

Universidade Federal do Rio Grande do Sul
Escola de Engenharia
Graduação em Engenharia de Controle e Automação

William Malacarne Boff

**SISTEMA DE ALARME CONECTADO VIA REDES MÓVEIS PARA
MONITORAMENTO E SEGURANÇA DE VEÍCULOS AUTOMOTIVOS ATRAVÉS DE
APLICATIVO DE TELEFONE CELULAR**

Porto Alegre
2017

William Malacarne Boff

**SISTEMA DE ALARME CONECTADO VIA REDES MÓVEIS PARA
MONITORAMENTO E SEGURANÇA DE VEÍCULOS AUTOMOTIVOS ATRAVÉS DE
APLICATIVO DE TELEFONE CELULAR**

Trabalho de Conclusão de Curso de Graduação apresentado ao Curso de Engenharia de Controle e Automação da Universidade Federal do Rio Grande do Sul, como requisito parcial e obrigatório para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Carlos Eduardo Pereira

Porto Alegre
2017

CIP - Catalogação na Publicação

Boff, William Malacarne

Sistema de Alarme Conectado via Redes Móveis para Monitoramento e Segurança de Veículos Automotivos através de Aplicativo de Telefone Celular / William Malacarne Boff. -- 2017.

54 f.

Orientador: Carlos Eduardo Pereira.

Trabalho de conclusão de curso (Graduação) -- Universidade Federal do Rio Grande do Sul, Escola de Engenharia, Curso de Engenharia de Controle e Automação, Porto Alegre, BR-RS, 2017.

1. Arduino. 2. Aplicativo. 3. Alarme Automotivo. 4. Android. 5. Redes Móveis. I. Pereira, Carlos Eduardo, orient. II. Título.

William Malacarne Boff

**SISTEMA DE ALARME CONECTADO VIA REDES MÓVEIS PARA
MONITORAMENTO E SEGURANÇA DE VEÍCULOS AUTOMOTIVOS ATRAVÉS DE
APLICATIVO DE TELEFONE CELULAR**

Este trabalho foi julgado adequado para obtenção de créditos da disciplina Trabalho e Conclusão de Curso em Engenharia de Controle e Automação e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

.

Orientador:
Prof. Dr. Carlos Eduardo Pereira, UFRGS
Doutor pela Universidade de Stuttgart – Alemanha.

Banca Examinadora:

Marcelo Götz
Doutor pela Universidade de Paderborn - Alemanha

Renato Ventura Bayan Henriques
Doutor pela Universidade de Minas Gerais - Brasil

Alexandre da Silva Carissimi
Doutor pelo *Institut National Polytechnique de Grenoble* (INPG) - França

Porto Alegre
Dezembro de 2017

Agradecimentos

Agradeço, primeiramente, à minha família, meu pai Uilson, minha mãe Dilamar, meu irmão Lucas e minhas avós Sueli e Elzira, por todo o auxílio e atenção que disponibilizaram a mim, pois estes foram fundamentais para que chegasse até este momento. Com vocês esta trajetória se tornou ainda mais gratificante. Amo todos vocês!

Aos meus colegas de graduação, que durante os anos de vivência dentro da universidade compartilharam comigo histórias e opiniões, que permitiram-me aprimorar conhecimentos específicos de minha formação, mas também meus conhecimentos sobre a sociedade, pois a vivência dentro do campus mostrou muitas das maneiras de se viver e ser. Pude vivenciar e ter contato com novas culturas e pensamentos, que abriram minha mente e evoluíram minha maneira de pensar.

Aos meus vizinhos e amigos da casa do estudante da UFRGS, que me mostraram como é viver em uma casa cheia de vida e de diferentes ideologias.

Aos meus professores da graduação, que transmitiram seu conhecimento e acompanharam minha evolução durante os anos. E, não obstante, ao meu orientador Carlos Eduardo Pereira, por ter disponibilizado sua atenção e conhecimento para instruir-me da forma clara e segura, que permitiu-me alcançar o objetivo de concluir este trabalho. Devo meus agradecimentos também ao Me. Eng. Carlos Solon, pelo auxílio dado durante o desenvolvimento do projeto.

“Se fosse fácil achar o caminho das pedras, tantas pedras no caminho não seria ruim”.

Humberto Gessinger

Resumo

Este trabalho visa conjuntamente desenvolver *hardware* e aplicativo de telefone celular para a implementação de um sistema de alarme que possibilita o monitoramento de veículos através de dados enviados via redes móveis ao telefone do usuário. O *hardware* foi desenvolvido utilizando um alarme automotivo comercial modelo K-150, da Kostal®, adaptado à plataforma de prototipagem eletrônica Arduino®, que possui um módulo eletrônico acoplado permitindo a conexão em redes móveis, tanto por SMS utilizando a tecnologia GSM, quanto por protocolo HTTP, utilizando a tecnologia GPRS. Dessa forma, informações consideradas para este trabalho importantes para segurança do veículo, como: alarme disparado ou ativado, portas, capô ou porta-malas abertos, rádio ligado, faróis ligados, luz de teto ou ignição ligada, podem ser monitoradas pelo microcontrolador do Arduino® e enviadas para uma plataforma de monitoramento. Nesse caso, trata-se de um aplicativo desenvolvido e instalado no aparelho de telefone celular do usuário, que possua o sistema operacional Android®. O aplicativo, além de permitir, monitoramento das informações, também permite que o usuário realize a ação de ativar o alarme a distância, através de um comando dado por SMS, caso a ativação tenha sido esquecida no momento de estacionar o carro. O projeto foi implementado em bancada de teste utilizando *protoboard* e fonte de tensão externa para simular o comportamento do carro que será monitorado. O aplicativo de telefone foi desenvolvido utilizando o ambiente de desenvolvimento integrado (IDE) Android Studio, disponibilizado gratuitamente pela Google®. Para o sistema completo, fez-se necessário também a utilização de um servidor *web*, responsável pela manipulação e hospedagem do banco de dados para realizar as funcionalidades por Internet.

O sistema implementado em *protoboard* simulou satisfatoriamente o veículo e seus mecanismos monitorados. As funcionalidades por redes móveis mostraram-se estáveis nas maiorias dos testes, com exceções para algumas regiões onde o sinal da operadora telefônica utilizada não era de boa qualidade. De qualquer modo, o projeto mostrou resultados eficientes para possibilitar ao usuário monitoramento da segurança de seu veículo, especialmente com a funcionalidade oferecida pelo aplicativo desenvolvido para *smartphone*.

Palavras-chave: GSM, alarme automotivo, redes móveis, Android, Arduino, aplicativo.

Abstract

This work aims to develop jointly hardware and mobile app for the implementation of an alarm system that enables the monitoring of vehicles through data sent via mobile networks to the user's phone. The hardware was developed using a Kostal® K-150 commercial automotive alarm, adapted to the Arduino® electronic prototyping platform, which has an electronic module coupled allowing it to connect in mobile networks, either by SMS, using GSM technology, or per HTTP protocol, using GPRS technology.. In this way, information considered important for the safety of the vehicle, such as: alarm triggered or activated, doors, hood or trunk opened, radio on, headlamps on, internal light or ignition on, can be monitored by the Arduino® micro-controller and sent to a monitoring platform. In this case, it is an application developed and installed on the user's phone device, which has the Android® operating system. The system, besides allowing information monitoring, also allows the user to perform the action of activating the remote alarm, through a command given by SMS, if the activation was forgotten when the car was parked. The project was implemented in test bench using protoboard and external voltage source to simulate the behavior of the car that will be monitored. The mobile app was developed using the Android Studio integrated development environment (IDE), made available for free by Google®. For the complete system, it was also necessary to use a web server, responsible for the manipulation and hosting of the database to perform as internet functionalities.

The protoboard system simulated the vehicle and its monitored mechanisms satisfactorily. Mobile network functionalities have proven stable in most tests, with exceptions for some regions where the telephone operator's signal is not good quality. Either way, the project showed efficient results to enable the user to monitor the safety of their vehicle, especially with the functionality provided by the application developed for mobile.

Keywords: GSM, automotive alarm, mobile networks, Android, Arduino, mobile app.

LISTA DE FIGURAS

FIGURA 1 - TAXA DE FURTOS (A) E ROUBOS (B) DE VEÍCULOS OCORRIDOS EM 2014 NO BRASIL.....	17
FIGURA 2 - EVOLUÇÃO DAS TAXAS DE FURTO (A) E ROUBO (B) DE VEÍCULOS ENTRE 2011 E 2014 NO BRASIL.....	18
FIGURA 3 - ARDUINO MEGA.	20
FIGURA 4 - ESTRUTURA DE UM SISTEMA CELULAR BÁSICO.....	22
FIGURA 5 - ACESSOS DE TELEFONIA REGISTRADOS NO BRASIL.....	25
FIGURA 6 - APLICATIVOS DISPONÍVEIS NA GOOGLE PLAY.	26
FIGURA 7 - ARQUITETURA SIMPLIFICADA DO SISTEMA.....	27
FIGURA 8 - CONEXÕES COM O ARDUINO.	31
FIGURA 9 - SHIELD GSM ICOMSAT V1.1.....	32
FIGURA 10 - CONEXÃO PARA COMUNICAÇÃO SERIAL ENTRE ARDUINO E SHIELD GSM/GPRS.	33
FIGURA 11 - FLUXOGRAMA DE FUNCIONAMENTO DA CENTRAL.....	35
FIGURA 12 - AQUISIÇÃO E COMPARAÇÃO NO LAÇO DE DETECÇÃO.....	36
FIGURA 13 - VETOR DE BYTES PARA ENVIO DA INFORMAÇÃO.....	37
FIGURA 14 - ÁRVORE DE PROJETO ANDROID STUDIO.	40
FIGURA 15 - FLUXOGRAMA DE FUNCIONAMENTO DO APLICATIVO.....	42
FIGURA 16 -FUNCIONALIDADES DO SISTEMA NO SMARTPHONE.	44
FIGURA B.1 - DIAGRAMA DE INSTALAÇÃO ALARME KOSTAL K-150.....	50
FIGURA B.2 - ESQUEMÁTICO ELÉTRICO PRINCIPAL DA PROTOBOARD.....	51
FIGURA B.3 - ESQUEMÁTICO ELÉTRICO DE VISUALIZAÇÃO DA PROTOBOARD.....	51
FIGURA B.4 - MONTAGEM PROTOBOARD DO MODELO DA PLANTA	52
FIGURA B.5 - MONTAGEM CENTRAL DE MONITORAMENTO E COMUNICAÇÃO	53

LISTA DE TABELAS

TABELA 1 - PADRÕES IEEE - TECNOLOGIAS DE REDE <i>WIRELESS</i>	21
TABELA 2 - ALARMES COMERCIAIS ANALISADOS.....	28
TABELA 3 - VALORES RESISTORES DOS DIVISORES DE TENSÃO	30
TABELA C.1 - DEFINIÇÕES DE NOMECLATURAS E NÍVEIS DE TENSÃO MEDIDOS.....	54
TABELA C.2 - TABELA UTILIZADA NO BANCO DE DADOS DO SERVIDOR.....	54

LISTA DE SÍMBOLOS

®	Marca registrada
%	Por cento
Ω	Ohms

LISTA DE ABREVIATURAS E SIGLAS

1G	Primeira Geração
2G	Segunda Geração
2,5G	Segunda e meia Geração
3G	Terceira Geração
3GPP	<i>3rd Generation Partnership Project</i>
4G	Quarta Geração
A	Ampere
AMPS	<i>Advanced Mobile Phone Service</i>
BSC	<i>Base Station Controller</i>
CAN	<i>Controller Area Network</i>
CCC	Central de Controle e Comutação
CDMA	<i>Code Division Multiple Access</i>
DB	<i>Database</i>
EDGE	<i>Enhanced Data rates for Global Evolution</i>
ERB	Estação Rádio Base
EVDO	<i>Evolution Data Only</i>
FDMA	<i>Frequency Division Multiple Access</i>
FTP	<i>File Transfer Protocol</i>
GHz	Gigahertz
GPRS	<i>General Packet Radio Service</i>
GSM	<i>Global System for Mobile Communications</i>
Hz	Hertz
HTTP	<i>HyperText Transfer Protocol</i>
HTML	<i>HyperText Markup Language</i>
IDE	<i>Integrated development environment</i>
IMEI	<i>International Mobile Equipment Identity</i>
IP	<i>Internet Protocol</i>
IoT	<i>Internet of Things</i>
IO	<i>Input/Output</i>
JSON	<i>JavaScript Object Notation</i>
kΩ	quilo ohms
Kbps	quilobit por segundo
LED	<i>Light Emitter Diode</i>
LTE	<i>Long Term Evolution</i>
MHz	Megahertz
MQTT	<i>Message Queue Telemetry Transport</i>
KB	quilobytes
km	quilômetros
m	metros
Mbps	Megabit por segundo
MvSQL	<i>Database Management System for Structured Query Language</i>
OBD	<i>On-board Diagnostics</i>
OBD II	<i>On-board Diagnostics II</i>

OHA	<i>Open Handset Alliance</i>
PHP	<i>Hypertext Preprocessor</i>
RAM	<i>Random Access Memory</i>
RF	Rádio frequência
RMS	<i>Root Mean Square</i> (Valor eficaz)
RPTC	Rede de telefonia pública comutada
RS	Rio Grande do Sul
R\$	Real – moeda corrente oficial da República Federativa do Brasil
SDK	<i>Software Development Kit</i>
SIM CARD	Subscriber Identity Module Card
SINESP	Sistema Nacional de Informações de Segurança Pública
SMS	<i>Short Message Service</i>
SQL	<i>Structured Query Language</i>
SSP	Secretaria de Segurança Pública
TDMA	<i>Time Division Multiple Access</i>
USB	<i>Universal Serial Bus</i>
URL	<i>Uniform Resource Locator</i>
V	Volts
VCC	Tensão corrente contínua
WAP	<i>Wireless Application Protocol</i>
WCDMA	<i>Wideband Code Division Multiple Access</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access</i>

Sumário

1	Introdução	16
1.1	Objetivos do Trabalho.....	19
1.2	Organização.....	19
2	Revisão Bibliográfica.....	20
2.1	Arduino.....	20
2.2	Redes Móveis	21
2.2.1	Sistema Móvel Celular.....	21
2.2.2	Evolução das Tecnologias de Telefonia Celular.....	23
2.3	Sistema Operacional Android.....	26
3	Materiais e Métodos.....	27
3.1	Modelo da Planta.....	28
3.1.1	Escolha do alarme.....	28
3.1.2	Montagem da planta simulando o carro.....	29
3.2	Central de Monitoramento e Comunicação	30
3.2.1	Arquitetura.....	31
3.2.2	Circuito do comando remoto.....	34
3.2.3	Montagem da central de monitoramento e comunicação	34
3.2.4	Funcionamento geral da central	34
3.2.5	Leitura das variáveis e chamada das funções.....	36
3.3	Servidor de Dados.....	37
3.3.1	Banco de Dados.....	37
3.3.2	Arquivos PHP.....	38
3.4	Dispositivo móvel.....	39
3.4.1	Definições de projeto para o aplicativo	41
3.2.4	Funcionamento geral do aplicativo	41
4	Resultados e Discussão	43
4.1	Comportamento da planta.....	43
4.2	Comportamento da central.....	43
4.3	Comportamento do aplicativo	44

5 Considerações Finais e Trabalhos Futuros	45
Referências.....	46
APÊNDICE A - CÓDIGOS.....	50
APÊNDICE B - IMAGENS	50
APÊNDICE C - TABELAS.....	54

1 Introdução

Já não mais tratada com novidade no campo jurídico, a violência tem provocado nos últimos anos um profundo sentimento de instabilidade e de insegurança na sociedade (ROSA, 2010). Diariamente pode-se acompanhar pelos canais de comunicação as notícias, que não são poucas, expondo a criminalidade e os sérios problemas de violência que ocorrem não apenas no Brasil, mas em todo o mundo. Não obstante, a sociedade vivencia constantes frustrações no tocante à fragilidade da vida pública e social em relação a violência (ABREU e LOURENÇO, 2010).

