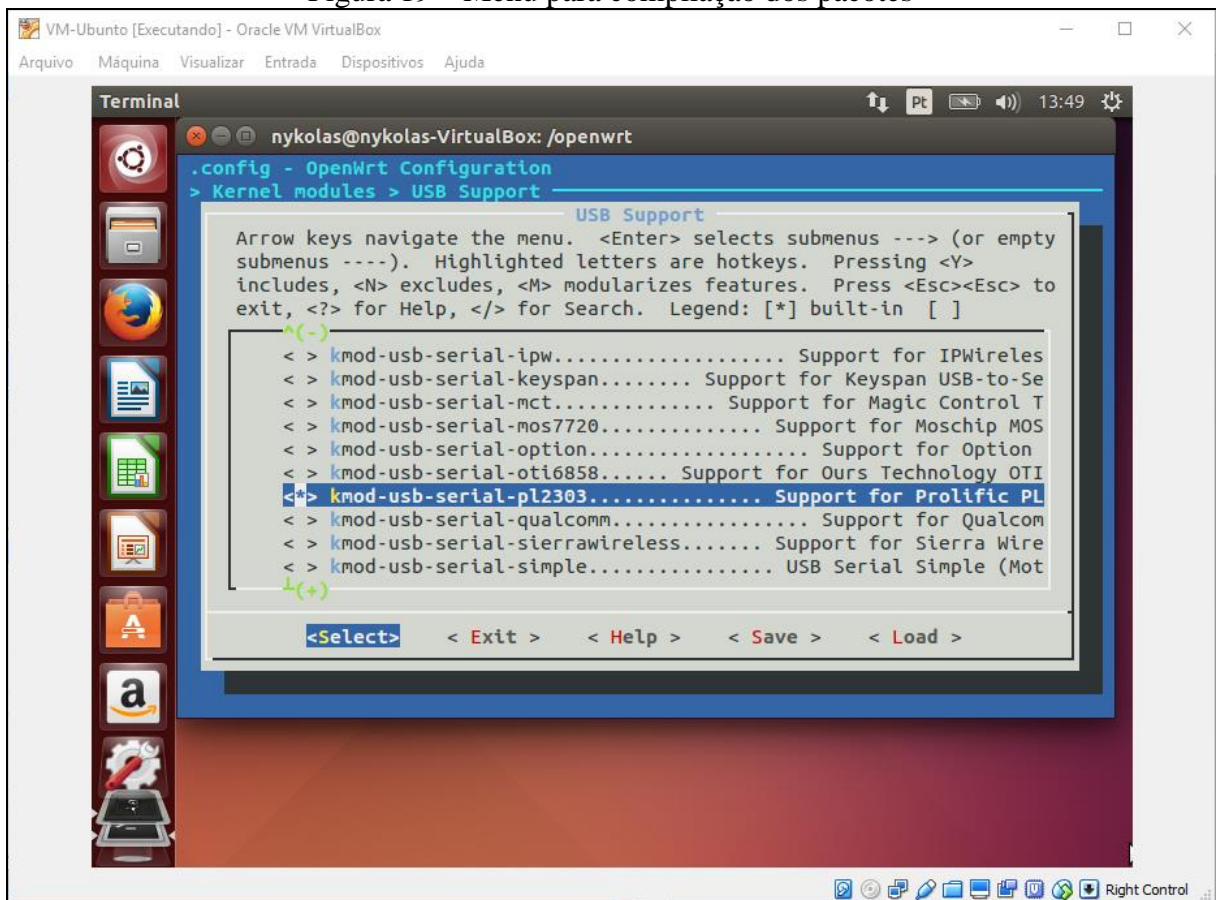
Fonte: Elaborado pelo autor.

### 3.3.3 Compilação de módulos e pacotes

A distribuição do Linux OpenWRT, dispõe de vários drivers, pacotes e módulos para serem utilizados. O menuconfig é um software disponibilizado para realizar a compilação dos pacotes para instalação no OpenWRT vide Figura 19. Para utilizá-lo é necessário baixar os arquivos do repositório do github no link <https://github.com/WRTnode/openwrt.git> e realizar a montagem do menu. Após a montagem do menu, pode-se escolher a versão do firmware para qual há a necessidade de compilar os drivers, módulos e pacotes, selecionar os desejados e compilar. Na Figura 19 está sendo exibido um exemplo de seleção do pacote `kmod-usb-serial-pl2303` que é o pacote de driver para o adaptador USB/serial com CI PL-2303.



Figura 19 – Menu para compilação dos pacotes



Fonte: Elaborado pelo autor.

Para possibilitar o desenvolvimento do sistema, a hospedagem do mesmo e o acesso a todos os periféricos utilizados neste trabalho, foi necessária a compilação e instalação de diversos pacotes (a relação dos pacotes utilizados pode ser vista no Apêndice A). Alguns dos pacotes foram instalados para suprir a necessidade de base para o software, como é o caso dos pacotes relacionados ao servidor MySQL e PHP, tanto o serviço quanto as extensões. Além disso, foram necessárias instalações de pacotes relacionados ao suporte para o desenvolvimento como é o caso de pacotes com `minicom` e `picocom` que são aplicações para “escutar” as portas seriais. Outra aplicação que foi necessária para auxiliar durante o desenvolvimento foi o servidor SFTP, para que fosse possível transmitir os arquivos fonte para o WRTnode via WinSCP.

### 3.3.4 Captura de imagem

Inicialmente para conseguir ter acesso à câmera e realizar a captura de imagens através dela, faz-se necessária a instalação dos pacotes `kmod-video-uvc` e `mjpg-streamer`. A instalação do pacote `mjpg-streamer` depende de outros dois pacotes que também precisam ser instalados, são eles `libpthread` e `libjpeg`. Feitas as instalações dos pacotes necessários,

basta habilitar o uso do pacote `mjpg-streamer` e inicializar o serviço do mesmo, para que seja possível capturar as imagens da câmera através de um browser. Por default, para acessar a imagem da câmera via browser, é necessário requisitar via método GET para a porta 8080 passando o parâmetro `action` com valor `snapshot` para capturar uma foto ou `stream` para capturar uma *stream* de imagens. No Quadro 1, tem-se um exemplo de requisição utilizado no rastreador para capturar uma imagem e criar um arquivo de imagem com o nome `captura.jpeg`.

Quadro 1 – Exemplo de captura de imagem

24	<code>//Realiza a captura de uma imagem</code>
25	<code>exec("wget localhost:8080/?action=snapshot -O captura.jpeg");</code>

Fonte: Elaborado pelo autor.

No desenvolvimento do rastreador optou-se por armazenar a imagem sempre com o mesmo nome, para que exista somente um arquivo de imagem armazenado na memória. Esta escolha foi tomada pois o espaço para armazenamento de dados no dispositivo é muito pequeno. As imagens capturadas para o envio de notificações são geradas da mesma forma e isto acontece sempre que chega o horário correto para a execução das notificações.

### 3.3.5 Captura da localização

Para a captura da localização que é feita do módulo de GPS foi necessário utilizar um adaptador USB/Serial. Para esta conexão foi utilizado um adaptador com CI PL-2303 para que este adaptador fosse reconhecido e utilizado de forma correta pelo sistema operacional foi necessário instalar o pacote `kmod-usb-serial-pl2303` com o driver para este periférico.

Para “escutar” o retorno do módulo GPS e retornar a latitude e a longitude, foi desenvolvido um script em php chamado `req_busca_local.php`, o qual é chamado via ajax na tela `real_time`. Para a rotina de envio de notificações, foi criada uma função para retornar a localização. No Quadro 2 pode-se ver como é feita a abertura de uma porta serial utilizando do comando `dio_open` na linha 54, entre as linhas 58 e 63 são realizadas configurações de comunicação, para conseguir “escutar” a porta serial de forma correta. A leitura dos dados ocorre efetivamente na linha 67 onde é realizada a chamada da função `dio_read` que fará a leitura de 1256 bytes.

Quadro 2 – Fonte para escutar porta serial

```
54 $fd = dio_open('/dev/ttyUSB0', O_RDWR | O_NOCTTY | O_NONBLOCK);
55
56 dio_fcntl($fd, F_SETFL, O_SYNC);
57
58 dio_tcsetattr($fd, array(
59     'baud' => 9600,
60     'bits' => 8,
61     'stop' => 1,
62     'parity' => 0
63 ));
64
65 sleep(2);
66
67 $data = dio_read($fd, 1256);
```

Fonte: Elaborado pelo autor.

O módulo GPS retorna os dados no padrão NMEA, portanto, é necessário fazer a mineração no retorno para capturar somente a latitude e a longitude, que são as informações utilizadas pelo rastreador. O padrão NMEA retorna basicamente 4 sentenças, sendo elas \$GPGGA, \$GPGSA, \$GPGSV e \$GPRMC. No Quadro 3 está exposto o trecho do código onde são recuperadas as informações de localização. A latitude e a longitude se encontram na sentença \$GPGGA então para recuperar estas informações realiza-se um `substr` para primeiramente separar somente esta sentença entre as linhas 69 e 74. Após separar somente a sentença que possui as informações relevantes faz-se um `explode` para retornar um array com as informações, linha 76. Caso não encontre a localização a função retornará “SEM\_SINAL”, controle feito entre as linhas 78 e 81. Entre as linhas 83 e 93 é realizada a conversão para das coordenada que no padrão NMEA vem no formato Degrees Minutes Seconds (DMS) para o formato Decimal Degrees (DD). É realizada esta conversão para passar para a API do Google que faz a exibição do mapa.

Quadro 3 – Captura de localização

```

69     if ($data) {
70         $aux = substr($data, strpos($data, '$GPGGA'));
71         $latlong = substr($aux, 0, (strpos($aux, '$GPGSA')-1);
72     }
73     $retorno = explode(',', $latlong);
74
75     if(empty($retorno[2]))
76     {
77         return 'SEM_SINAL';
78     }
79
80     $latitude = substr($retorno[2],0,2)+(substr($retorno[2],2)/60);
81     if($retorno[3] = 'S')
82     {
83         $latitude = -1 * $latitude;
84     }
85     $longitude = substr($retorno[4],0,3)+(substr($retorno[4],3)/60);
86     if($retorno[5] = 'W')
87     {
88         $longitude = -1 * $longitude;
89     }
90     $var = $latitude . ', ' . $longitude;
91
92     return $var;
93
94
95

```

Fonte: Elaborado pelo autor.

### 3.3.6 Google Maps JavaScript API

Para a exibição da localização do veículo, foi utilizada a API do Google que é chamada somente via JavaScript. Esta API permite o uso de diversos parâmetros de configuração para exibição. Para o rastreador foram utilizados somente os parâmetros `center`, `zoom` e `marker`. O parâmetro `center` define que ponto do mapa, definido por latitude e longitude será apresentado no centro da área do mapa exibido na tela. O parâmetro `zoom`, define a quantidade de zoom que será exibida no mapa, a grande maioria das imagens existentes no maps permitem zoom de até 18 vezes. Por sua vez o parâmetro `marker`, seta um marcador de localização no mapa, este marcador pode ser customizado, utilizando os parâmetros `icon` e `title` para setar uma imagem e uma descrição para o marcador respectivamente.

Esta API permite também a visualização do mapa por diversas camadas, que podem ser definidas através do parâmetro `mapTypeId`. Os tipos de camadas de mapas são definidos através das seguintes constantes:

- a) `MapTypeId.ROADMAP`: exhibe o mapa com ruas e rodovias. Tipo default de

- mapa;
- b) MapTypeId.SATELLITE: exibe o mapa com imagens de satélite do Google Earth;
  - c) MapTypeId.HYBRID: exibe o mapa unindo dados do mapa de ruas e rodovias com imagens de satélite;
  - d) MapTypeId.TERRAIN: exibe o mapa com informações de relevo e terreno.

No Quadro 4 está a chamada da API onde é realizada a montagem do mapa que é exibida na tela principal do sistema, na área de tempo real. No momento da chamada da API as coordenadas já estão prontas para serem usadas, retornadas do objeto `req_busca_local.php`. A recuperação da localização feita neste objeto é executada da mesma forma demonstrada na seção 3.3.5.

Quadro 4 – Chamada da API Maps JavaScript

```

42 function atualiza_pos(){
43     $.ajax({
44         method: "POST",
45         url: "req_busca_local.php",
46         success: function(data){
47     if(data != "SEM_SINAL")
48     {
49     var latlong = data.split("|");
50
51     var mapCanvas = document.getElementById("map");
52     var mapOptions = {
53         center: new google.maps.LatLng(latlong[0], latlong[1]),
54         zoom: 17
55     }
56     var map = new google.maps.Map(mapCanvas, mapOptions);
57
58     var image = 'images/car_location_marker.png';
59
60     var marker = new google.maps.Marker({
61         position: new google.maps.LatLng(latlong[0], latlong[1]),
62         mapTypeId: google.maps.MapTypeId.HYBRID,
63         title: "Seu veículo está aqui!",
64         icon: image
65     });
66
67     // To add the marker to the map, call setMap();
68     marker.setMap(map);
69     map.setMapTypeId(google.maps.MapTypeId.HYBRID);
70 }
71     setTimeout(function(){atualiza_pos()}, 2000);
72     }
73 })
74 }

```

Fonte: Elaborado pelo autor.

### 3.3.7 Processo de execução das notificações

O processo de execução de notificações ocorre de forma independente do sistema, sendo que para que isso ocorra, foi utilizado um recurso do sistema operacional chamado `cron`. Este recurso funciona como uma tarefa agendada que pode ser configurada com um intervalo de tempo, de minutos, horas, dias do mês, mês, dias da semana. Para executar as notificações foi criada uma configuração com um intervalo de tempo de execução de 1

minuto, onde será recuperado o local a partir do módulo de GPS conectado ao WRTnode através de uma das portas USB do Standard Shield.