Antigamente, tinha-se uma visão de que a violência ocorria apenas no “mundo da pobreza”, isto é, nas periferias e nas favelas (Rosa, 2010). Nessa perspectiva, parecia haver um entrelaçamento do entendimento da violência como um fenômeno de ordem econômica, com a periculosidade identificada nos jovens de periferia, fato que acabava estigmatizando as pessoas que residiam nesses locais como perigosas (Pain & Smith, 2008; Dubet, 2010). Entretanto, atualmente, sabe-se que a violência está presente independentemente da situação econômica e localização de território e afeta a população de modo desigual, gerando riscos diferenciados em função de gênero, raça/cor e idade (Souza, 2006). Diversos autores apontam que os desvios cometidos pela elite, a corrupção, a segregação social, a desigualdade social e o preconceito são os principais fatores para o crescente aumento da criminalidade e violência urbana (Caldeira, 2000; Bauman, 2003; Vaz, 2005).

Estudos realizados pelo *Institute for Global and Peace* (2017) apontam o Brasil como o 55º país mais inseguro do mundo. Tal insegurança também gera prejuízos ao país. O estudo também informa que o custo da violência no país alcançou 12,6% do PIB em 2016. Cerqueira et al., 2007, realizaram estudo semelhante e estimaram que a violência no Brasil em 2004 custou aos cofres públicos 92,2 bilhões de reais, equivalente a 5,09 % do PIB. Orçamento anual que, de acordo com os estudos, vem crescendo a cada ano.

Para a visão culturalista, o Estado deve procurar promover o crescimento econômico, a geração de empregos e a distribuição de renda, combatendo a pobreza e a desigualdade. O problema desse enfoque é que diversos estudos já mostraram que a redução de desemprego, crescimento econômico e aumento da escolaridade, por si só, não têm efeito sobre as taxas de criminalidade nas grandes cidades (BECKER, 1977; SAPORI e WANDERLEY, 2001).

Englobado nos quadros de violência, está o roubo de veículos, que, vem aumentando a cada ano, acompanhando o crescimento relatados dos índices de criminalidade. Segundo Rosa (2010), as unidades federativas brasileiras buscam todo ano aprimorar as estratégias para a diminuição desse quadro. Outros fatores que buscam essa diminuição são as novas tecnologias e a evolução dos aparatos antifurto, como chaves codificadas, alarmes sofisticados e bloqueadores. O avanço da tecnologia *Internet of Things* (IoT), a conectividade cada vez maior de dispositivos, a globalização dos *smartphones* e a evolução das tecnologias embarcadas permite melhores avanços em soluções de segurança. Mesmo assim, a situação é alarmante. Dados fornecidos pelo SINESP mostram que entre os anos de 2011 e 2014, as ocorrências de roubo de veículos no país aumentaram 50,4%, enquanto as ocorrências de furto, no mesmo período, aumentaram 33,8%. Estudos mais detalhados também são fornecidos pelo

SINESP, onde é possível verificar as ocorrências registradas em cada estado. Para permitir a comparação das diferentes frotas de veículos entre os estados, o SINESP utiliza a taxa por cem mil veículos, que mantém os indicadores proporcionais ao tamanho da frota de cada estado. Sendo assim, para cada estado, divide-se o número de ocorrências pelo tamanho da frota e posteriormente multiplica-se por 100.000. As figuras 1 e 2 mostram mais detalhadamente estes dados.

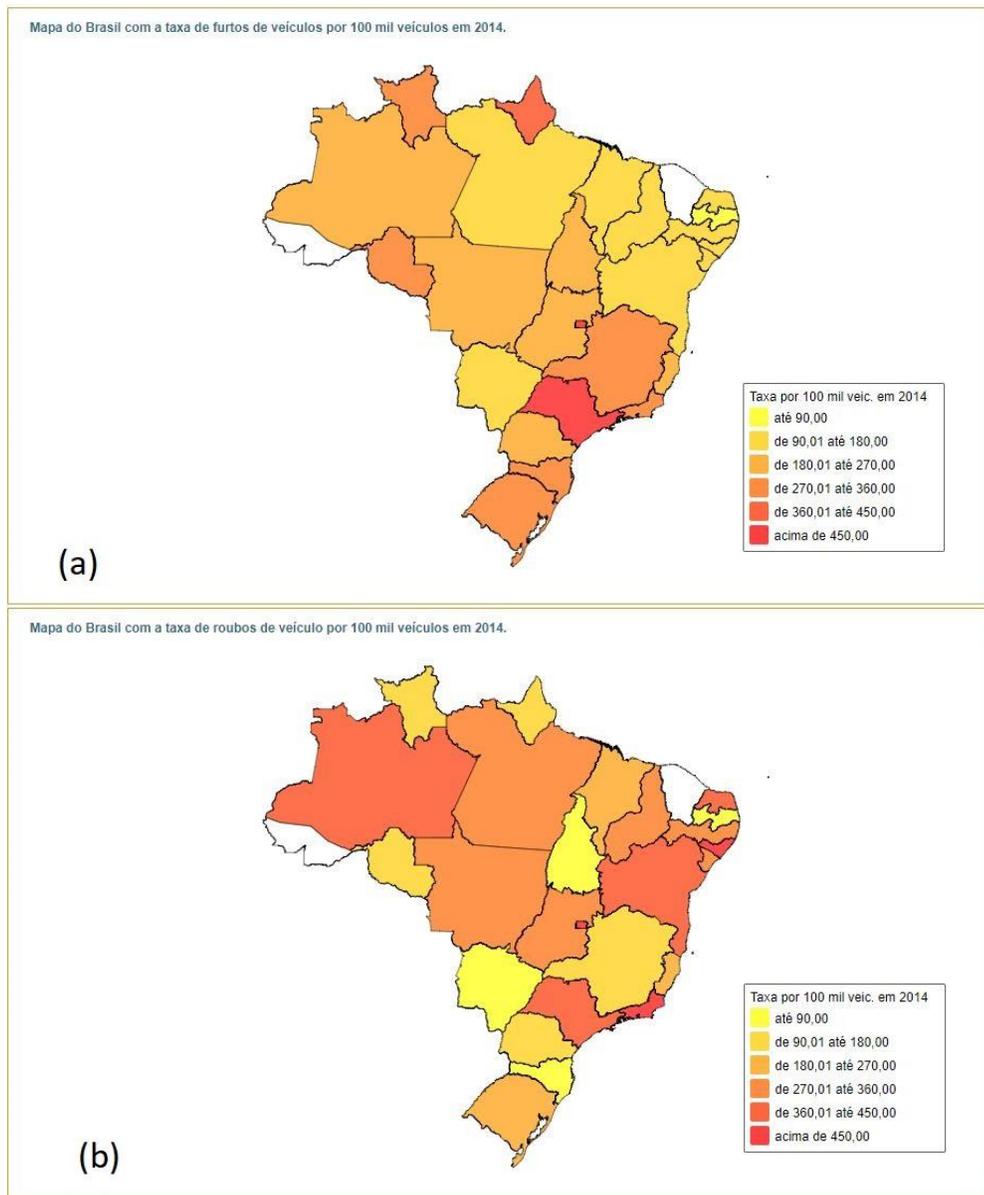


Figura 1 - Taxa de furtos (a) e roubos (b) de veículos ocorridos em 2014 no Brasil.

Fonte: SINESP, 2016.

Analisando-se os mapas ilustrativos do SINESP, nota-se que este tipo de atividade criminosa ocorre em todo o país, com destaque para os estados mais populosos. Em 2014, o estado de São Paulo foi o que mais registrou ocorrências de furto, enquanto o estado do Rio de Janeiro registrou a maior ocorrência de roubos.

A evolução das ocorrências no país também pode ser verificada pela Figura 2, disponibilizadas pelo SINESP.



Figura 2 - Evolução das taxas de furto (a) e roubo (b) de veículos entre 2011 e 2014 no Brasil.
Fonte: SINESP, 2016.

De acordo com a Figura 2, a ocorrência de furtos e roubos no país vem aumentando todos os anos, com um destaque de crescimento entre os anos de 2012 e 2013. No Rio Grande do Sul, a situação também é alarmante. Dados fornecidos pela Secretaria de Segurança Pública do RS (SSP-RS) mostram que, em 2011, foram registrados 14.650 furtos e 10.967 roubos relacionados a veículos, enquanto 2016 registrou 19.542 furtos e 17.640 roubos. Um aumento de 33,4 % e 60,8 %, respectivamente.

Conforme mencionado anteriormente, novas tecnologias e a evolução dos aparatos antifurto buscam melhorar estes indicadores, porém os resultados e indicadores ainda não são satisfatórios, conforme demonstrado. A motivação do desenvolvimento do presente trabalho busca cooperar com as estratégias de segurança pública e trazer aos usuários de veículos, maior comodidade e tranquilidade quando o seu bem está fora da sua linha de visão. Deste modo, o presente trabalho visa desenvolver um sistema de alarme que permita maior segurança para veículos e seus usuários, permitindo-os acesso à comunicação para troca de informações ou comandos à distância, através de aplicativo para *smartphone*.

1.1 Objetivos do Trabalho

Os objetivos do trabalho são:

- Simular o comportamento de um veículo que possui alarme utilizando *protoboard* e componentes eletrônicos permitindo assim acoplar o modelo simulado do carro a um sistema de aquisição e envio de dados;
- Conectar o sistema de monitoramento implementado no Arduino® a uma rede que permita registro e acesso aos dados obtidos;
- Desenvolver o código aplicado ao Arduino® para realizar as funcionalidades envio e recebimento de dados via redes móveis;
- Monitorar as seguintes informações do veículo:
 - *Status* do alarme;
 - Faróis;
 - Ignição do veículo;
 - Luz de teto do veículo;
 - Abertura e fechamento de portas;
 - Abertura e fechamento de capô e porta-malas;
 - Rádio;
- Desenvolver o aplicativo para sistema operacional Android® para interface, disponibilizando ao usuário todas as informações de segurança do carro e permitindo-o ativar o alarme remotamente;
- Avaliar se a implementação do sistema aumenta e permite maior tranquilidade e segurança ao usuário em relação ao seu veículo.

1.2 Organização

O presente trabalho foi organizado da seguinte forma:

No Capítulo 2 são introduzidas as referências teóricas importantes utilizadas para o desenvolvimento do projeto. No Capítulo 3 é apresentada toda a metodologia utilizada para desenvolver o projeto e obter-se o resultado apresentado no Capítulo 4. No Capítulo 4, além de se discutir os resultados, são apresentadas algumas soluções já existentes no mercado que se assemelham com o projeto. Por fim, no Capítulo 5 são apresentadas as considerações finais e possíveis trabalhos futuros para o projeto.

2 Revisão Bibliográfica

Este capítulo descreve a teoria básica utilizada para a estruturação do presente trabalho.

2.1 Arduino

Em termos práticos, um Arduino é um pequeno computador que se pode programar para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele. O Arduino é o que se chama de plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de hardware e software (McRoberts, 2011).

A maior vantagem do Arduino sobre outras plataformas de desenvolvimento de microcontroladores é a facilidade de sua utilização; pessoas que não são da área técnica podem, rapidamente, aprender o básico e criar seus próprios projetos em um intervalo de tempo relativamente curto. Artistas, mais especificamente, parecem considerá-lo a forma perfeita de criar obras de arte interativas rapidamente, e sem conhecimento especializado em eletrônica. Há uma grande comunidade de pessoas utilizando Arduino, compartilhando seus códigos e diagramas de circuito para que outros os copiem e modifiquem. A maioria dessa comunidade também está muito disposta a auxiliar outros desenvolvedores (McRoberts, 2011).

Segundo Monk (2017), uma outra razão para o sucesso do Arduino é que ele não se limita a placas com microcontrolador. Há um enorme número de placas acessórias (denominadas *shields*) compatíveis com o Arduino. Essas placas são encaixadas diretamente por cima da placa do Arduino. É útil também saber que há diversos modelos de placas de Arduino. Em seu livro, Monk (2017) descreve alguns dos modelos conhecidos da família Arduino®, como os modelos Uno, Duemilanove, Diecimila, Mega, Nano, Lilypad, entre outros. Todas são programadas com a mesma linguagem e a maioria usa as mesmas conexões com o mundo exterior, possibilitando facilmente utilizar uma placa diferente. A Figura 3 mostra o modelo Mega, modelo utilizado neste trabalho. Na Seção 3.2 são discutidos os motivos da utilização deste modelo.



Figura 3 - Arduino Mega.

Fonte: Arduino.cc

2.2 Redes Móveis

Toda a comunicação feita através de uma rede sem fio é baseada no seguinte princípio: quando os elétrons se movem, criam ondas eletromagnéticas que podem se propagar através do espaço livre. O número de oscilações por segundo de uma onda eletromagnética é chamado de frequência que é medida em Hz. Quando se instala uma antena com o tamanho apropriado, as ondas eletromagnéticas podem ser transmitidas e recebidas com eficiência por receptores localizados a uma distância que depende de vários fatores, como por exemplo: frequência, potência do transmissor, etc (TANEMBAUM, 1997).

Alguns padrões e características principais das redes *wireless* e celular, como frequência, área de alcance e taxa de transmissão são apresentados na Tabela 1, sendo que os valores são aproximados, pois variam conforme o fabricante do equipamento aplicado na rede, bem como as condições do ambiente onde estas tecnologias são aplicadas (GRUBER, 2007).

Tabela 1 - Padrões IEEE - Tecnologias de rede *wireless*.

Fonte: SCHWEITZER, 2006.

Padrão IEEE	Frequência	Alcance	Taxa
802.15.1(Bluetooth)	2.4 GHz	<10 m	723 Kbps
802.15.3 (UWB)	2.4 GHz	30-50 m	10 – 55 Mbps
802.13.3^a	3.1 – 10.6 GHz	<10 m	110 – 480 Mbps
802.15.4 (Zig Bee)	868 MHz, 915 MHz, 2.4 GHz	10- 75 m	20 – 250 Mbps
802.11^a	5 GHz	< 50 m	6 – 54 Mbps
802.11b	2.4 GHz	< 100 m	2 – 11 Mbps
802.11g	2.4 GHz	< 100 m	20 – 54 Mbps
802.11i	2.4 GHz	< 100 m	20 – 54 Mbps
802.16 (WIMAX)	10 – 66 GHz	= 10 km	60 – 100 Mbps
802.16e	2 – 6 GHz	= 10 km	70 Mbps
802.20	2 – 6 GHz	= 15 km	70 Mbps
Celular (FDMA, TDMA, CDMA, GSM e suas Evoluções)	900, 1700, 1800 MHz	= sem limite, depende da rede	4,8 Mbps

2.2.1 Sistema Móvel Celular

Na década de 1950, foi proposto, inicialmente, pelo Bell Labs, o conceito de um sistema celular, utilizando a redução geográfica do alcance, para permitir o reuso de espectro de frequências para aplicação em um sistema de comunicações móveis. Entretanto, a necessidade de interconectar usuários do sistema móvel, servidos por diferentes transceptores (estações rádio base) e usuários do sistema móvel a usuários do sistema fixo, implicava num sistema de controle que requeria grande capacidade de computação, inclusive no terminal móvel (unidade do usuário). Assim, embora

extremamente bem concebido, o sistema não era viável com a tecnologia disponível na época (BERNAL, 2002).

A ideia do conceito celular, segundo Gruber (2007), constitui-se basicamente na substituição do transmissor único de alta potência (responsável pela cobertura de uma grande área) por vários transmissores de baixa potência, cada um provendo cobertura de sinal a uma pequena região (célula) da área total. A cada uma dessas estações base é alocada uma porção de números de canais disponíveis para todo o sistema. Às estações base são alocados diferentes grupos de canais, de forma que todos os canais disponíveis no sistema são alocados a um determinado número de estações vizinhas.

Gruber (2007), descreve um sistema celular básico sendo composto de:

- **Central de Comutação e Controle (CCC):** É o coração de um sistema celular, responsável pela validação dos assinantes, processamento de chamadas, interface com a rede fixa de telefonia, interface com outras CCCs, sejam elas de outra operadora ou não, geração de bilhetes de chamada, gerenciamento de *hand-off* (passagem do móvel de uma célula para outra), monitoração de alarmes das Estações Rádio Base (ERB), entre muitas outras funções. A CCC possui também a BSC (*Base Station Controller*), responsável pelo gerenciamento das ERBs. Dependendo dos serviços oferecidos por uma operadora, a CCC pode conter equipamentos como:

- *Gateway* para interface entre celulares WAP e aplicações Internet (serviços de notícias, geolocalização, e-mails, m-commerce, etc.);
- Plataformas de gerenciamento de assinantes do serviço pré-pago;
- Plataformas de serviço de correio de voz;

- **Rede de telefonia pública comutada (RPTC):** suportam os telefones fixos e interligam as CCC's dos diversos prestadores de serviço em telefonia móvel (ALENCAR, 2001).

- **Estação Rádio Base (ERB):** equipamento encarregado da comunicação com as estações móveis em uma determinada área que constitui uma célula. É a ERB quem se comunica com o assinante através da interface escolhida (CDMA, TDMA, GSM, etc.), e com a CCC através de canais PCM de voz e sinalização.

- **Estação Móvel (EM):** constitui basicamente em um transceptor portátil de voz/dados, desenvolvido para comunicar-se com os rádios das estações base em qualquer dos canais alocados.

Um sistema de celular básico pode ser ilustrado conforme a Figura 4.

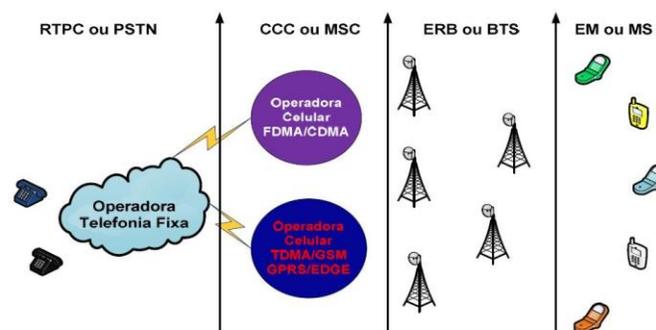


Figura 4 - Estrutura de um sistema celular básico.

Fonte: GRUBER, 2007

2.2.2 Evolução das Tecnologias de Telefonia Celular

A seguir, são apresentadas as evoluções das tecnologias de telefonia celular.

2.2.2.1 Primeira Geração (1G)

A primeira geração de sistemas celulares, chamada 1G, teve seu grande impacto na sociedade principalmente pela novidade: sem fio. Até então, todos os sistemas de telefonia móvel eram centralizados e, como consequência, tinham uma baixa capacidade de tráfego e alto custo. Essas características restringiam a poucos usuários a possibilidade de se comunicar em movimento com um serviço de telefonia (SVERZUT, 2005). Os primeiros celulares utilizavam o chamado sistema analógico de telefonia móvel — AMPS, com múltiplo acesso por divisão de frequência. Porém, o crescimento da utilização de sistemas celulares levou a necessidade do aumento da capacidade destes motivando o desenvolvimento dos sistemas digitais de segunda geração (GRUBER, 2007).

2.2.2.2 Segunda Geração (2G)

Os sistemas da segunda geração, chamada 2G, utilizam técnicas de modulação digital para transmissão via rádio. A grande diferença relativa às redes 1G prende-se principalmente na evolução analógico/digital e, conseqüentemente, pelas novas aplicações possíveis, graças ao aumento da taxa de bit. Um canal de frequência agora é partilhado por vários utilizadores, por TDMA, CDMA e GSM, e é introduzido um novo sistema hierárquico de células: as *macro-cells*, as *micro-cells* e as *pico-cells* (HÉLIO e MICHEL, 1997).

- **TDMA** – o múltiplo acesso de vários terminais a uma mesma ERB é feito alocando uma frequência para cada terminal (AMPS), ou compartilhando uma mesma faixa de frequência, mas transmitindo em tempos diferentes (GARG, SMOLIK e WILKES, 1997).
- **CDMA** – o múltiplo acesso de canais que compartilham uma mesma banda de frequências é feito pela utilização de códigos diferentes pelos vários terminais. Desta forma, vários usuários compartilham uma mesma banda de frequência (GARG, SMOLIK e WILKES, 1997).
- **GSM** – o terminal utilizado pelo assinante (Estação Móvel) é carregado com um cartão inteligente conhecido como *SIM Card*, que contratando o serviço junto a uma operadora, passa a ter acesso aos serviços fornecidos por ela. O *SIM Card* armazena entre outras informações um número de 15 dígitos que identifica unicamente uma dada estação móvel. Já o terminal é caracterizado por um número também com 15 dígitos, atribuído pelo fabricante, denominado IMEI (GRUBER, 2007).

Quando os sistemas da segunda geração (CDMA e TDMA) foram implantados, o AMPS passou a ser utilizado como alternativa para complementar a cobertura destes sistemas, devido a sua extensa cobertura. Desse modo, os terminais móveis podem passar a operar automaticamente em AMPS quando o sistema digital não está disponível. No Brasil, todas as operadoras mantêm canais AMPS em toda a área de

cobertura, de modo a garantir o *roaming* nacional para todos os assinantes (TUDE, 2003).

A estrutura flexível dos canais físicos do GSM facilitou a introdução de serviços como SMS, Fax e transporte de dados com taxas de 2,4 a 9,6 Kbps. O crescimento das aplicações de dados como acesso à Internet, e-mail, entretenimento, levou a necessidade de desenvolver soluções que permitissem o transporte de dados a taxas maiores (FIORESE, 2005). Surge assim, então, a segunda e meia geração.

2.2.2.3 Segunda e meia geração (2,5G)

A segunda e meia geração oferece serviços de dados por pacotes e sem a necessidade de estabelecimento de uma conexão permanente, a taxas de até 144 Kbps. É um passo intermediário na evolução para a 3G. Os principais sistemas dessa geração são o GPRS e o EDGE (GRUBER, 2007).

- **GPRS** – permite a estação móvel uma conexão à Internet sem a necessidade de se estabelecer uma chamada telefônica. A informação pode ser enviada e recebida imediatamente conforme a necessidade do usuário, com taxas que pode chegar teoricamente a 171,2 Kbps. Três vezes mais rápida do que as possíveis nas redes de telecomunicações fixas. Esse sistema facilita muitas aplicações, como navegação na *Web* até transferência de arquivos para automação de residências, não disponíveis das redes GSM, dadas as limitações na taxa de transferência (GRUBER, 2007). Segundo Gruber (2007), a conexão de um terminal a uma rede GPRS é feita através dos seguintes passos
 - Um terminal GPRS, ao ser energizado, será reconhecido pela rede de forma semelhante ao que ocorre com um terminal GSM para voz.
 - O próximo passo é conseguir um endereço IP estabelecendo uma conexão em GPRS. Esse endereço IP é normalmente dinâmico sendo fornecido pela operadora móvel, ou outro operador dependendo de como a rede está configurada.
 - O terminal GPRS está então pronto para enviar e receber pacotes. Ele pode então assumir os seguintes estados de forma a economizar energia: *idle* (ocioso), *ready* (pronto) em que ele pode enviar e receber pacotes instantaneamente ou *stand-by*.
- **EDGE** – está relacionado ao aumento da capacidade de transmissão da interface no corrente padrão GSM e adiciona novas características na rede GSM mantendo compatibilidade com os telefones celulares GSM/GPRS e com os equipamentos da rede (GRUBER, 2007).

2.2.2.4 Terceira Geração (3G)

A terceira geração, chamada 3G, define vários serviços móveis que proporcionam melhorias na qualidade de voz, Internet de alta velocidade e serviços multimídia. Essa evolução exigiu a definição de um novo padrão para a interface entre Estação Móvel e ERB com canais de RF de 5 MHz. Os principais sistemas são o

WCDMA e o CDMA 1xEVDO, que são evoluções dos sistemas GSM e CDMA (GRUBER, 2007).

De acordo com dados fornecidos pela Teleco, (2017a), foram registrados em agosto de 2017, 98,44 milhões de acessos 3G no Brasil. Os dados mostram uma tendência de diminuição do acesso 3G devido ao crescimento do acesso 4G, conforme ilustra a Figura 5.

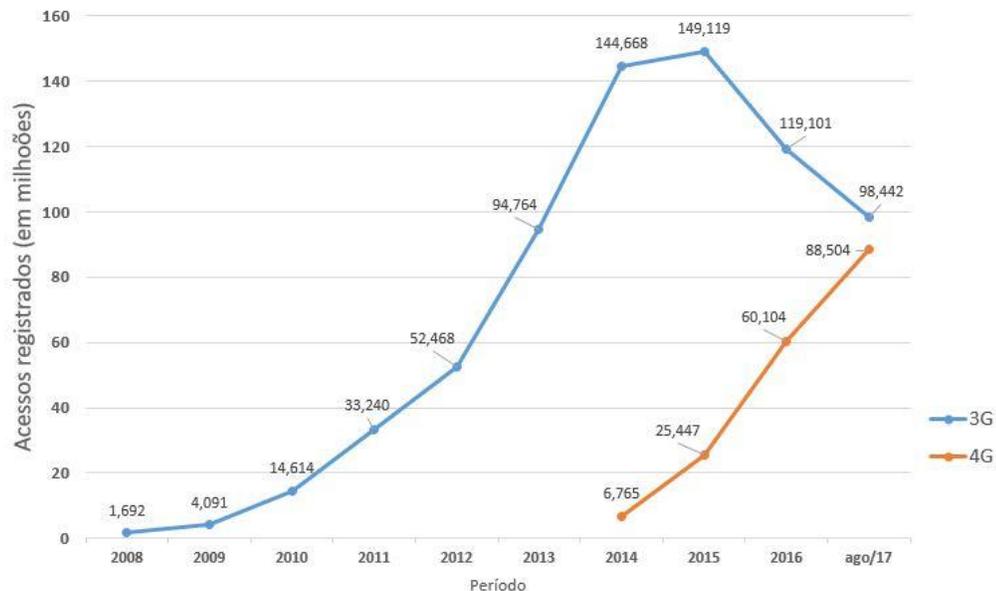


Figura 5 - Acessos de telefonia registrados no Brasil.

Fonte dos dados: Teleco, 2017a.

2.2.2.5 Quarta Geração (4G)

A cada dia que passa a necessidade por serviços de banda larga cresce. Alguns serviços que há poucos anos eram praticamente inacessíveis à maioria da população e das empresas, hoje são considerados essenciais e se tornaram amplamente difundidos. Serviços como videoconferência, *download* de vídeos, jogos interativos e Voz sobre IP, que já são considerados por muitos como necessários, e outros serviços como *Cloud computing* (nome que se dá ao crescente fenômeno de estocar dados e executar programas não no próprio computador, mas em servidores gigantes e remotos acessados pela Internet) devem aumentar cada vez mais a demanda por largura de banda (MOTA, 2009).

Ferraz et al., 2012, exemplificam que dentre os sistemas que utilizam essa tecnologia, podem ser citados o WiMAX e o LTE. Ambos atendem os requisitos da telefonia da 4ª geração, propostos pelo ITU, que são alta qualidade dos serviços móveis e taxas de dados acima de 100 Mbps.

De acordo com dados fornecidos pela Teleco, (2017b), o Brasil em agosto de 2017 já possuía 89,3 % de cobertura 4G em seu território, com aproximadamente 88,5 milhões de aparelhos ativos nas operadoras utilizando acesso pelo 4G, conforme mostra a Figura 5.

2.3 Sistema Operacional Android

O Android é uma plataforma de desenvolvimento para aplicativos móveis como *smartphones* e contém um sistema operacional baseado no núcleo Linux, uma interface visual rica, GPS, diversas aplicações já instaladas e ainda um ambiente de desenvolvimento bastante poderoso, inovador e flexível (LECHETA, 2009). O Android causou grande impacto quando foi anunciado, atraindo atenção de muitos. Podemos dizer que isso aconteceu porque por trás dele está nada mais nada menos do que o Google. Entretanto, não é apenas o Google que está na jogada, e sim um grupo formado por empresas líderes do mercado de telefonia como a Motorola, LG, Samsung, Sony Ericsson e muitas outras, criando assim o grupo *Open Handset Alliance* (OHA), que tem a intenção de padronizar uma plataforma de código aberto e livre para celulares. O resultado dessa união foi o nascimento do Android (LECHETA, 2009).

As aplicações para o Android são escritas em linguagem Java e utilizam os recursos da mesma, o código fonte (*.java*) é compilado em *bytecode* (*.class*) e então convertido para o formato (*.dex*), em *Dalvik Executable*, que representa a aplicação do Android compilada. Após essa compilação, a aplicação, com os arquivos *.dex* e outros recursos utilizados é compactada em um único arquivo com a extensão *.apk* (*Android Package File*), que representa a aplicação final pronta para ser distribuída e instalada nos dispositivos (LECHETA, 2009). O ambiente de desenvolvimento do Android (IDE), disponibiliza as ferramentas e SDK necessárias para começar a desenvolver as aplicações na plataforma utilizando a linguagem Java (OLIVEIRA, 2012).

Para auxiliar a distribuição das aplicações do Android, além da divulgação de sua nova plataforma, foi criado o site *Android Market*, renomeado em 6 de março de 2012 como Google Play. O objetivo do *site* é fornecer aos desenvolvedores de aplicativos um lugar comum para disponibilizar suas aplicações. Para publicar uma aplicação o desenvolvedor precisa pagar uma taxa e concordar com os termos de uso. Depois disso, o aplicativo já pode ser instalado pelos usuários. Existem aplicativos que são gratuitos, enquanto outros são pagos (OLIVEIRA, 2012). A Figura 6 ilustra o crescimento do número de aplicativos disponíveis no Google Play, entre dezembro de 2009 e setembro de 2017.

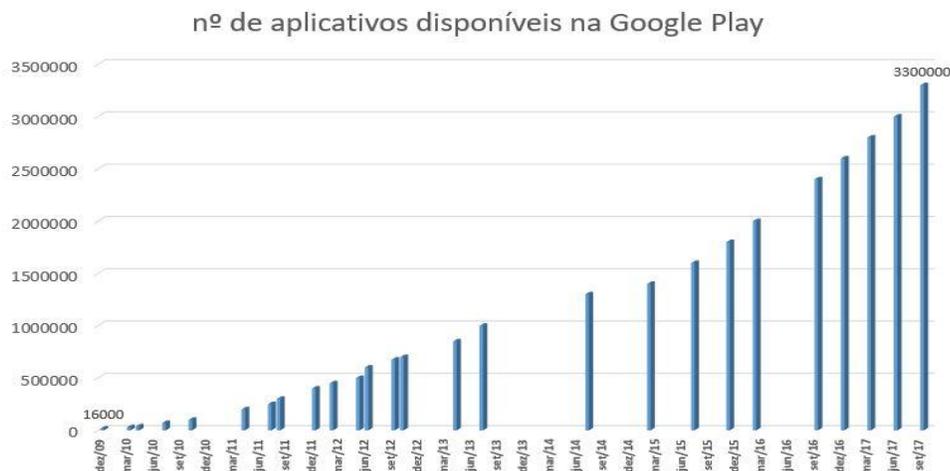


Figura 6 - Aplicativos disponíveis na Google Play.