O evento configurado irá executar o script `proc_notif.php` exibido no Quadro 5 entre as linhas 17 e 21 é realizada uma consulta na base de dados para retornar quais notificações precisam ser executadas com base na diferença entre o horário atual e o horário da última execução. Além de realizar a execução das notificações o script `proc_notif.php` atualiza a tabela `UltimosLocais` que é responsável por armazenar a data e hora das coordenadas que vão sendo capturadas enquanto o rastreador estiver ligado, esta operação é realizada entre as linhas 10 e 15.

Quadro 5 – Script `proc_notif.php`

```

1  <?php
2
3  include("conecta_db.php");
4
5  date_timezone_set('America/Sao_Paulo');
6
7  //Realiza a busca do local através do GPS
8  $latlong = capturaLocal();
9
10 if($latlong != 'SEM_SINAL'){
11
12     $sql = 'insert into UltimosLocais (data_local, latlong) values (sysdate(),\' ' . $latlong . '\')';
13
14     $retval = mysql_query($sql);
15 }
16
17 $sql = 'select cn.titulo, cn.conteudo, cn.destinatarios, cn.codigo
18        from ConfigNotif cn
19        where cn.interval_exec <= IFNULL(TIMESTAMPDIFF(MINUTE,IFNULL(
20        cn.dat_ult_exec,sysdate()),sysdate()),0) or cn.dat_ult_exec is null;';
21
22 $result = mysql_query($sql);
23
24 while ($row = mysql_fetch_row($result)) {
25     if($row[1] == 'I')
26     {
27         //Realiza a captura de uma imagem
28         exec("wget localhost:8080/?action=snapshot -O captura.jpeg");
29
30         exec('mailsend -smtp smtp.gmail.com -port 465 -t ' . $row[2] . ' -f root@openwrt -sub "' .
31             utf8_decode($row[0]) . '" -ssl -auth -user tccfindcar@gmail.com -pass "f1ndc4rs" -attach "/
32             www/tcc/captura.jpeg" -M "A Imagem em anexo foi capturada em ' . date('Y-m-d H:i') . '"');
33     }
34 }

```

Fonte: Elaborado pelo autor.

### 3.3.8 Desenvolvimento *front end*

Para o desenvolvimento da interface com o usuário, foram utilizados HTML5, CSS3 e Bootstrap3. Para que o sistema permaneça sempre com o mesmo menu na parte superior, sem necessitar fazer cópia de fonte para todos os arquivos, foi criado um único HTML de menu que é incluído a todas as telas do sistema no arquivo PHP de cada uma delas. Desta forma quando houver alguma mudança no menu, a alteração poderá ser feita em apenas um local. Para cada tela do sistema temos então necessariamente um arquivo PHP e um arquivo HTML pelo menos.

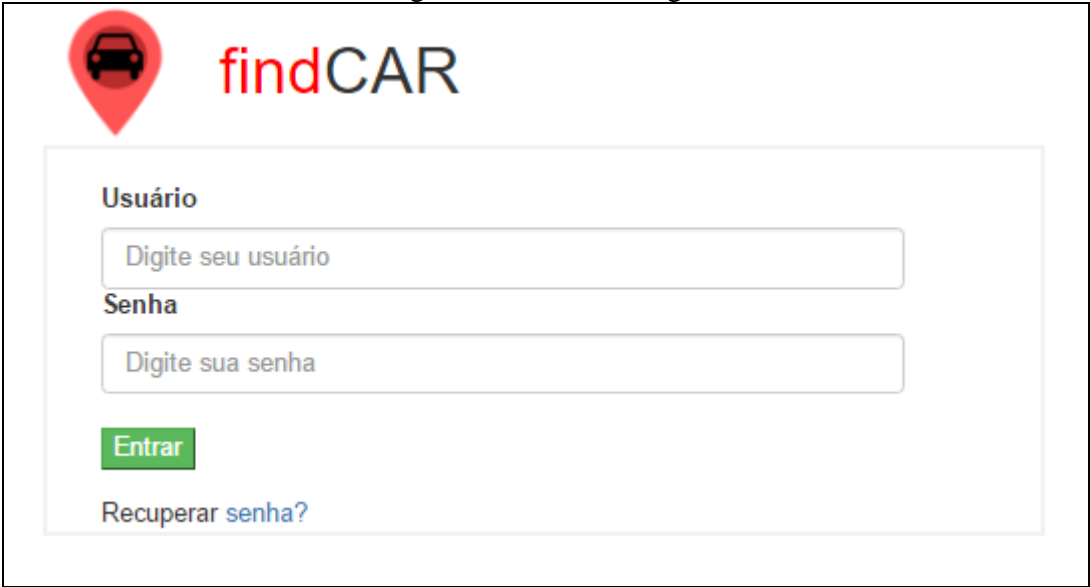
Para o menu foi utilizada a classe `nav-tabs`, esta classe é responsável por aplicar os efeitos do menu, como a forma, a cor, o layout e etc. O Bootstrap auxilia na criação de uma interface melhor acabada e de mais fácil entendimento por parte do usuário de uma forma muito mais rápida. A interface do sistema utilizada basicamente modais para as telas de cadastro, alterações e mensagens de alerta do sistema. Para estas telas foi utilizada a classe `modal fade` do Bootstrap e para cada uma foram feitas pequenas modificações, como adição de botões para fechar, cancelar ou aceitar.

### 3.3.9 Operacionalidade da implementação

Nesta seção é abordada a operacionalidade do sistema de rastreamento FINDCAR desenvolvido neste trabalho. Estão demonstradas as telas de cadastro e consulta relacionadas nos casos de uso.

Ao acessar o sistema o usuário irá deparar-se primeiramente com a tela de login (Figura 20). Esta tela solicita um usuário e senha para permitir o acesso ao sistema de rastreamento. Caso seja o primeiro login no sistema, deverão ser utilizados usuário e senha default ambas as informações foram definidas como `admin`. Feito o login com usuário default, o sistema será redirecionado para a tela de cadastro de usuários, para que o usuário crie uma conta de usuário com as informações de sua preferência (Figura 21).

Figura 20 – Tela de login



Fonte: Elaborado pelo autor.