Fonte dos dados: <https://www.statista.com/statistics/266210>.

3 Materiais e Métodos

A arquitetura básica que estrutura a implementação do projeto é ilustrada na Figura 7, onde utiliza-se um modelo de planta para o carro com alarme, onde haverá sinais de saída (*Outputs*) entrada (*Inputs*), sendo constantemente acessadas por uma central de monitoramento e comunicação, que poderá enviar os dados para um servidor *online*, disponibilizando acesso ao usuário do dispositivo móvel autorizado.

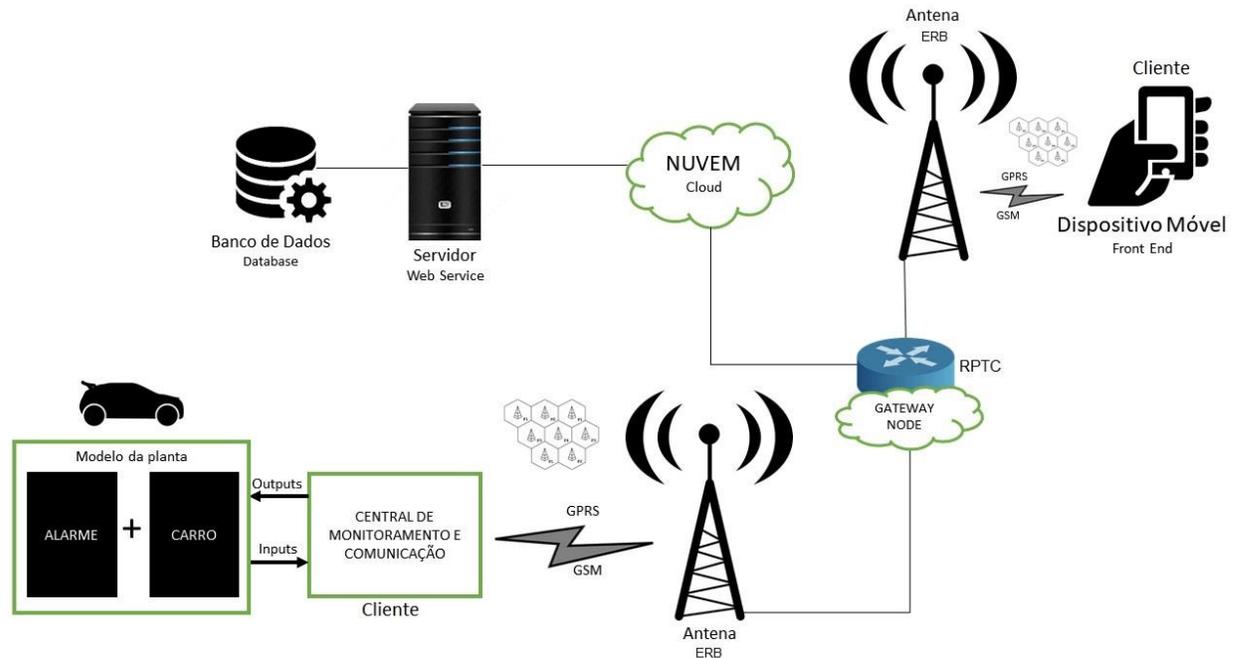


Figura 7 - Arquitetura simplificada do sistema.

Fonte: Autor.

Especificando, o dispositivo móvel é um aparelho de telefone celular com sistema operacional Android, com o aplicativo desenvolvido instalado. Através dele, será possível acessar as informações presentes no servidor na Internet, que contém um banco de dados que armazena as informações geradas pela central de monitoramento, instalada junto ao veículo. Através do dispositivo móvel também será possível enviar comandos à central, que por sua vez realizará a tarefa desejada. Tais comunicações utilizam as tecnologias de rede GSM e GPRS, para envio e recebimento de SMS e para comunicação HTTP, respectivamente. A funcionalidade por SMS é utilizada como comunicações de emergência, quando há um perigo eminente sendo relatado pela central de monitoramento. Essa opção tem como objetivo aumentar a segurança oferecida pelo sistema, pois oferece a possibilidade de monitoramento mesmo quando o usuário não possui Internet disponível no momento. Resumidamente, a arquitetura do projeto envolve quatro elementos principais: o modelo da planta, a central de monitoramento e comunicação, o servidor e o dispositivo móvel.

3.1 Modelo da Planta

A estruturação da modelo da planta é realizada utilizando o elemento principal que engloba o conceito deste trabalho: o alarme.

3.1.1 Escolha do alarme

Para poder implementar o sistema e verificar os resultados, criou-se uma planta que simula um veículo dotado de um alarme convencional. A escolha do alarme se deu após uma pesquisa de mercado realizada pelo autor, onde averiguou-se alguns modelos e avaliou-se o melhor custo-benefício encontrado. A Tabela 2 mostra os modelos pesquisados.

Tabela 2 - Alarmes Comerciais analisados.

Fonte: Autor.

Marca	Modelo	Diferencial	Preço ¹
 POSITRON	Cyber PX 360 BT Universal	Conexão via <i>bluetooth</i> para acesso a funcionalidades por aplicativo.	R\$ 264,90
 POSITRON	EX 360 Exact Universal	Exclusiva Tecnologia PAN (Pósitron Area Network), que permite que o alarme se comunique com outros produtos da linha Pósitron.	R\$ 209,90
 KOSTAL	Linha K-connect K150 Universal	Configuração de diversas funcionalidades do alarme pelo computador, via USB.	R\$ 150,00

¹ preços cotados em agosto de 2017.

Foram escolhidos para fins de comparação os modelos presentes na Tabela 2 pelo fato de serem populares no mercado automotivo e facilmente encontrados no comércio. Além disso, é importante citar que o modelo Cyber PX 360 BT Universal da Positron, na maioria dos orçamentos realizados, foi aconselhado pelos comerciantes como o mais avançado disponível em suas lojas. Tal informação foi considerada relevante para o presente trabalho, pois averiguou-se que, no mercado atual, ainda não são disponibilizadas aos usuários soluções de alarmes automotivos com aplicativos conectados à Internet, a não ser as opções de se contratar um serviço especializado de segurança. O modelo popular mais avançado encontrado fornece como maior conectividade comunicação via *bluetooth*.

Para o desenvolvimento deste projeto, foi escolhido o modelo da Kostal K-150, pelo melhor custo-benefício apresentado e por oferecer todas as funcionalidades necessárias básicas de um alarme automotivo, como travamento e destravamento por

controle remoto, sensoriamento de ultrassom e abertura de portas, saídas auxiliares para levante e fechamento de vidros e também bloqueio de partida. O kit do alarme inclui a central eletrônica, o chicote de condutores, dois sensores de ultrassom, uma sirene e dois controles remotos.

3.1.2 Montagem da planta simulando o carro

Após escolhido o modelo do alarme, utilizou-se em conjunto com seu chicote de fios padrão, uma *proto-board* para as conexões elétricas, com o intuito de simular o comportamento de um carro sendo monitorado por um alarme. Utilizou-se então o manual do usuário e de instalação da linha de alarmes Kostal.

Primeiramente, verificou-se a funcionalidade de cada fio presente no chicote do alarme, mostrado na Figura B.1, presente no Apêndice B, e encontrado com maiores detalhes no manual da Kostal. O chicote, em uma extremidade, é formado por dois conectores principais, nomeados conector de potência e conector de sinal. Na outra extremidade, os fios são soltos e desencapados, sem conector, de forma a possibilitar a conexão de cada fio condutor com a sua respectiva aplicação.

De acordo com o manual, prevê-se que o veículo onde o alarme será instalado possua alimentação de tensão de 12 V, além de toda a fiação indicada na Figura B.1, onde cada fio condutor do veículo correspondente ao chicote do alarme será desencapado e unido ao chicote. Devido ao fato do projeto não possuir um veículo disponível para a implementação, optou-se por simulá-lo em uma *proto-board*, utilizando componentes eletrônicos que fornecem um comportamento praticamente idêntico aos das aplicações previstas. Desse modo, a Tabela C.1, presente no Apêndice C, indica cada aplicação do conector de sinal que foi utilizada na *proto-board*, com sua respectiva nomenclatura escolhida e os níveis de tensão medidos.

Além dos pinos indicados da Tabela C.1, utilizou-se o pino 24 do conector de sinal e o pino 4 do conector de potência para fornecer a alimentação de 12 VCC ao alarme. Essa tensão é gerada por uma fonte de tensão, de fabricação própria do autor.

Tendo conhecimento dos níveis de tensão utilizados pelo alarme, em cada pino escolhido, e as nomenclaturas definidas, pôde-se desenhar o esquemático elétrico que posteriormente foi montado na *proto-board*. O esquemático possui dois circuitos principais, representado na Figura B.2, presente no Apêndice B: o circuito de entradas/saídas do veículo, caracterizado na região inferior e o de divisores de tensão, caracterizado na região superior. Para os sinais de entrada do veículo faz-se o uso de botões com trava, enquanto para as saídas do veículo, para melhor visualização dos acontecimentos, utiliza-se LEDs em cada uma.

O uso do divisor de tensão se faz necessário para ajustar os níveis de tensão utilizados pelo alarme ao nível de tensão utilizado nas entradas digitais do microprocessador embarcado na placa Arduino MEGA 2560, que para esse caso, não podem ultrapassar os 5,5 V, de acordo com o fabricante ATmega. Logo, a tensão desejada para as entradas digitais foi definida em 5 V.

Através da técnica de divisor de tensão por resistores, é possível obter a tensão desejada pela relação entre os resistores utilizados. Os valores foram calculados utilizando os níveis de tensão relatados na Tabela C.1 e escolhendo, primeiramente, o valor de um dos resistores, podendo assim obter-se o valor do outro resistor para cada

um dos segmentos. A Tabela 3 mostra os valores dos resistores calculados, já aproximados aos valores comerciais.

Tabela 3 - Valores Resistores dos divisores de tensão

Fonte: Autor.

Ramo	V entrada	V desejada	Ra	Rb
FAROL	12 V	5 V	15 k Ω	10 k Ω
IGN	12 V	5 V	15 k Ω	10 k Ω
LUZTETO	12 V	5 V	15 k Ω	10 k Ω
PORTAS	5V (PULL UP)	5 V	-	
CAPOEPM	5V (PULL UP)	5 V	-	
ABREPORTAS	9,1 V	5 V	8,2 k Ω	10 k Ω
TRAVAPORTAS	10 V	5 V	10 k Ω	10 k Ω
VIDROUP	9,2 V	5 V	8,2 k Ω	10 k Ω
VIDRODOWN	9,3 V	5 V	8,2 k Ω	10 k Ω
RADIO	12 V	5 V	15 k Ω	10 k Ω
SIRENE	5,6 V RMS	5 V	1 k Ω	10 k Ω
ATIVA_ALARME	5 V	3,3 V	6,8 k Ω	10 k Ω

Conforme mostrado na Tabela 3, não foi necessário o cálculo para os resistores dos ramos PORTAS E CAPOEPM, devido a esses, quando não conectados à referência, ficam flutuantes. Deste modo, o uso de resistores *Pull Up* é a técnica utilizada para obter a tensão de 5 V desejada quando estes estão flutuantes. Ao centro do circuito da Figura B.2 é possível verificar suas conexões. Outra observação importante é com relação ao ramo SIRENE, que fornece na tensão de entrada um sinal modulado, pois o princípio de funcionamento do som da sirene utiliza esta técnica de modulação. O que interessa para este circuito não é a forma de sinal, mas sim seu valor eficaz da tensão. Desse modo, foi medido um valor eficaz de 5,6 V.

Para os circuitos que auxiliam a visualização dos eventos, utilizando LEDs, foi implementado os circuitos ilustrados no esquemático da Figura B.3.

A montagem da planta, realizada na *proto-board* é mostrada na Figura B.4, também disponível no Apêndice B, com cada ramo do circuito referenciado. A Figura B.4 mostra com maior detalhe os componentes utilizados para simular o veículo, onde os fios que surgem da região inferior da imagem são os condutores da central eletrônica do alarme Kostal, ilustrados anteriormente na Figura B.2 e referenciados na Figura B.1. Além disso, é mostrado o uso do circuito integrado 7805, responsável por fornecer a tensão de 5 V nos ramos que necessitam desse nível de tensão. No caso, os resistores de *Pull Up*. Optou-se por não utilizar o pino 5 V do Arduino MEGA 2560 para garantir uma maior estabilidade do nível de tensão e deixar a placa livre desta responsabilidade.

3.2 Central de Monitoramento e Comunicação

Considera-se a central de monitoramento e comunicação o elemento principal do sistema, pois é através dela que ocorrem as principais tarefas, que são a leitura dos

dados da planta e a comunicação com a rede móvel, que fornecerá os dados ao usuário. Decidiu-se para o presente projeto utilizar uma plataforma de prototipagem eletrônica, que contém um microprocessador responsável pelo processamento das informações e comandos da aplicação. A empresa Arduino® disponibiliza vários modelos de plataformas de prototipagem, cada uma com especificações e características diferentes. Para este projeto, utilizou-se o modelo MEGA 2560, ilustrado na Figura 3, por possuir as seguintes características que foram consideradas essenciais para o desenvolvimento do projeto:

- **Tensão de operação: 5 V;**
 - O *shield* GSM/GPRS que é acoplado ao Arduino opera em 5 V também;
- **Tensão de alimentação: 12 V;**
 - O sistema pode posteriormente ser adaptado diretamente ao veículo, que possui bateria de 12 V;
- **54 pinos digitais (I/O)**, com 6 entradas podendo ser utilizadas como interrupções externas;
 - O projeto pode exigir durante seu desenvolvimento uma quantidade grande de entradas digitais;
- **Memória *Flash*:** 256 kB, capaz de armazenar todo o programa desenvolvido;
- **Memória *RAM*:** 8 kB, capaz de administrar o fluxo do processamento de dados exigidos pela aplicação;
- **Alta compatibilidade** com outros modelos e *shields* do mercado;

3.2.1 Arquitetura

Utilizando o esquemático elétrico ilustrado na Figura B.2, projetou-se então as conexões da central ao modelo da planta, conforme é mostrado na Figura 8.

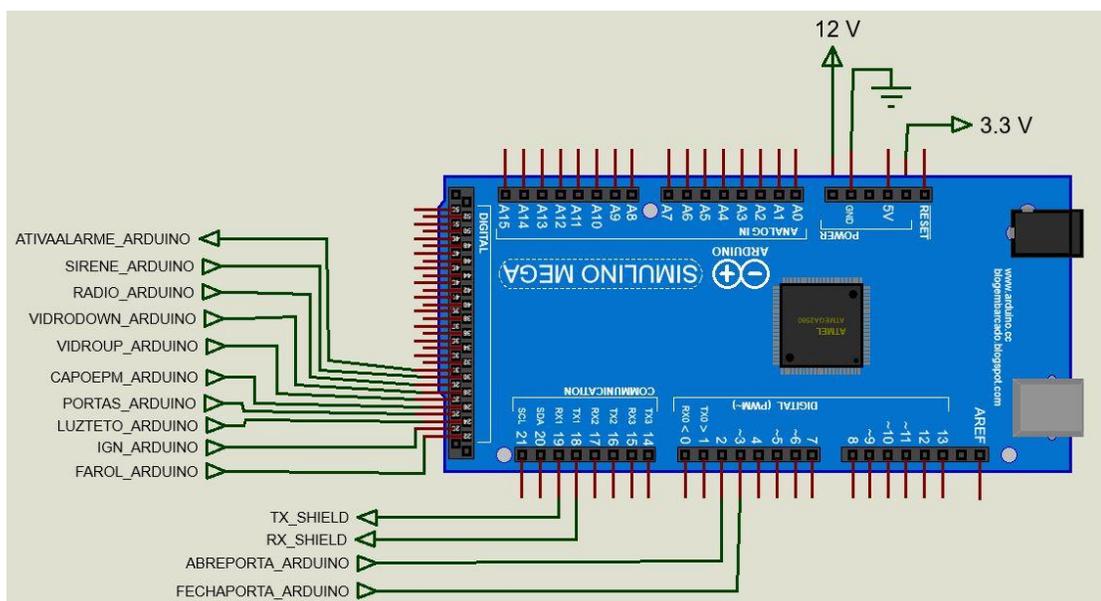


Figura 8 - Conexões com o Arduino.

Fonte: Autor.

Foram utilizados os pinos 2 e 3 do Arduino MEGA 2560 para as variáveis ABREPORTA_ARDUINO e FECHAPORTA_ARDUINO, pois esses pinos oferecem suporte para interrupções externas e, como no funcionamento do alarme, o evento dessas variáveis são dados por um pulso negativo de poucos segundos, é interessante utilizar essa funcionalidade, pois a qualquer momento, independentemente de onde o código esteja processando, a função atribuída à interrupção do pino será executada. Após o término dessa função, o código retoma o processamento de onde parou.

Devido aos pinos 0 e 1 serem utilizados por padrão para comunicação serial entre Arduino e o computador, para utilizar o Monitor Serial da IDE, faz-se necessário a configuração de uma nova comunicação serial, pois o *shield* GSM/GPRS acoplado se comunica com o Arduino através deste tipo de protocolo. Para isso, utiliza-se o método chamado *Software Serial*, que permite configurar outros pinos digitais do Arduino para funcionarem como transmissor (TX) e receptor (RX) da comunicação. Para o presente projeto, escolheu-se os pinos 18 (TX) e 19 (RX) para estabelecerem a comunicação serial entre o Arduino MEGA 2560 e o *shield* GSM/GPRS.

O modelo do *shield* GSM/GPRS utilizado para a execução do projeto é um IComsat v1.1, da fabricante Itead®, baseado no módulo SIM900 *Quad-band* GSM/GPRS e totalmente compatível com os modelos UNO e MEGA das placas Arduino. Seu funcionamento é a base de comandos AT e, atualmente, são disponibilizadas bibliotecas que podem ser incluídas às bibliotecas da IDE do Arduino para facilitar as rotinas de programação. Essas bibliotecas foram utilizadas para o projeto e são facilmente encontradas na Internet. O modelo do *shield* mencionado é mostrado na Figura 9.



Figura 9 - Shield GSM IcomSat v1.1.

Fonte: Itead.cc

A primeira etapa da configuração do *shield* foi conectá-lo à placa Arduino. Seus pinos são padronizados para encaixarem perfeitamente sobre o padrão dos pinos das placas Arduino.

A Figura 10 mostra como foram conectados os pinos responsáveis pela comunicação serial entre o *shield* e o Arduino MEGA.

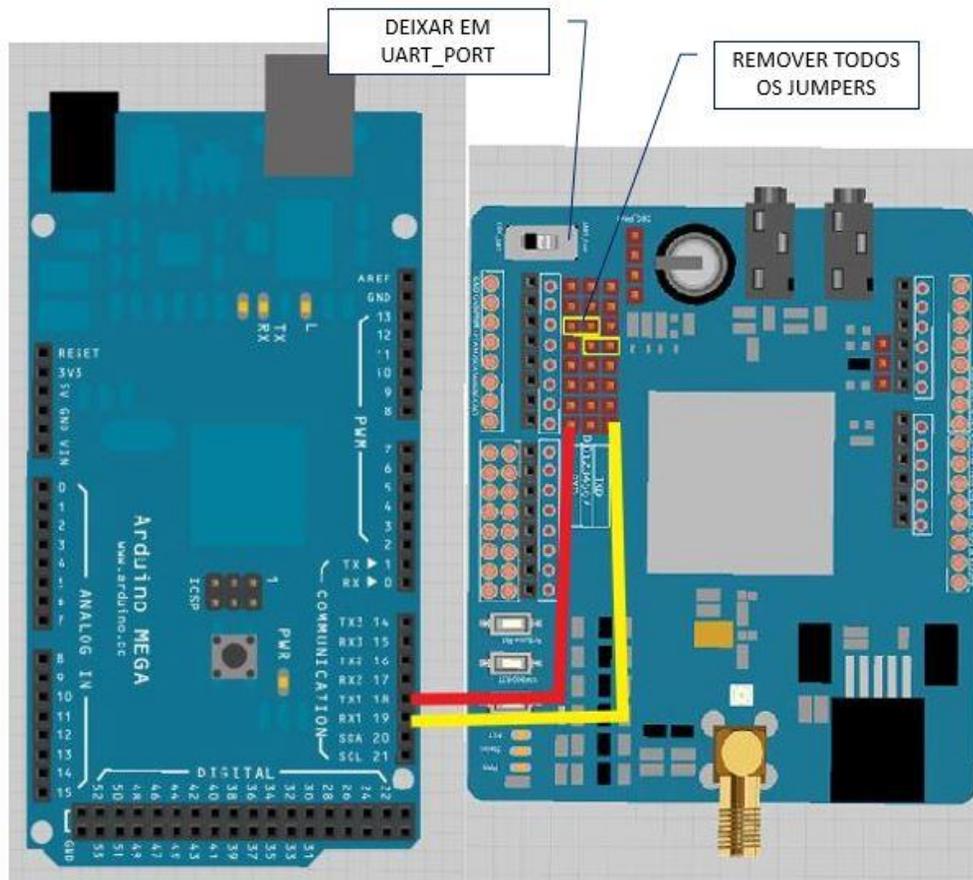


Figura 10 - Conexão para comunicação serial entre Arduino e *shield* GSM/GPRS.

Fonte: Autor.

As figuras 8 e 10 não ilustram o encaixe entre a placa Arduino e o *shield* GSM/GPRS para facilitar a visualização. Além de realizar as conexões da Figura 10, e encaixar o *shield* ao Arduino MEGA, foi preciso conferir e configurar alguns arquivos da biblioteca incluída na IDE do Arduino. Foram necessárias as seguintes alterações, respectivamente:

- Arquivo GSM.cpp
 - Definir os pinos utilizados como *Software Serial*:

```
#define _GSM_TXPIN_ 18
#define _GSM_RXPIN_ 19
```

- Arquivo GSM.h
 - Definir a placa sendo utilizada com o *shield* (comentar a não utilizada):

```
// #define UNO
#define MEGA
```

- Arquivo HWSerial.h
 - Tirar dos comentários a primeira linha:

```
#define MEGA
```

Outro detalhe importante tomado antes de ligar as duas placas é em relação a fonte de alimentação. De acordo com o fabricante, o *shiled* GSM IcomSat necessita de uma fonte que consiga fornecer no mínimo 1 A, pois, em algumas requisições, o *shield* pode exigir uma considerável quantidade de corrente. Sendo assim, não se pode alimentar o acoplamento das duas placas apenas com a alimentação da conexão USB do computador, pois a mesma não consegue fornecer corrente suficiente. Deve-se fazer uso de uma fonte externa que consiga fornecer correntes de no mínimo 1 A. A fonte utilizada no projeto, comentada na seção 3.1.2, de fabricação própria do autor, consegue fornecer até 2 A de corrente máxima, atendendo assim a exigência do sistema.

3.2.2 Circuito do comando remoto

Para permitir que a central envie um comando ao alarme e este ative-o, fez-se a adaptação de um dos controles remotos adquiridos do alarme, onde se abriu a carcaça e soldou-se um condutor entre o pino ATIVAALARME_ARDUINO (pino 30) do Arduino e o botão mecânico presente no controle remoto. Desse modo, ao invés de o sinal por rádio frequência emitido pelo controle ser enviado ao aperto do botão, esse sinal é enviado quando um sinal de 0V é aplicado na porta ATIVAALARME_ARDUINO por um pequeno intervalo de tempo. As figuras 8 e B.2 ilustram as ligações elétricas desse circuito. Complementando, o pino 30 do Arduino foi configurado para permanecer com 5 V e apenas quando o comando remoto for solicitado, o pino ficará com 0 V por alguns instantes de tempo.

3.2.3 Montagem da central de monitoramento e comunicação

A montagem da central, com o circuito de acionamento remoto é mostrada na Figura B.5, disponível no Apêndice B, com cada ramo do circuito referenciado.

3.2.4 Funcionamento geral da central

Tendo conhecimento do esquemático elétrico do veículo ilustrado nas figuras B.2 e B.3, além de ter conhecimento de todas as variáveis que a central deve gerenciar, determina-se um fluxograma de funcionamento do sistema, representado na Figura 11.

O monitoramento dos sensores de ultrassom é feito pelo próprio alarme, não sendo utilizada instrumentação direta entre esses sensores e alguma porta do Arduino. A solução adotada para saber se o alarme disparou pelo ultrassom foi monitorar a sirene do alarme juntamente com todas as outras variáveis. Ou seja, se nenhuma variável indica disparo, porém a sirene indica, significa que o disparo ocorreu pelos ultrassons.

A seguir é ilustrada o fluxograma citado da Figura 11.

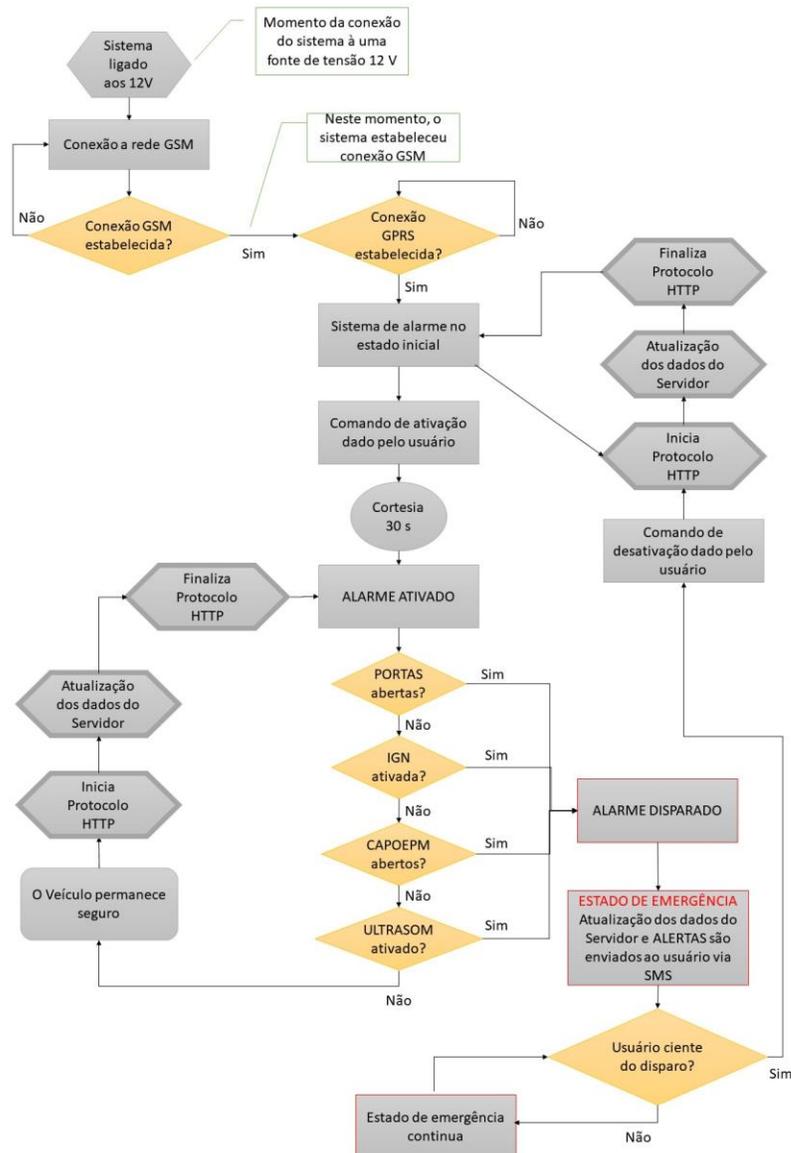


Figura 11 - Fluxograma de funcionamento da central.

Fonte: Autor.

Duas conexões são estabelecidas pela central. A primeira, na tecnologia GSM, que permite à central já utilizar a funcionalidade de SMS, caso necessário. Após estabelecida esta primeira conexão, a central entra em um laço de detecção, lendo as variáveis de interesse e enviado ao servidor. Para isso, uma segunda conexão é estabelecida, desta vez por GPRS, que permite que o protocolo HTTP seja executado.

Ao detectar que o usuário enviou um comando de ativação, o alarme por padrão disponibiliza 30 segundos de cortesia, antes do alarme ser efetivamente ativado, para

caso o usuário necessite rapidamente pegar algo que tenha esquecido dentro do veículo. Com o alarme plenamente ativado, o laço de detecção continua, onde cada variável de interesse é lida e verificada a condição para o alarme estar disparado. Caso nenhuma restrição de disparo seja detectada, o sistema conclui que o carro permanece seguro e atualiza os dados no servidor, executando novamente o protocolo HTTP.

O tempo de ciclo desse laço é determinado pelos tempos de execução dos algoritmos de leitura e comparação das variáveis, somados ou aos tempos de execução dos comandos AT para o envio de SMS por GSM, ou aos tempos de execução dos comandos AT para o protocolo HTTP por GPRS. Além disso, entre os comandos AT são colocados alguns atrasos, chamados *delays*, a fim de garantir que os comandos AT enviados ao *shield* GSM/GPRS sejam executados de maneira correta. O tempo gasto no processamento dos algoritmos de leitura e comparação é praticamente desprezível em relação ao relacionado aos comandos AT, sendo esses os que determinam o tempo de ciclo total da rotina de monitoramento e envio dos dados.

Caso alguma das variáveis lidas forneça a condição de violação do veículo, o sistema entra em estado de emergência, onde além de atualizar os dados no servidor, envia alertas via SMS ao usuário. Apenas quando o usuário se mostrar ciente do disparo e enviar um comando de desativação, o alarme atualizará os dados no servidor e retornará ao estado inicial.

3.2.5 Leitura das variáveis e chamada das funções

Para verificar o estado atual de cada variável de interesse do sistema, é utilizado o comando *digitalRead()*, presente na biblioteca padrão do Arduino. Deste modo, logo no início do laço de detecção este comando é executado em cada pino responsável pela variável. Após isso, uma sequência de comparações é realizada com os resultados da aquisição para definir se é necessário algum tipo de tarefa ou comando.

Um exemplo de como isso foi implementado é ilustrado no código da Figura 12:

```
...
void loop()
{
  farois = digitalRead(pin_farois);
  pos_chave = digitalRead(pin_pos_chave);
  luz_teto = digitalRead(pin_luz_teto);
  portas = digitalRead(pin_portas);
  capoepm = digitalRead(pin_capoepm);
  radio = digitalRead(pin_radio);
  sirene = digitalRead(pin_sirene);
  ...
  if(pos_chave==HIGH && cortesia==0)
  {
    ...
  }
  ...
}
...
```

Figura 12 - Aquisição e comparação no laço de detecção.

Fonte: Autor.

Caso alguma interrupção externa seja detectada, a função relacionada a interrupção é executada e, só após o término dessa função, que o programa retoma de onde parou.

O código completo com o restante das linhas programadas pode ser conferido no *link* disponível em APÊNDICE A. Os códigos completos não foram disponibilizados no texto do presente trabalho devido à grande quantidade de linhas de código, inviabilizando a disposição dos mesmos no texto.