Figura 21 – Tela de cadastro de usuário

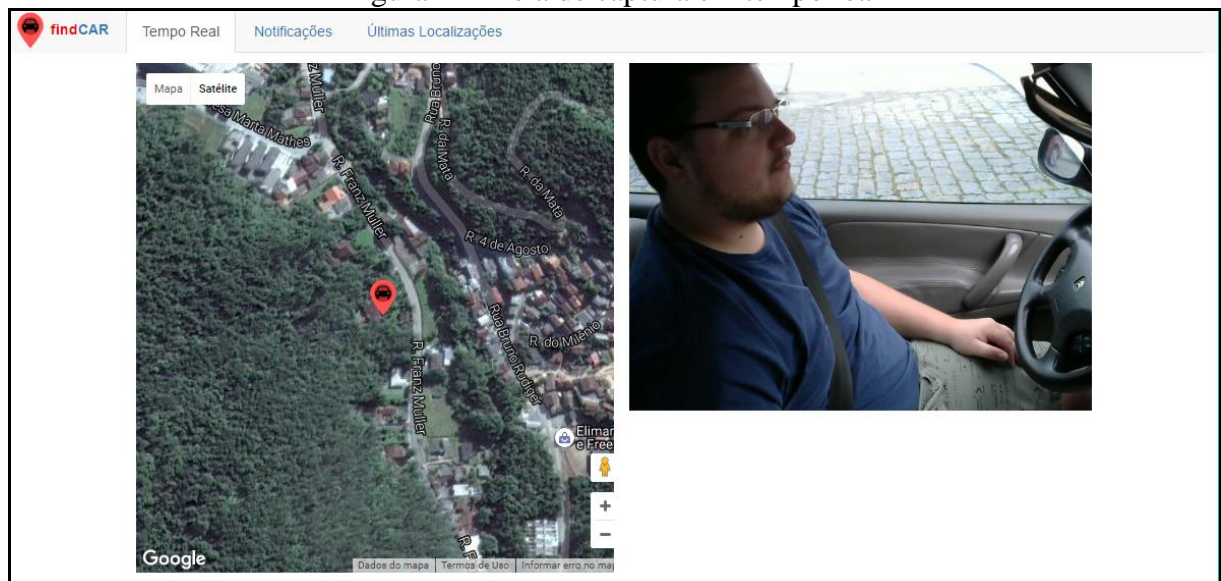
The registration form consists of the following elements:

- Usuário:** A text input field with the placeholder text 'Usuário'.
- Senha:** A text input field with the placeholder text 'Senha'.
- Confirmação de senha:** A text input field with the placeholder text 'Confirmação de senha'.
- E-mail:** A text input field with the placeholder text 'E-mail para recuperação de senha'.
- Salvar:** A green button with white text.

Fonte: Elaborado pelo autor.

Caso o usuário já tenha realizado o cadastro no sistema FINDCAR e tenha logado com as suas informações de usuário e senha, ele será redirecionado para a tela Tempo Real (Figura 22). Nesta tela o usuário recebe a localização do veículo através do mapa e uma imagem capturada através da câmera. Estas duas informações são capturadas durante o tempo que o usuário permanece na tela e são atualizadas em intervalo de poucos segundos. Este intervalo pode variar em alguns segundos dependendo da disponibilidade do hardware fornecedor da informação.

Figura 22 –Tela de captura em tempo real



Fonte: Elaborado pelo autor.



O sistema dispõe de uma tela com uma lista das configurações de notificações realizadas pelo usuário (Figura 23). Nesta lista estão disponíveis todas as informações das notificações configuradas e também localizados bem a direita na lista estão as opções de alterar e excluir as configurações.

Figura 23 – Tela de consulta de notificações

The screenshot shows the 'Notificações' tab selected. At the top left is a red car icon and the 'findCAR' logo. Below it is a green '+ Adicionar' button. The main content is a table with the following data:

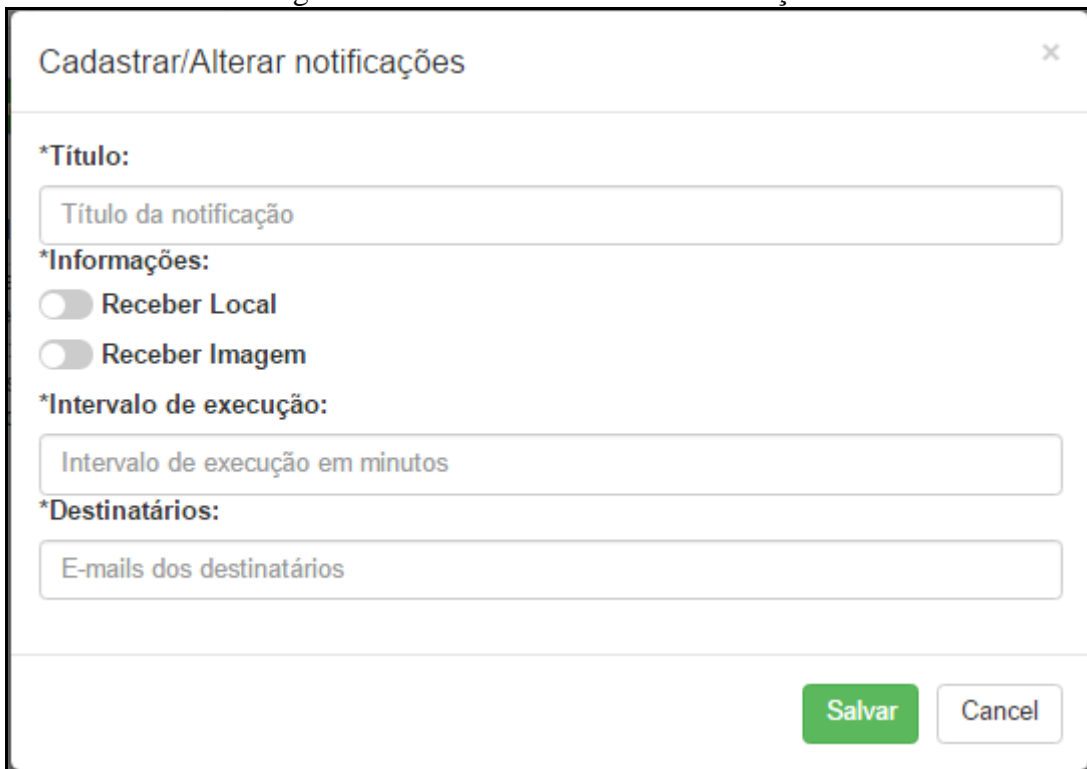
Título da Notificação	Informações	Intervalo de Execução	Destinatários	Ações
Local e imagem do veículo		15min	nykbaum@hotmail.com, rejiluh@gmail.com	Alterar Excluir
Imagem do veículo		10min	nykbaum@hotmail.com	Alterar Excluir
Local do veículo		15min	nykbaum@hotmail.com, nykbaum@gmail.com	Alterar Excluir

\*Todas as notificações serão executadas a primeira vez ao salvar!

Fonte: Elaborado pelo autor.

Ao clicar no botão +Adicionar é exibida uma tela de cadastro de notificações (Figura 24) onde é solicitado o preenchimento obrigatório dos campos Título, Informações, Intervalo de execução e Destinatários. A mesma tela é exibida quando o usuário clica sobre o botão Alterar em algum item da lista, porém, neste caso a tela já é carregada com os valores da notificação desejada vide Figura 25.

Figura 24 – Tela de cadastro de notificações



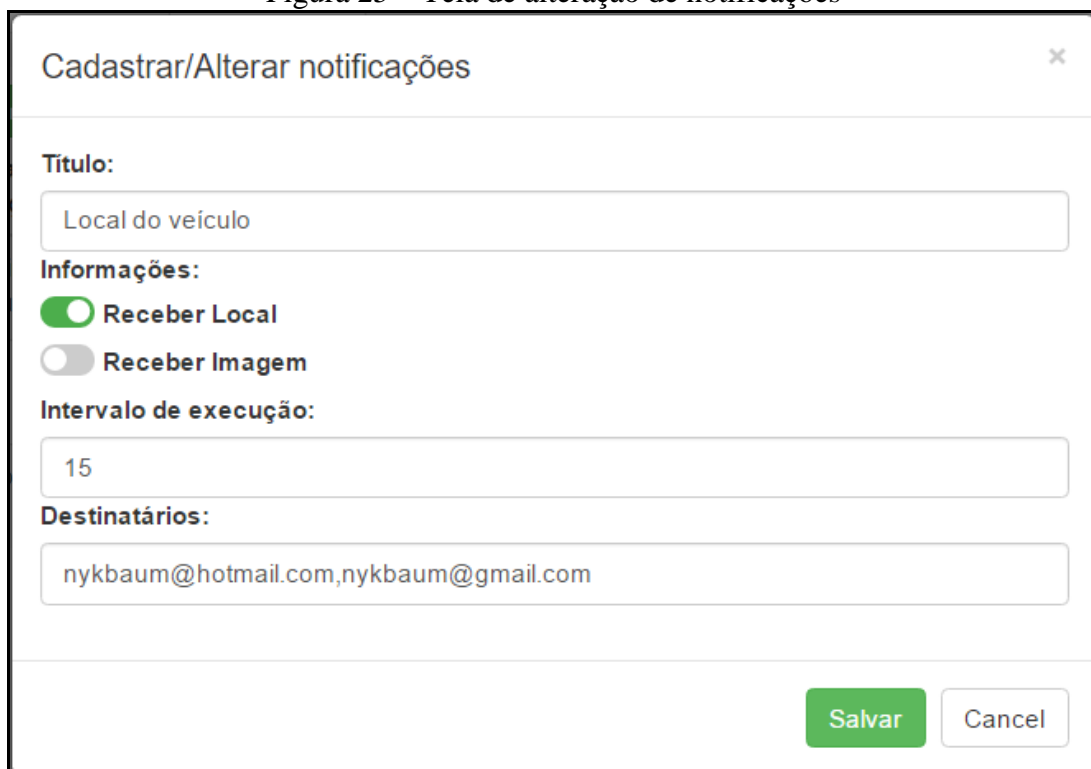
The screenshot shows a web form titled "Cadastrar/Alterar notificações" with a close button (X) in the top right corner. The form contains the following fields and controls:

- \*Título:** A text input field containing the placeholder text "Título da notificação".
- \*Informações:** Two toggle switches, both currently turned off:
  - Receber Local
  - Receber Imagem
- \*Intervalo de execução:** A text input field containing the placeholder text "Intervalo de execução em minutos".
- \*Destinatários:** A text input field containing the placeholder text "E-mails dos destinatários".

At the bottom right of the form, there are two buttons: a green "Salvar" button and a white "Cancel" button.

Fonte: Elaborado pelo autor.

Figura 25 – Tela de alteração de notificações



The screenshot shows the same "Cadastrar/Alterar notificações" form, but with the following changes:

- Título:** The text input field now contains "Local do veículo".
- Informações:** The "Receber Local" toggle switch is now turned on (green), while "Receber Imagem" remains off.
- Intervalo de execução:** The text input field now contains the number "15".
- Destinatários:** The text input field now contains the email addresses "nykbaum@hotmail.com,nykbaum@gmail.com".

The "Salvar" and "Cancel" buttons remain at the bottom right.

Fonte: Elaborado pelo autor.

Para realizar a consulta das últimas localizações onde o veículo esteve o usuário pode acessar a tela *Últimas Localizações* (Figura 26). Esta tela possui uma lista de localizações,

ordenada do registro mais recente para o mais antigo. Para realizar a consulta da localização no mapa o usuário deve clicar no botão Consultar.

Figura 26 – Tela de consulta de últimas localizações

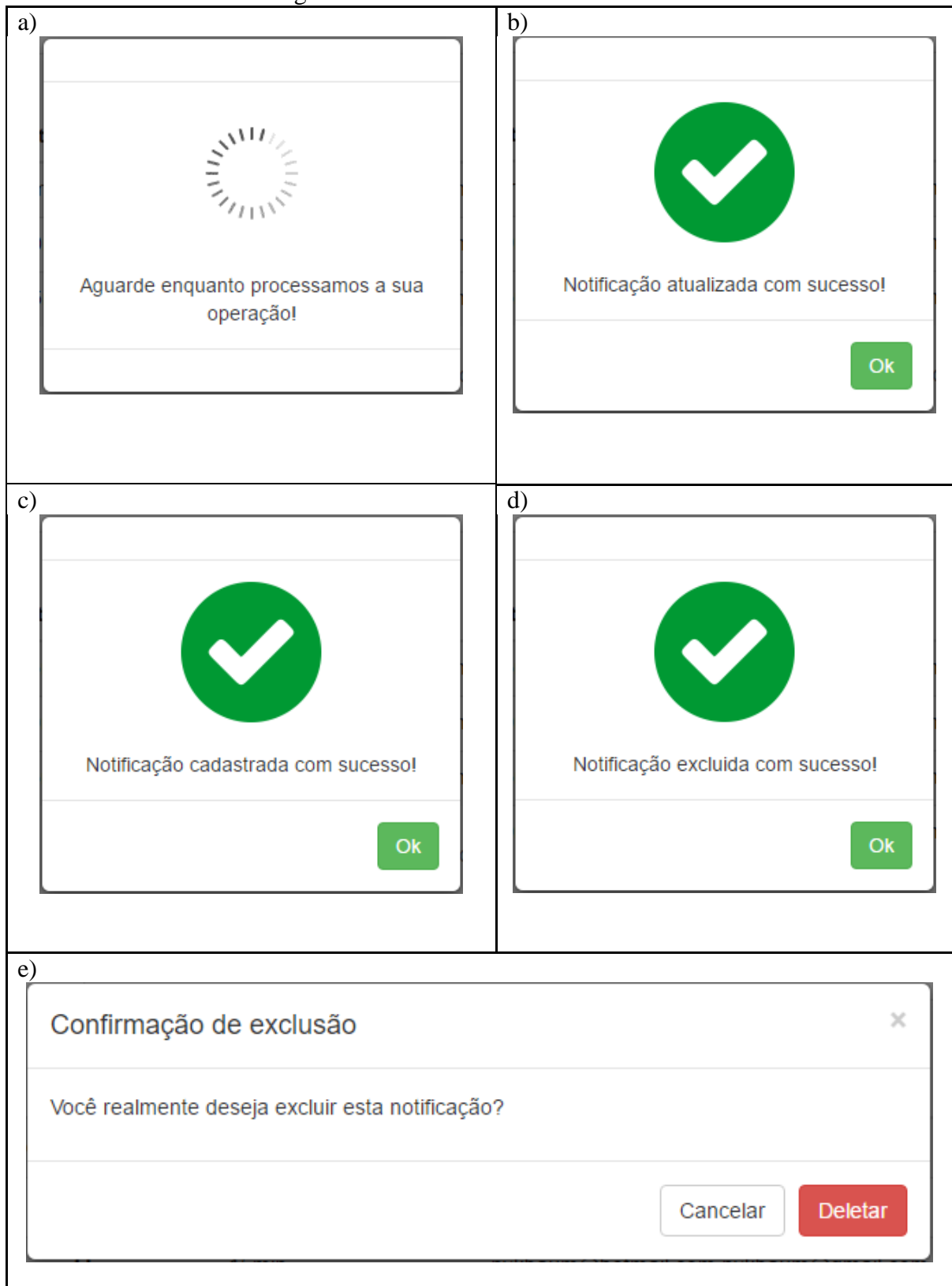
The screenshot displays the 'findCAR' application interface. At the top, there are navigation tabs: 'Tempo Real', 'Notificações', and 'Últimas Localizações'. The 'Últimas Localizações' tab is active, showing a table with two columns: 'Data/hora' and 'Local'. Each row in the table contains a timestamp and a 'Consultar' button. To the right of the table is a satellite map from Google Maps showing an urban area with various streets and landmarks. A red location pin is placed on the map, corresponding to the most recent entry in the table. The map includes labels for streets like 'R. São Paulo', 'Av. Martin Luther', 'R. Des. Oscar Leite', 'R. Antônio da Veiga', 'R. Maranhão', and 'R. Maranhão III', as well as landmarks like 'Fundação Universidade Regional de Blumenau', 'Refrigeração Martendal', 'El Burrito', 'Observatorio Bar e Restaurante', 'SENAI Blumenau', and 'Restaurante & M'. The Google logo and map data links are visible at the bottom of the map.