Para a criação dos dados que são enviados para o servidor via HTTP, ou por SMS, é utilizado um vetor de *bytes*, onde cada posição do vetor representa uma variável, no qual o valor do *byte* representa o estado em que a variável se encontra. A atualização do valor do *byte* se dá logo após a leitura das entradas digitais. O valor do *byte* escolhido para representar cada estado de cada variável não foi divulgado, pois pode ser qualquer valor que se deseja utilizar. A Figura 13 exemplifica como tal metodologia foi implementada.

```

char array_dados[9] = "00000000";
/*array_dados[0] => disparado          valorX= não          valorY= sim
 array_dados[1] => status alarme      valorX= desativado  valorY= ativado
 array_dados[2] => farois             valorX= desligados  valorY= ligados
 array_dados[3] => pos_chave          valorX= desligada   valorY= ativada
 array_dados[4] => luz_teto           valorX= desligada   valorY= ligada
 array_dados[5] => portas              valorX= fechadas    valorY= abertas
 array_dados[6] => capoepm            valorX= fechados    valorY= abertos
 array_dados[7] => radio               valorX= desligado   valorY= ligado
*/

```

Figura 13 - Vetor de *bytes* para envio da informação.

Fonte: Autor.

Tanto na programação do Arduino, quanto, posteriormente, na programação do aplicativo, o estado atual do veículo é obtido utilizando essa metodologia, que compara cada posição do vetor de *bytes*.

3.3 Servidor de Dados

Para a implementação do servidor de dados, responsável pelo armazenamento e disposição das informações obtidas no veículo na Internet, utilizou-se de um serviço online, disponibilizado pela 000webhost, que fornece, gratuitamente aos seus usuários cadastrados, hospedagem de sites com suporte para PHP, MySQL, FTP, entre outras funcionalidades importantes para quem necessita de um *Webservice*.

A etapa inicial do servidor foi registrar uma conta neste site de hospedagem e criar um URL para o servidor do projeto, de modo a ser utilizada como endereço das comunicações por HTTP.

3.3.1 Banco de Dados

Após a criação da URL, foi configurado o banco de dados que será responsável por guardar as informações recebidas da central de monitoramento. Ao se criar um

banco de dados, é solicitado a definição de um nome, um usuário e uma senha para ele. Após estas definições iniciais, o banco de dados já está acessível para configuração.

Sabe-se que é de interesse da aplicação duas informações básicas:

- Status do veículo recebido
- Horário do recebimento do status

Desse modo, acessando o banco de dados pela opção *phpMyAdmin*, que permite configurar os campos do banco, incluir/excluir registros, executar comandos SQL, criar e alterar tabelas, entre outras opções, criou-se a tabela, com duas colunas e seus respectivos nomes, responsável pelo armazenamento das informações mencionadas acima, através da interface disponibilizada. A tabela criada no banco de dados é mostrada com maiores detalhes na Tabela C.2, no Apêndice C.

3.3.2 Arquivos PHP

Os arquivos PHP são importantes na aplicação pois são eles os responsáveis por acessar e atualizar o banco de dados criado, além de possibilitar protocolos HTTP para enviar e receber informações. Não obstante, através deles também pode-se montar a página web da maneira que o programador desejar exibir as informações do banco de dados, através da linguagem HTML.

Para o presente trabalho, foram criados 4 arquivos .php para gerenciar o servidor.

- **connect.php** : contém as informações para acesso ao banco de dados (usuário e senha);
- **add.php**: recebe via protocolo HTTP a informação enviada por comando GET do *shield* GSM, acessa o banco de dados e grava a informação recebida;
- **getdata.php**: esse arquivo obtém os dados do bando de dados, cria um vetor de objetos JSON que serão utilizados pela aplicação Android, quando esse solicitar ao servidor;
- **index.php**: monta a página na web de modo a permitir que o usuário visualize os dados contidos no banco de dados;

Os códigos para todos os arquivos *.php* citados encontram-se no *link* disponibilizado no Apêndice A. Os nomes dados à sintaxes a seguir são genéricos, não sendo os mesmos utilizados no projeto.

A principal linha de código do arquivo *connect.php* é a função que conecta o servidor ao banco de dados. Tal função utiliza a seguinte sintaxe:

```
$connection = mysqli_connect($server, $user, $pass, $db);
```

No arquivo *add.php*, após estabelecer a conexão utilizando a função presente em *connect.php*, se houver uma requisição GET no servidor, dada pelo *shield*, a

um projeto de aplicativo é feita basicamente pela árvore de projeto apresentada na Figura 14.

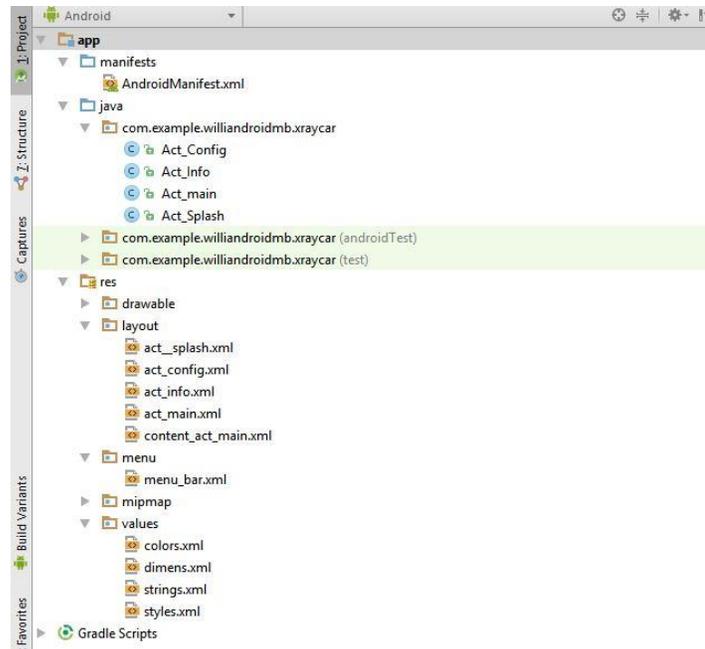


Figura 14 - Árvore de projeto Android Studio.

Fonte: Autor.

O diretório */manifest* possui um arquivo *.xml* nomeado *AndroidManifest.xml*. Este arquivo tem a importância de fornecer permissões ao aplicativo, como acesso à Internet, vibração, acesso à câmera, entre outros, além de definir alguns parâmetros das *activities*, explicadas no decorrer do texto.

O diretório */java* contém os códigos *.java* das tarefas que cada *activity* criada deve executar, com a ocorrência ou não de eventos na tela do *smartphone*.

O diretório */res* contém todos os arquivos utilizados para gerar a interface e a aparência da aplicação. Nele estão contidos outros subdiretórios. O subdiretório */res/drawable* contém todos os arquivos de imagem que são utilizados no projeto. Em */res/layout* estão definidas as *activities* (*.xml*) do projeto.

Uma *activity* é um componente do aplicativo que fornece uma tela com a qual os usuários podem interagir para executar alguma tarefa, como discar um número no telefone, enviar um SMS, solicitar dados, etc. Cada *activity* recebe uma janela que exibe a interface do usuário. Em cada uma são adicionados os elementos de iteração desejados, como botões, caixas de texto, tabelas, menus, etc. Todos com milhares de possibilidades de edições. Por esse motivo que cada *activity* recebe um arquivo *.java* responsável pela escuta dos eventos da tela e execução das tarefas programadas.

O diretório */res* ainda contém o subdiretório */res/mipmap*, que guarda os ícones utilizados, com diferentes resoluções para se adaptar automaticamente aos diferentes tamanhos de tela. E finalmente, o subdiretório */res/values* contém alguns valores que são globalmente utilizados pelo projeto, como definição de cores, textos que serão exibidos, estilos aplicados aos componentes das *activities* ou até mesmo nas próprias *activities*, entre outros parâmetros.

3.4.1 Definições de projeto para o aplicativo

Os requisitos iniciais definidos para o aplicativo foram os seguintes:

1. Conter uma tabela/*grid* que mostre como cada variável monitorada se encontra;
2. Informar o horário da última atualização das variáveis no servidor;
3. Através de um toque específico na tela, buscar no servidor os dados da última atualização e atualizar a tabela;
4. Permitir, através do aplicativo, acessar a aplicação padrão de SMS para envio de mensagens à central;

Os códigos completos do projeto do aplicativo podem ser encontrados no link disponibilizado no Apêndice A.

A interface final disponibiliza ao usuário a tabela principal, que fornece todas as informações, além de um indicador que mostra qual o horário referente a atualização que é mostrada na tabela. Além disso, há um botão para acesso aos comandos possíveis por SMS, localizado no canto inferior direito. Os comandos são: ATIVA, para ativar o alarme remotamente, e STATUS, para obter através de uma SMS as mesmas informações disponibilizadas na tabela/*grid* do aplicativo. Há também um botão para ativar alarme remotamente via comando HTTP. Essa funcionalidade não foi implementada para o presente projeto, pois não estava prevista nos requisitos iniciais. Porém, tal funcionalidade é discutida nas considerações e trabalhos futuros do Capítulo 5. O resultado da interface é mostrado posteriormente no Capítulo 4 dos resultados.

3.2.4 Funcionamento geral do aplicativo

A ideia do aplicativo é ter uma *activity* principal, na qual haverá a tabela de informações do veículo, facilmente visualizadas pelo usuário. Quando houver a necessidade de atualizar a tabela, o usuário deve deferir um toque sobre a mesma. Todo o processo de conexão e leitura das informações do servidor é feita por uma classe utilizando uma extensão da classe *AsyncTask*, presente nas ferramentas do Java, que possibilita rotinas acontecerem paralelamente, de forma assíncrona. Sendo assim, o processo de obtenção das informações no servidor ocorre de maneira que a tela principal não fique travada até que a tabela seja atualizada.

Quando houver a necessidade de utilizar as funções por SMS, a *activity* principal deve fornecer ao usuário um botão que lhe dá acesso a tal funcionalidade. Já no aplicativo padrão de SMS, o usuário deve procurar o número de telefone correspondente ao SIM CARD do sistema, que está gravado na sua lista de contatos, e realizar o comando desejado. Isso implica que o usuário possua um plano de dados com uma operadora móvel de telefonia para manter os serviços de SMS e Internet ativos. Consequentemente, o sistema possui um custo mensal atrelado a esse plano.

Avaliando os requisitos apresentados anteriormente, definiu-se um fluxograma básico que exemplifica o funcionamento do aplicativo, conforme mostrado na Figura 15.

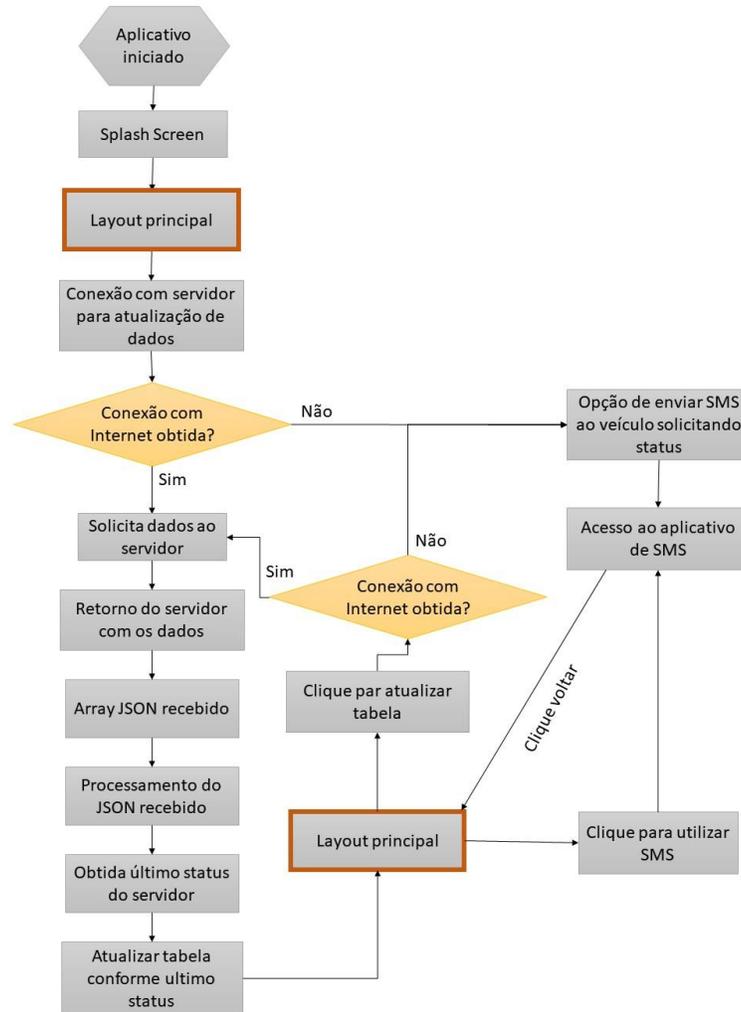


Figura 15 - Fluxograma de funcionamento do aplicativo

Fonte: Autor.

Deste modo, ao se iniciar o aplicativo, a busca pelos dados armazenados no servidor já é executada para atualizar a tabela, retornando à *activity* principal os dados atualizando os *textviews* presentes na tabela. Quando o usuário desejar utilizar o SMS, basta clicar no botão inferior direito da tela, visualizado na tela esquerda da Figura 16. Para essa execução, foi necessário descobrir o *package name* da aplicação padrão de mensagens do Android. Toda aplicação já criada possui um diretório único, chamado *package name*. É através dele que os aplicativos da Play Store são rastreados. O *package name* da aplicação padrão de mensagens do Android é "com.android.mms". Para o presente aplicativo, foi criado o *package name* "com.example.williamandroidmb.xraycar". Se futuramente o aplicativo for disponibilizado na *Play Store*, este é o diretório que será vinculado a ele.

4 Resultados e Discussão

A seguir são apresentados as discussões e resultados obtidos na execução do projeto.

4.1 Comportamento da planta

Em todas as ocasiões e testes realizados com a planta montada em *proto-board*, não houve nenhum comportamento anormal averiguado, sendo todos os eventos de entrada e saída se comportando estavelmente. A fonte de tensão utilizada foi o suficiente para alimentar todo o circuito, pois, como foi mencionado na seção 3.2.1, a exigência maior de potência do sistema se dá nas requisições GSM/GPRS do *shield*, que podem consumir aproximadamente 1 A de corrente. A fonte de tensão utilizada consegue fornecer até 2 A.

Tanto a ativação do alarme por controle remoto quanto a ativação pela central de monitoramento e comunicação funcionaram perfeitamente. O alarme ativava instantaneamente após ocorrer o comando. Para a ativação pela central, foi definida na programação da rotina de execução, em que o sinal momentâneo de 0 V é aplicado duas vezes à porta ATIVAALARME_ARDUINO. Desse modo, criou-se uma repetição para buscar garantir que o comando seja recebido pelo alarme.

4.2 Comportamento da central

A central se mostrou suficiente para executar toda a programação implementada ao sistema. Com a fonte externa fornecendo a potência ao sistema, em nenhum momento houve reinicialização, ou falha, na execução das tarefas por falta de potência fornecida.

O fato de a transmissão dos dados ser relativamente pequena, ou seja, utilizado apenas um vetor com 8 *bytes* de informação, permitia que a quantidade de dados no envio fosse pequena, fornecendo ao usuário uma atualização rápida e suficiente sobre o estado de seu veículo. Foi medido que o tempo médio entre o disparo do alarme e o aviso recebido no telefone celular se dá por volta de 10 segundos. Além disso, o tempo médio de atualização dos dados no servidor se dá em torno de 40 segundos, devido, principalmente, aos vários *delays* incluídos na rotina de código.

Em relação à utilização do microprocessador do Arduino MEGA 2560 escolhido, é informado pela IDE do Arduino que foram utilizados apenas 16,6 kBytes (6%) do espaço total de armazenamento para programas, que é de aproximadamente 25 kBytes. Para a memória dinâmica (RAM), o projeto utiliza cerca de 3,9 kBytes (47%) da capacidade total, restando assim 4,3 kBytes.