Data/hora	Local
2016-11-24 17:09:08	Consultar
2016-11-24 16:46:05	Consultar
2016-11-24 16:04:02	Consultar
2016-11-24 16:03:02	Consultar
2016-11-24 16:02:02	Consultar
2016-11-24 16:01:02	Consultar
2016-11-24 16:00:02	Consultar
2016-11-24 15:59:02	Consultar
2016-11-24 15:58:02	Consultar
2016-11-24 15:57:02	Consultar
2016-11-24 15:56:02	Consultar
2016-11-24 15:55:02	Consultar

Fonte: Elaborado pelo autor.

Ao realizar operações de cadastro ou alterações no sistema, são emitidos alertas para informar o usuário sobre o processamento da operação a ser realizada (Figura 27 (a)). Outro tipo de notificação está presente no sistema para informar ao usuário que a operação de cadastro, atualização ou exclusão de algum registro ocorreu de forma correta, pode-se ver os exemplos (b, c, d) na Figura 27. Para realizar de forma definitiva a exclusão de algum registro de notificação o sistema pede através de uma mensagem como na Figura 27 (e) a confirmação desta operação.

Figura 27 – Alertas do sistema ao usuário



Fonte: Elaborado pelo autor.

### 3.4 ANÁLISE DOS RESULTADOS

Os testes foram realizados em alguns percursos realizados pela cidade de Blumenau. Para conexão com a internet foi utilizado um chip com acesso à tecnologia 3G da operadora Vivo. O rastreador foi posicionado no interior do porta luvas de um automóvel Citroën Xsara Picasso. A câmera foi posicionada de forma que capturasse o condutor, geralmente no canto frontal superior direito do veículo.

Em relação a conexão com a internet utilizando o modem 3G pode-se dizer que a conexão é estabelecida de forma muito rápida. O modem 3G consegue reestabelecer a conexão rapidamente mesmo quando ocorre alguma perda significativa de sinal e em poucos segundos a conexão é estabilizada. A taxa de *upload* oscila de forma considerável na região de Blumenau, o que acaba impactando um pouco na resposta do sistema.

A captura da localização durante os testes mostrou-se muito satisfatória, mesmo tendo dificuldade de recuperar a localização atual do veículo em alguns momentos. A localização retornada mostrou-se bastante precisa, sendo que a margem de erro identificada fica em uma faixa de 30 a 50 metros. Em algumas ocasiões o GPS não conseguiu retornar a latitude e longitude para o sistema, porém levando em conta o trajeto percorrido, esta perda de dados não foi muito significativa.

As imagens capturadas pelo sistema através a webcam Logitech C270 possuem uma ótima definição e claridade. Outro ponto de destaque das imagens capturadas foi o foco. Em sua grande maioria as imagens recuperadas com o veículo em trânsito mostraram-se bastante nítidas, sem tremores ou borrões. Em alguns trajetos realizados em período noturno a câmera conseguiu capturar imagens com boa luminosidade mesmo que a taxa de luminosidade no interior do veículo estivesse baixa.

Durante o desenvolvimento do rastreador surgiram alguns problemas que tomaram muito tempo do cronograma do projeto. Um dos problemas foi estabelecer uma conexão com a internet através do modem 3G Huawei E3272. Esta situação consumiu muito tempo que foi gasto realizando pesquisas em fóruns e sites relacionados ao OpenWRT, até identificar a solução. A solução para este problema é a instalação de um pacote chamado `usb-modeswitch`. Este pacote é necessário para os modems que possuem uma tecnologia chamada HiLink. Esta tecnologia emula uma porta *ethernet* quando o modem se conecta a internet. Inicialmente a configuração da conexão via modem 3G estava sendo realizada através de Point-to-Point Protocol over Ethernet (PPPoE) com uso dos pacotes `kmod-ppp` e `kmod-pppoe`. Esta conexão não foi bem-sucedida, pois os modems com tecnologia HiLink

efetuam a conexão de forma autônoma e emulam uma porta *ethernet* no dispositivo onde estão conectados.

Uma grande dificuldade encontrada para o desenvolvimento deste projeto é o pouquíssimo conteúdo encontrado online para consulta e troca de informações sobre o OpenWRT e o WRTnode. Para conseguir superar as dificuldades e sanar as dúvidas que surgiram foram necessárias pesquisas em conteúdos internacionais. Isto se deve também ao objetivo principal para qual o OpenWRT foi desenvolvido inicialmente, este sistema tinha como principal foco permitir um melhor controle sobre o acesso à modems e roteadores.

O rastreador desenvolvido mostrou-se a altura aos trabalhos correlatos selecionados. Em relação a portabilidade do hardware e viabilidade de tornar-se um produto, pode-se destacar superioridade do FINDCAR. No Quadro 6 é exibida uma comparação entre o trabalho desenvolvido e os correlatos.

Quadro 6 – Comparação com correlatos

Trabalhos correlatos	Beszczynski (2008)	Santos Júnior (2009)	Vicenzi (2015)	FINDCAR
Características				
Permite configurar envio de notificações				*
Captura de imagens	*			*
Captura de localização	*			*
Botão de pânico	*			
Acesso/ação sobre o veículo	*	*		
Consulta últimas localizações	*			*
Informações em tempo real		*	*	*
Calcula tempo para o trajeto			*	

Fonte: Elaborado pelo autor.

## 4 CONCLUSÕES

O objetivo de desenvolver um rastreador veicular utilizando OpenWRT foi atendido adequadamente. O rastreador disponibiliza as funcionalidades de forma bem simples para o usuário, além de permiti-lo configurar notificações para receber a localização e uma imagem do condutor do veículo.

O uso da linguagem PHP supriu todas as necessidades do sistema, tanto para a interação com os periféricos utilizados no sistema quanto para o desenvolvimento da parte *back end* do sistema. Além de ser utilizada para realizar a parte *back end* do sistema a linguagem foi utilizada em um script que faz requisições aos periféricos e envia os e-mails de notificação. Poder utilizar somente uma linguagem de programação para desenvolver um sistema com essa variedade de entrada de dados, facilita muito a execução deste processo.

O banco de dados MySQL utilizado para armazenar os dados cadastrados no sistema web, mostrou-se suficientemente estável e ágil para o pequeno número de manipulação de dados efetuada sobre a base. Este banco possui uma certa facilidade de ser manipulado, e o que mais pesou para sua escolha foi o fato de já existir um pacote disponível para ser compilado para o OpenWRT.

As ferramentas utilizadas para o desenvolvimento e para a especificação atenderam bem a todas as necessidades que surgiram. E o uso das funcionalidades exigidas se mostrou fácil de utilizar. Em relação aos hardwares e periféricos utilizados, também pode-se dizer que atenderam as necessidades do rastreador. O WRTnode possui uma limitação de memória, porém em versões mais novas este quesito vem sendo melhorado. O modem 3G e o módulo de GPS mesmo mostrando certa instabilidade em alguns testes por conta do ambiente onde foram testados tiveram um bom resultado como um todo.

Tendo em vista os resultados obtidos com o desenvolvimento deste rastreador, pode-se concluir que o projeto como um todo alcançou os objetivos. Um quesito que merece destaque no projeto é a sua autonomia em relação ao servidor web, pois o sistema é hospedado no próprio dispositivo. Esta característica oferece uma independência ao usuário, não o prendendo a pagamentos de mensalidades que é algo comumente praticados pelas empresas que prestam este tipo de serviço.

### 4.1 EXTENSÕES

Sugestão de extensões para trabalhos futuros:

- a) adicionar o periférico OBDII para conseguir recuperar informações da central

- eletrônica do veículo;
- b) implementar a visualização de rotas percorridas delimitadas por um período de tempo;
  - c) viabilizar o envio de notificações via SMS, para não tornar-se dependente somente da rede 3G;
  - d) desenvolver o rastreamento por triangulação GSM para ter uma segunda opção de rastreamento, caso a localização via GPS tenha dificuldades;
  - e) adicionar um dispositivo de memória adicional, para ampliar o espaço de armazenamento;
  - f) realizar o armazenamento e exibição das imagens capturadas, para que o usuário tenha acesso também a esta informação sem ficar dependente somente das notificações.



## REFERÊNCIAS

- AGENCIA O GLOBO. **Adesão de usuários ao 4G decepciona operadoras**. Curitiba/PR, 2014. Disponível em: < <http://www.gazetadopovo.com.br/economia/adesao-de-usuarios-ao-4g-decepciona-operadoras-9non86o7qsv2ykk4ywbhu4x8u> >. Acesso em: 30 mar. 2016.
- BATISTA, J.. **Índice de recuperação de veículos roubados e furtados aumenta em Joinville**. Joinville, 2015. Disponível em: <<http://www.ndonline.com.br/joinville/noticias/265076-indice-de-recuperacao-de-veiculos-roubados-e-furtados-aumenta-em-joinville.html>>. Acesso em: 29 mar. 2016.
- BESZCZYNSKI, Leandro. **Protótipo de um Sistema de Rastreamento Veicular Baseado no Módulo Telit**. 2008. 67 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Fundação Universidade Regional de Blumenau, Blumenau.
- CAMPBELL, I. V. **Sistema de Posicionamento Global - GPS**. Rio de Janeiro/RJ, 2008. Disponível em: <[http://www.gta.ufrj.br/grad/08\\_1/gps/](http://www.gta.ufrj.br/grad/08_1/gps/)>. Acesso em: 29 mar. 2016.
- DIÁRIO DO VALE. **Santa Catarina registra aumento de roubos e furtos**. Joaçaba, 2015. Disponível em: <<http://www.diariodovalesc.com.br/noticias.php?id=5862> >. Acesso em: 30 mar. 2016.
- FREITAS, E. **Orientação e localização**. [S.l.], 2016 Disponível em: <<http://mundoeducacao.bol.uol.com.br/geografia/orientacao-localizacao.htm> >. Acesso em: 28 mar. 2016.
- GARRIDO, Adriana C. O. **Fatores Sociais de Criminalidade**. Paracatu/MG, 2007. Disponível em: <<http://www.atenas.edu.br/Faculdade/arquivos/NucleoIniciacaoCiencia/REVISTAS/REVISTA2007/5.pdf> >. Acesso em: 28 mar. 2016.
- GPS CENTER. **Conceito de GPS**. São Paulo/SP, 2007. Disponível em: <<http://www.gpscenter.com.br/index64.html> >. Acesso em: 29 mar. 2016.
- HAMMERSCHMIDT, R. **O que é 3G?**. [S.l.],2008. Disponível em: <<http://www.tecmundo.com.br/celular/226-o-que-e-3g-.htm>>. Acesso em: 19 mar. 2016.
- INFOESCOLA. **GPS (Sistema de Posicionamento Global)**. Joaçaba, 2015. Disponível em: <<http://www.infoescola.com/cartografia/gps-sistema-de-posicionamento-global/>>. Acesso em: 28 mar. 2016.
- NOTÍCIAS AUTOMOTIVAS. **OBD II: Conectividade ajuda motorista a saber mais sobre o carro**. [SI], 2014. Disponível em: <<http://www.noticiasautomotivas.com.br/obd-ii-conectividade-ajuda-motorista-a-saber-mais-sobre-o-carro/>> Acesso em: 16 mai.2016.
- OPENWRT WIRELESS FREEDOM. **Abordagem Básica do OpenWrt**. [S.l.], 2015. Disponível em: < <https://wiki.openwrt.org/pt-br/doc/howto/user.beginner>>. Acesso em: 28 mar. 2016.
- PHP. **O que o PHP pode fazer?**. [S.l.], 2016. Disponível em: <[https://secure.php.net/manual/pt\\_BR/intro-whatcando.php](https://secure.php.net/manual/pt_BR/intro-whatcando.php)>. Acesso em: 29 mar. 2016.
- SANTOS JÚNIOR, Álvaro Santana dos. **Sistema de Monitoramento Eletrônico Automotivo**. 2009. 94 f. Trabalho de Conclusão de Curso (Engenharia de Computação) – Centro Universitário de Brasília, Faculdade de Tecnologia e Ciências Sociais Aplicadas, Brasília.

SOARES, K. **Modem 3G**: um guia de compra com dicas para ter a melhor opção. [S.l.], 2013. Disponível em: < <http://www.techtudo.com.br/noticias/noticia/2013/12/modem-3g-um-guia-de-compra-com-dicas-para-ter-melhor-opcao.html>>. Acesso em: 29 mar. 2016.

TELECO. **3G**: Tecnologias de Celular. São José dos Campos/SP, 2014. Disponível em: <[http://www.teleco.com.br/3g\\_tecnologia.asp](http://www.teleco.com.br/3g_tecnologia.asp)>. Acesso em: 30 mar. 2016.