4.3 Comportamento do aplicativo

O aplicativo, para rodar o que foi previsto na seção 3.4, operou sem nenhuma falha. Porém, algumas falhas foram notadas durante a execução, em situações e condições que não foram previstas no projeto, como falha de conexão com o servidor. Com exceção destas observações, o aplicativo funcionou perfeitamente para os preceitos previamente definidos. O resultado final da interface do aplicativo, assim como a tela do *smartphone* utilizando a funcionalidade por SMS, funcionando com todos os comandos possíveis e indicando uma situação em que o alarme foi disparado, são mostrados na Figura 16. A tela da esquerda mostra como ficou a visualização do aplicativo desenvolvido, quanto a tela da direita mostra o aplicativo SMS padrão do *smartphone* sendo utilizado.



Figura 16 -Funcionalidades do sistema no *smartphone*.

Fonte: Autor.

Em nível de comparação, a solução implementada no presente projeto se assemelha com algumas soluções já disponíveis. A empresa americana Viper, por exemplo, fornece uma tecnologia que eles chamam de *SmartStart Modules*, que são diferentes módulos, cada um com uma funcionalidade específica. Dentre elas: rastreamento por GPS, desativação de alarme por tela ou apenas por proximidade do *smartphone* ao veículo. Todas as soluções utilizam de um aplicativo instalado no telefone. Puhala, 2014, exemplifica outra solução semelhante, chamada *CarLock*, que utiliza a conexão OBD (*On-Board Diagnostics*) do veículo para obter as informações e disponibilizá-las em um aplicativo completo que, consegue informar se a bateria foi desconectada, se o motor deu partida, se o veículo começou a se mover, permitindo inclusive o rastreamento do mesmo, entre outras funcionalidades.

5 Considerações Finais e Trabalhos Futuros

Nota-se que, para um sistema de segurança, o projeto implementado ainda possui muitas oportunidades de falhas e violações, como por exemplo, a utilização do método GET no protocolo HTTP. Analisando de forma preditiva, qualquer pessoa/*hacker*, tendo conhecimento da URL utilizada e de como os arquivos PHP do servidor funcionam, podem adicionar em qualquer navegador de Internet um comando pela barra de procura, que registra falsamente os dados no banco de dados, fornecendo uma informação falsa ao usuário. Contudo, foram ainda notadas algumas falhas de execução no aplicativo, não previstas no projeto, como a falha encontrada ao abrir o aplicativo. Se não há nenhum tipo de conexão com a Internet, seja por WiFi ou por 4G/3G, a execução da tarefa de obtenção dos dados colapsa e o aplicativo fecha inesperadamente.

Como trabalhos futuros, pode-se pensar em otimizar o aplicativo, implementando uma segunda versão, que corrija os *bugs* encontrados nesta primeira versão, além de oferecer ao usuário novas funcionalidades, como localização GPS e o comando remoto por protocolo HTTP, utilizando o botão inferior maior da tela da esquerda da Figura 16, além do já implementado via SMS. Outra melhoria interessante de se realizar ao projeto é, ao invés de utilizar uma plataforma de prototipagem em conjunto com um alarme, juntar as funcionalidades de ambas em um circuito só, fazendo com que o mesmo microprocessador realize as funções do alarme e também as funções da central de monitoramento. Todo esse circuito pode ser implementado em uma placa de circuito impresso dedicada, trazendo assim maior robustez ao circuito elétrico do sistema. Pode-se pensar também em adaptar o projeto à tecnologia OBD II, presente no Brasil desde 2010, que se trata de uma conexão padronizada nos veículos que permite acesso às informações do sistema do veículo para diagnósticos rápidos. Outra solução interessante também seria adaptar o projeto para utilizar protocolo MQTT, ao invés do HTTP, pois permite diversas vantagens em relação ao protocolo atual, como alta taxa de entrega de mensagens e melhor estabilidade em conexões de baixa qualidade, como a conexão GSM, por exemplo.

Mesmo diante das possibilidades de falhas de segurança encontradas, o sistema se mostrou eficiente no monitoramento e segurança de veículos utilizando um alarme automotivo convencional, pois atualiza constantemente os dados de maneira suficientemente rápida para a aplicação, permitindo uma rápida resposta do usuário à uma situação de furto ou semelhantes.

Por fim, o sistema projetado se mostrou, apesar das falhas de segurança ainda encontradas, uma forte alternativa para a segurança pública de veículos, pois possibilita maior comodidade e tranquilidade aos seus proprietários, além de alertá-los rapidamente para uma futura ação corretiva em caso de furto. A cobertura de sinal das operadoras telefônicas no Brasil é ampla, permitindo que o sistema funcione em praticamente qualquer lugar. Basta o usuário ter adquirido um *SIM CARD* e manter ele em dia com os pagamentos dos serviços de internet e SMS disponibilizados pela operadora escolhida.

Referências

ABREU, J. M. e LOURENÇO, L.C. **Mídia, violência e segurança pública: novos aspectos da violência e da criminalidade no Brasil**. Revista Âmbito Jurídico. São Paulo, n. 74, 2010.

ALENCAR, M. S. **Sistemas de Comunicações**. São Paulo, Editora Érica, 2001. ISBN:85-7194-838-0.

ARDUINO E CIA. **Tutorial: como acessar a internet com o Arduino GSM Shield SIM900**. 2015 [online]. Disponível na internet via URL:<<https://www.arduinoecia.com.br/2015/11/acessar-internet-arduino-gsm-shield-sim-900.html> >. Acessado dia 22/11/2017.

BAUMAN, Z. **Comunidade: a busca por segurança no mundo atual**. Rio de Janeiro: Jorge Zahar Ed., 2003.

BECKER, H. S. **Uma teoria da ação coletiva**. Rio de Janeiro: Zahar, 1997. 227 p.

BERNAL, P. S. M. **Comunicações Móveis: Tecnologias e Aplicações**. São Paulo. Editora Érica. 2002. ISBN: 85-7194-898-4.

CALDEIRA, T. **Cidade de Muros: crime, segregação e cidadania em São Paulo**. São Paulo: ed. 34 / Edusp, 2000.

CERQUEIRA, D.; LOBÃO, W. e CARVALHO, A. X. **O jogo dos sete mitos e a miséria da segurança pública no Brasil**. Texto para discussão n. 1144. Rio de Janeiro: IPEA, 2005.

CERQUEIRA, D; CARVALHO, A.; LOBÃO, W.; RODRIGUES, R. **Análise dos custos e consequências da violência no Brasil**. Texto para discussão n. 1284. Brasília: IPEA 2007.

DUBET, F. **Declive de la institución escolar y conflictos de principios**. Debates de Educación, Barcelona, Edició: Fundació Jaume Bofí II, p 1-24, 2010.

SOUZA, E. R.; LIMA, M. L. C. **Panorama da violência urbana no Brasil e suas capitais**. Ciênc. saúde coletiva, Rio de Janeiro, v. 11, supl. p. 1211-1222, 2006 .

FERRAZ, L. G. C.; GARCIA, R. R. G.; NUNES, D. (2012). **Estudo comparativo entre sistemas propostos para 4G: LTE e WiMAX móvel**. Anais do Congresso de iniciação científica do INATEL - INCITEL 2012. Pg. 73.

FIORESE, V. **Wireless: Introdução às Redes de Telecomunicações Móveis Celulares**. Rio de Janeiro: Brasport, 2005. 336 p. ISBN 85-7452-214-7.

GARG, V. K.; SMOLIK, K.; WILKES, J. E. **Applications of CDMA in Wireless / Personal Communications**. Upper Saddle River. Prentice-Hall PTR. 1997.

GRUBER, V. **Sistemas de monitoramento remoto baseado em rede de celular GSM/GPRS para gerenciamento de desgaste de pastilha de freio e vibração de torre em aerogeradores**. 2007, 94 p. Dissertação (Mestrado em Engenharia) – Universidade Federal do Rio Grande do Sul.

HÉLIO, W.; MICHEL, D. Y. **Telecomunicações – Princípios e Tendências**. São Paulo. Editora Érica. 1997.

INSTITUTE FOR GLOBAL AND PEACE. **Measuring peace in a complex world**. Global Peace Index, 2017.

KHAN, B. **Android JSON Parsing – Retrieve From MySQL Database**, 2015. [online]. Disponível na internet via URL:< <https://www.simplifiedcoding.net/android-json-parsing-tutorial/> >. Acessado dia 04/12/2017.

KOSTAL. **Manual do usuário e de instalação Linha de alarmes: K150/K350/K550**. KOSTAL Eletromecânica Ltda, 2017.

LANDONI, B. **The GSM/GPRS Shield and GPS Shield: some http connection examples**. 2011. [online]. Disponível na internet via URL:< <https://www.open-electronics.org/the-gsmgprs-gps-shield-some-http-connections-examples/> >. Acessado dia 03/12/2017.

LECHETA, R. R. **Google Android : Aprenda a criar aplicações para dispositivos móveis com Android SDK**. São Paulo: Novatec Editora, 2009.

MCROBERTS, M. **Arduino Básico**. São Paulo: Novatec, 2011. 443 p.

MONK, S. **Programação com Arduino: começando com sketches**. 2. ed. Porto Alegre: Bookman, 2017. 200p

MOTA, T. A. **Redes 3G e a evolução para as redes 4G**. 2009. [online]. Disponível na internet via URL: < <http://www.teleco.com.br/pdfs/tutoriale4g.pdf> >. Acessado dia 20/10/2017.

OLIVEIRA, C. C. **Estudo sobre a utilização de mapas em android para itinerários de ônibus**. 2012. 44 p. Monografia de Especialização – Universidade Tecnológica Federal do Paraná, Departamento Acadêmico de Informática. Curitiba.

PAIN & SMITH, R.; SMITH, S. J. (2008). **Fear: Critical Geopolitics and everyday life**. Hampshire: Ashgate.

PUHALA, K. **CarLock Alerts YOU (Not the Whole Neighborhood) If Your Alarm Goes Off. 2014.** [online]. Disponível na internet via URL: < <https://www.brit.co/carlock-security-system/> >. Acessado em 07/12/2017.

ROSA, B. S. **Segurança pública e política de prevenção inteligente à criminalidade: uma proposta para conter furtos de veículos no município de Tijucas/SC.** 2010. 76 p. Dissertação (Graduação em Direito) - Universidade do Vale do Itajaí, Centro de Ciências Sociais e Jurídicas, *campus* de Tijucas.

SAPORI, L. F. e WANDERLEY, C. B. **A relação entre o desemprego e a violência na sociedade brasileira: entre o mito e a realidade.** Cadernos de ADENAUER II. São Paulo: Fundação Konrad Adenauer, 2001.

SCHWEITZER, C. M.; et al. **Tecnologias de Redes sem fio: WPANs, WLANs e WMANs, Desafios de Segurança, Vulnerabilidade e Soluções.** Universidade de São Paulo – USP. São Paulo. 2006.

SINESP. **Relatório Consolidado de Ocorrências de roubos de veículo registrados pela Polícia Civil.** [online]. Disponível na internet via URL: < <https://www.sinesp.gov.br/estatisticas-publicas> >. Acessado em 16/10/2017.

SSP-RS. **Indicadores criminais geral – 2011-2016.** [online]. Disponível na internet via URL: < <http://www.ssp.rs.gov.br/indicadores-criminais> >. Acessado em 18/10/2017.

SVERZUT, J. U. **Redes Gsm, Gprs, Edge e Umts - Evolução a Caminho da Terceira Geração (3g).** São Paulo. Editora Érica. 2005.

TELECO. **3G no Brasil.** 2017. [online]. Disponível na internet via URL: < http://www.teleco.com.br/3g_brasil.asp >. Acessado em 20/10/2017.

TELECO. **4G: 4ª Geração de Celular no Brasil.** 2017. [online]. Disponível na internet via URL: < http://www.teleco.com.br/4g_cobertura.asp >. Acessado em 20/10/2017.

TELECO. **Celulares 4G em Ago/17.** 2017. [online]. Disponível na internet via URL: < http://www.teleco.com.br/4g_brasil.asp >. Acessado em 20/10/2017.

TENEMBAUM, A. S. **Redes de Computadores.** Rio de Janeiro. Editora Campus. 1997.

TUDE, E. **Tutoriais Telefonia Celular.** 2003. [online]. Disponível na internet via URL: < <http://www.teleco.com.br/tutoriais/tutorialtdma/default.asp> >. Acessado em 19/10/2017.

VAZ, P.; CAVALCANTI, M.; CARVALHO, C. S.; JULIÃO, L. **Pobreza e Risco: a imagem da favela no noticiário de crime.** In: XIV Encontro Anual da Compós, 2005, Niterói.

VIPER. **SMARTSTART**. [online]. Disponível na internet via URL: < <https://www.viper.com/smartstart/modules.aspx> >. Acessado em 07/12/2017.

APÊNDICE A - CÓDIGOS

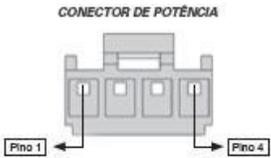
Conforme mencionado na metodologia do presente trabalho, devido à enorme quantidade de linhas de código utilizadas, teve-se que suprir os códigos dos textos e disponibilizá-los aos interessados através de um link. O link que contém os códigos é disponibilizado abaixo:

<https://drive.google.com/drive/folders/14d3On1C0XdKNbVIIYekZpXVvkzYSISWSa?usp=sharing>

APÊNDICE B - IMAGENS

A seguir são mostradas as imagens referenciadas durante o texto. A Figura B.1 mostra o diagrama de instalação presente no manual de alarmes Kostal.

4. DIAGRAMA DE INSTALAÇÃO



CONECTOR DE POTÊNCIA



CONECTOR DE SINAL

Referência - Saída dos fios

PINO	FIOS DO ALARME	APLICAÇÃO	INFORMAÇÕES IMPORTANTES
1	PTIAZ		Sinal modulado para a Sirene
2	PTVVD		Sinal positivo da Set
3	PTVVD		Sinal positivo da Set
4	VM		Alimentação positiva da bateria - Linha 30
1			Vazio
2			Vazio
3	VDIAM		Monitoramento de porta aberta através da luz de teto
4	LJ		NÃO USAR
6	BR		Monitoramento de lanternas ligadas (necessário o uso do ultrassom master)
6	VMPT		Sinal para destravamento - Negativo
7	VD		Interruptor de porta - Entrada de sinal negativo
8	BRVVD		Saída auxiliar 1 - Abertura dos vidros
9	BR		Interruptor de capô e porta-malas - Entrada de sinal negativo
10	CZ		Saída Auxiliar 2 - Corte de Partida
11	MRAZ		NÃO USAR
12	VDIPT		Saída para levantador de vidro
13	AZ		Pós-ignição
14	RX		Saída Auxiliar 5 - Rádio
15	BRVAZ		Sinal de sensor de velocidade
16	CZIPT		Saída Auxiliar 3 - Abertura de Porta-malas
18	CZVAZ		Saída Auxiliar 4 - Temporizador de Faróis
20	AM		NÃO USAR
22	VMUAM		Sinal para travamento - Negativo
24	PT		Alimentação negativa - Linha 31
26	PT		Chave Master
17	AMPT		Sinal de saída do Transmissor do Ultrassom
19	PT		Alimentação Negativa do Ultrassom
21	VM		Saída do LED
23	AMVM		Sinal de entrada do Receptor do Ultrassom
25	PT		Alimentação Negativa do Ultrassom




MANUAL DO USUÁRIO E DE INSTALAÇÃO
LINHA DE ALARMES



Alarmes: K150 | K350 | K550

Figura B.1 - Diagrama de Instalação alarme Kostal K-150.
Fonte: Manual Alarmes KOSTAL.

As figuras B.2 e B.3 ilustram os diagramas elétricos projetados.

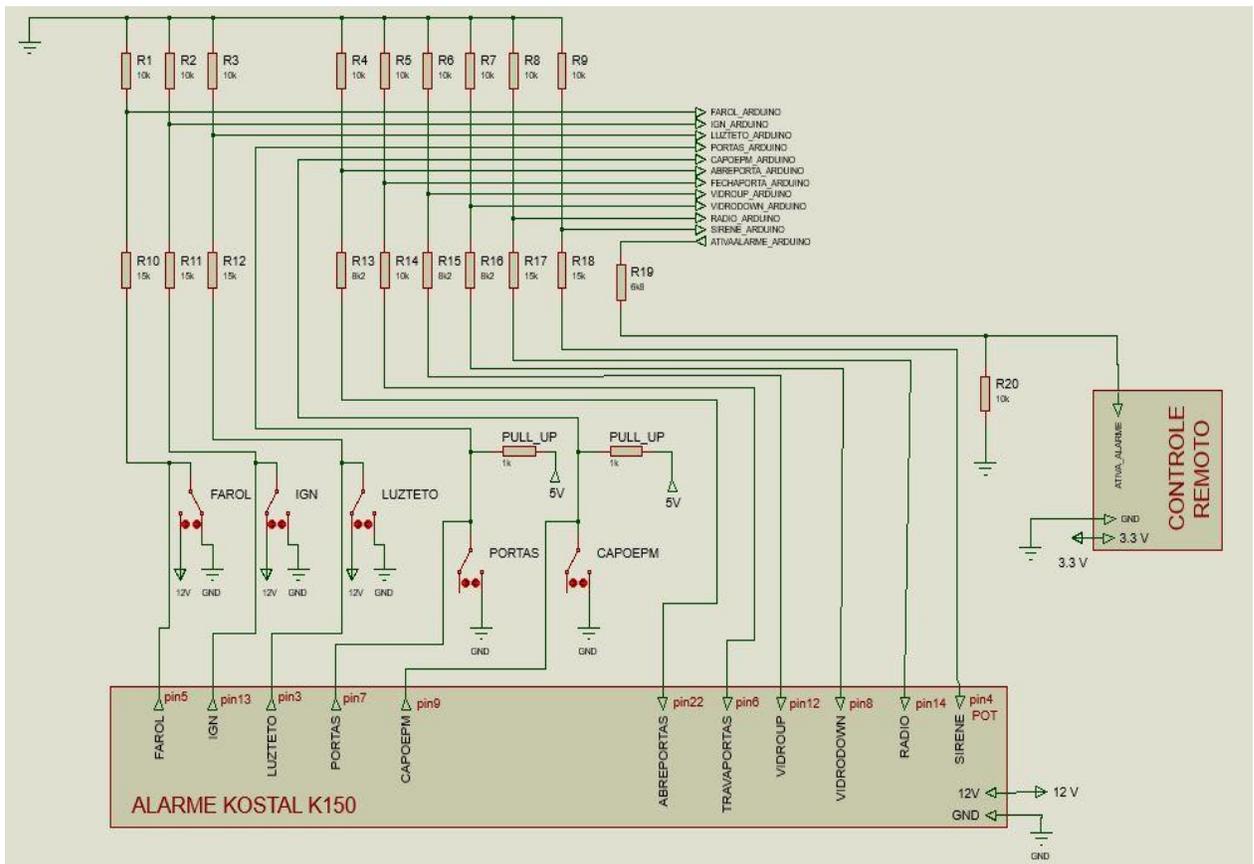


Figura B.2 - Esquemático elétrico principal da protoboard.

Fonte: Autor.

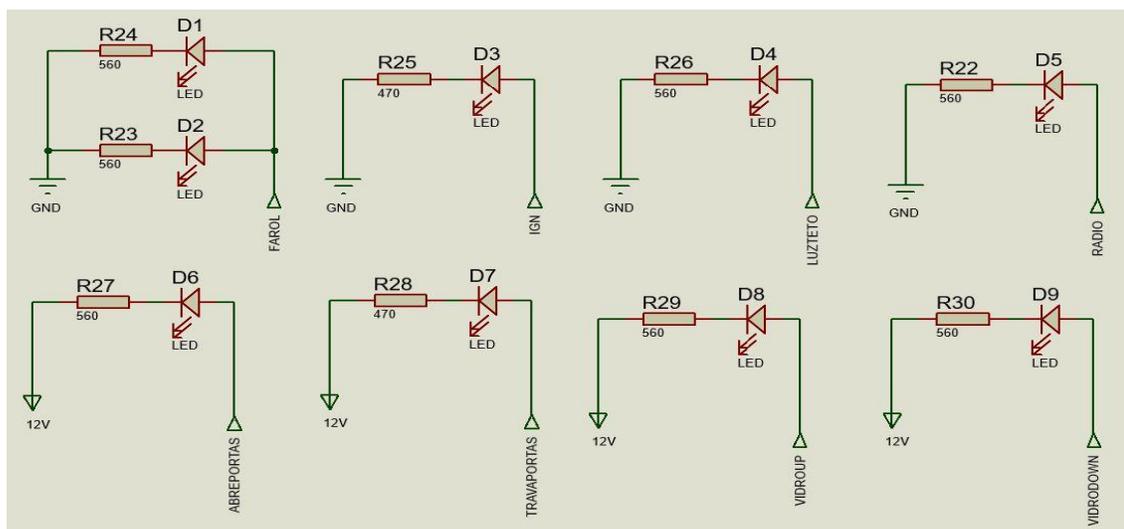


Figura B.3 - Esquemático elétrico de visualização da protoboard.

Fonte: Autor.

A montagem do circuito que simula o veículo com o alarme é mostrada na Figura B.4.

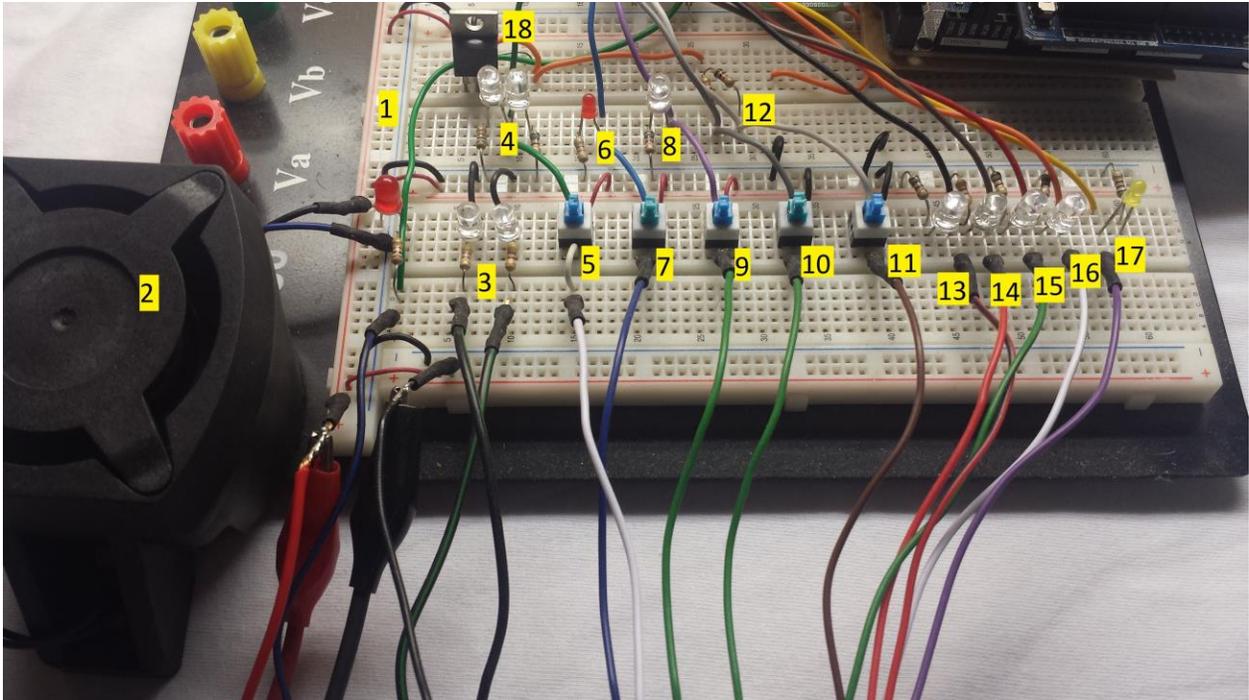


Figura B.4 - Montagem protoboard do modelo da planta.

Fonte: Autor.

Itens referenciados da Figura B.4:

- 1 – Barramentos com a alimentação de 12 VCC;
- 2 – Sirene (utilizado um LEDs para visualização, além do som da mesma);
- 3 – LEDs simulando os pisca-alertas do veículo;
- 4 – LEDs simulando os faróis do veículo;
- 5 – Botão simulando a chave do painel interior para ligar faróis do veículo;
- 6 – LED para indicar ignição ligada;
- 7 – Botão simulando a ativação da ignição;
- 8 – LED simulando a luz de teto do interior do veículo;
- 9 – Botão simulando o sinal elétrico que é levado a luz de teto;
- 10 – Botão simulando as chaves de contato presentes em cada porta;
- 11 – Botão simulando as chaves de contato presentes no capô e porta-malas;
- 12 – Resistores de *Pull-up* utilizados no circuito de PORTAS e CAPOEPM;
- 13 – LED indicador sinal de destravamento de portas;
- 14 – LED indicador para sinal de travamento de portas;
- 15 – LED indicador para sinal de levante de vidros;
- 16 – LED indicador para sinal de abaixamento de vidros;
- 17 – LED indicador para rádio ligado;
- 18 – Regulador de tensão 7805 para e utilizar nos resistores de *Pull-up*;

A montagem do circuito que implementa a central é mostrada na Figura B.5.

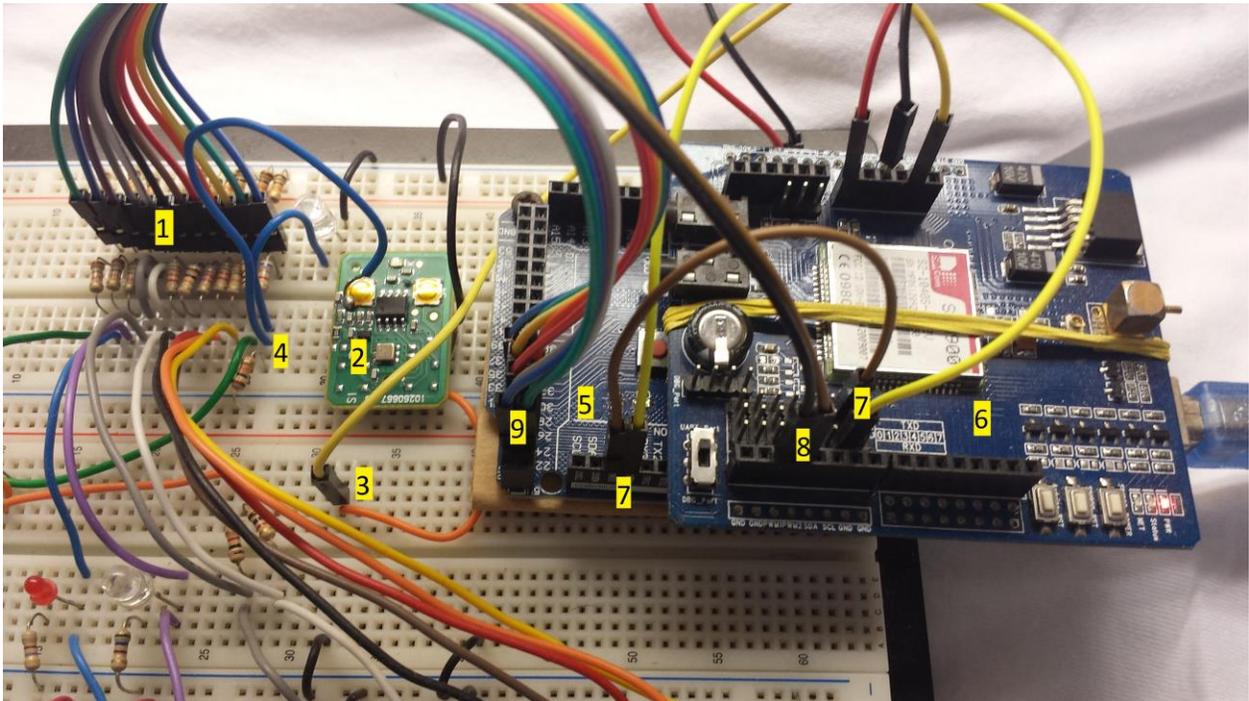


Figura B.5 - Montagem central de monitoramento e comunicação.

Fonte: Autor.

Itens referenciados da Figura B.5:

- 1 – Divisores de tensão;
- 2 – Controle remoto utilizado para o acionamento pela central;
- 3 – Alimentação de 3.3 V do controle remoto;
- 4 – Condutor do sinal ATIVA_ALARME da Figura 9;
- 5 – Arduino MEGA 2560;
- 6 – *Shield* GSM/GPRS IcomSat v1.1;
- 7 – Pinos utilizados para a comunicação *Software Serial*;
- 8 – Pinos utilizados para as interrupções externas;
- 9 – Pinos utilizados para o restante das variáveis monitoradas;

Os circuitos mostrados nas figuras B.4 e B.5 foram montados na mesma *protoboard*, deixando de fora apenas o alarme, que ficava disposto próximo à *protoboard* e a fonte de alimentação, pois o chicote de condutores possuía um comprimento suficiente para deixá-lo separado do resto dos circuitos.

APÊNDICE C - TABELAS

A tabela C.1 define as nomenclaturas utilizadas e os níveis de tensão medidos para cada.

Tabela C.1 - Definições de nomenclaturas e níveis de tensão medidos.

Fonte: Autor.

Pino	Informações importantes	Nomenclatura utilizada	Níveis de tensão medidos
3	Monitoramento das portas abertas através da luz de teto	LUZ_TETO	0 V – luz apagada 12 V – luz acesa
5	Monitoramento das lanternas ligadas	FAROL	0 V – faróis apagados 12 V – faróis ligados
6	Sinal de destravamento – Saída de sinal negativo	TRAVAPORTA	12 V – nada ocorre Pulso de 0 V – destrava portas
7	Interruptor de portas – Entrada de sinal negativo	PORTAS	0 V – portas abertas Flutuante ² – portas fechadas
8	Saída auxiliar 1 – Abertura de vidros	VIDRODOWN	0 V – vidro sobe 9.3 V – vidro parado
9	Interruptor de capô e porta-malas – Entrada de sinal negativo	CAPOEPM	0 V – capô/porta-malas aberto Flutuante ² - fechados
12	Saída para levantador de vidros	VIDROUP	0 V – vidro desce 9.2 V – vidro parado
13	Pós – ignição	IGN	0 V – ignição desativada 12 V – ignição ativada
14	Saída auxiliar 5 - Rádio	RADIO	0 V – rádio deligado 12 V – rádio ligado
22	Sinal de travamento – Saída negativo	ABREPORTA	9.1 V – nada ocorre Pulso de 0 V – trava portas
4 POT	Sinal modulado da sirene	SIRENE	5.5 V RMS – sirene emite som 0 V – nada ocorre

² Flutuante significa circuito aberto, sem conexão com nenhum nível de tensão.

A tabela C.2 ilustra a tabela utilizada no banco de dados do servidor criado.

Tabela C.2 - Tabela utilizada no banco de dados do servidor.

Fonte: Autor.

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
1	timeStamp	timestamp		on update CURRENT_TIMESTAMP	Não	CURRENT_TIMESTAMP		ON UPDATE CURRENT_TIMESTAMP	Muda Elimina Mais
2	status	text	utf8_unicode_ci		Não	None			Muda Elimina Mais