TOTAL CAR. **How to Install ELM327 Bluetooth OBD Scanner on Android Phone/Tablet**. Australia, 2013. Disponível em: <<http://www.totalcardiagnostics.com/support/Knowledgebase/Article/View/20/0/how-to-install-elm327-bluetooth-obd-scanner-on-android-phonetablet>>. Acesso em: 28 mar. 2016.

VICENZI, Alexandre. **BUSTRACKER**: Sistema de Rastreamento para Transporte Coletivo. 2015. 62 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Fundação Universidade Regional de Blumenau, Blumenau.

WEAVED. **Remote Manage Networked Devices Anywhere**. Palo Alto/CA, 2016. Disponível em: <<https://www.weaved.com/>>. Acesso em: 30 mar. 2016.

## APÊNDICE A – Relação de pacotes instalados no OpenWRT

Neste apêndice estão sendo exibidos os pacotes instalados no OpenWRT para possibilitar o desenvolvimento do rastreador. Esta relação de pacotes foi dividida em X quadros, para facilitar o entendimento do objetivo para o qual os pacotes foram instalados. Vários pacotes e programas já vem instalados por default no OpenWRT para facilitar o acesso e configuração. Os pacotes que já vieram instalados com o OpenWRT não serão relacionados nestas tabelas.

No Quadro 7 estão relacionados os pacotes que foram instalados para facilitar e auxiliar no desenvolvimento, tanto para acesso, para transmissão de arquivos e testes de acesso aos periféricos.

Quadro 7 – Pacotes para auxílio ao desenvolvimento

kmod-usb-ohci
kmod-usb-serial
kmod-usb-serial-ftdi
kmod-usb-uhci
kmod-usb2
kmod-usb-core
minicom
picocom
openssh-sftp-server
usbutils

Fonte: Elaborado pelo autor

No Quadro 8 estão relacionados os pacotes que foram instalados para permitir a utilização de todos os periféricos do rastreador.

Quadro 8 – Pacotes para periféricos

kmod-video-core
kmod-video-uv
kmod-video-videobuf2
kmod-usb-serial-pl2303
kmod-usb-net-cdc-ether
kmod-usb-net
libjpeg
mjpg-streamer
usb-modeswitch

Fonte: Elaborado pelo autor.

No Quadro 9 estão relacionados os pacotes necessários para utilizar o banco de dados MySQL e tornar do WRTnode o servidor web da aplicação. Neste quadro também foi adicionado o pacote `mailsend` que foi utilizado para o envio de e-mails.

Quadro 9 – Pacotes para base de dados e servidor web

php5
php5-cgi
php5-cli
php5-fastcgi
php5-fpm
php5-mod-calendar
php5-mod-ctype
php5-mod-curl
php5-mod-dom
php5-mod-exif
php5-mod-fileinfo
php5-mod-gd
php5-mod-gettext
php5-mod-gmp
php5-mod-hash
php5-mod-iconv
php5-mod-json
php5-mod-ldap
php5-mod-mbstring
php5-mod-mcrypt
php5-mod-mysql
php5-mod-mysqli
php5-mod-opcache
php5-mod-openssl
php5-mod-pdo
php5-mod-pdo-mysql
php5-mod-session
php5-mod-shmop
php5-mod-simplexml
php5-mod-sockets
php5-mod-sysvmsg
php5-mod-sysvsem
php5-mod-sysvshm
php5-mod-tokenizer
php5-mod-xml
php5-mod-xmlreader
php5-mod-xmlwriter
php5-mod-zip
php5-pecl-dio
uhttpd
uhttpd-mod-ubus
mysql-server
mailsend

Fonte: Elaborado pelo autor.

Por último no Quadro 10 seguem os demais pacotes instalados no OpenWRT, porém não necessariamente utilizados no projeto do rastreador.

Quadro 10 – Demais pacotes instalados

base-files
busybox
ca-certificates
dnsmasq
dropbear
firewall
fstools
hostapd-common
ip6tables
iptables
iwinfo
jshn
jsonfilter
kernel
kmod-dma-buf
kmod-gpio-button-hotplug
kmod-input-core
kmod-ip6tables
kmod-ipt-contrack
kmod-ipt-core
kmod-ipt-nat
kmod-ipv6
kmod-leds-gpio
kmod-ledtrig-usbdev
kmod-lib-crc-ccitt
kmod-mii
kmod-nf-contrack
kmod-nf-contrack6
kmod-nf-ipt
kmod-nf-ipt6
kmod-nf-nat
kmod-nf-nathelper
kmod-nls-base
kmod-ppp
kmod-pppoe
kmod-pppox
kmod-slhc
libblobmsg-json
libc
libcurl
libgcc
libgmp
libintl-full
libip4tc
libip6tc
libiwinfo
libiwinfo-lua

libjson-c
libjson-script
libltdl
liblua
libmcrypt
libmysqlclient
libncurses
libncursesw
libnl-tiny
libopenldap
libopenssl
libpcre
libpng
libpolarssl
libpq
libpthread
libreadline
librt
libsasl2
libsqlite3
libubox
libubus
libubus-lua
libuci
libuci-lua
libusb-1.0
libxml2
libxtables
logd
lua
luci
luci-app-firewall
luci-base
luci-lib-ip
luci-lib-nixio
luci-mod-admin-full
luci-proto-ipv6
luci-proto-ppp
luci-theme-bootstrap
maccalc
mtid
netifd
odhcp6c
odhcpd
openssl-util
opkg
ppp
ppp-mod-pppoe

procd
ralink-wifi
rpcd
shadow
shadow-chage
shadow-common
shadow-groupadd
shadow-groupdel
shadow-groupmod
shadow-groups
shadow-passwd
shadow-su
shadow-useradd
shadow-userdel
shadow-usermod
shadow-utils
swconfig
terminfo
ubox
ubus
ubusd
uci
uclibcxx
wireless-tools
wpad-mini
zlib
zoneinfo-core
zoneinfo-southamerica

Fonte: Elaborado pelo autor.

## APÊNDICE B – Custo parcial e total do projeto

Neste apêndice está sendo apresentado no Quadro 11 o custo por hardware/periférico e o custo total do projeto. O WRTnode e o Shield WRTnode foram comprados através do site aliexpress.com, os demais periféricos e peças foram comprados no mercadolivre.com.br e na proesi.com.br.

Quadro 11 – Custo do projeto

Peça	Valor importação s/ custo do transporte	Valor em Reais
WRTnode	US\$ 32,49	R\$ 117,69
Shield WRTnode	US\$ 10,20	R\$ 52,01
USB/Serial PL2303	-	R\$ 10,00
USB/Serial CH340G	-	R\$ 13,00
Modem 3g Huawei E3272	-	R\$ 149,99
Módulo de GPS GY-NEO6MV2	-	R\$ 74,95
Webcam Logitech C270	-	R\$ 79,54
Barra pinos	-	R\$ 0,67
Fios jumper 10 unidades de 20cm	-	R\$ 11,29
Custo total do projeto		R\$ 509,14

Fonte: Elaborado pelo autor.