

FACULDADE SENAC BLUMENAU  
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Marcelo Kafels Eftting  
Ricardo Godoz  
Roberto Umbelino

**SISTEMA DE SEGURANÇA DOMICILIAR UTILIZANDO INTERNET DAS COISAS  
E COMPUTAÇÃO EM NUVEM**

Blumenau  
2018

Marcelo Kafels Effting

Ricardo Godoz

Roberto Umbelino

**SISTEMA DE SEGURANÇA DOMICILIAR UTILIZANDO INTERNET DAS COISAS  
E COMPUTAÇÃO EM NUVEM**

Trabalho de conclusão de semestre, apresentado à Faculdade de Tecnologia Senac Blumenau, como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientadores: Cláudio Ratke Msc.

Fabiano Oss Esp.

Lariana Luy Peixoto Msc.

Blumenau

2018

Marcelo Kafels Eftting

Ricardo Godoz

Roberto Umbelino

**SISTEMA DE SEGURANÇA DOMICILIAR UTILIZANDO INTERNET DAS COISAS  
E COMPUTAÇÃO EM NUVEM**

Trabalho apresentado à Faculdade de  
Tecnologia Senac Blumenau como requisito  
parcial para obtenção do título de Tecnólogo em  
Análise e Desenvolvimento de Sistemas.

---

Cláudio Ratke, Msc.

---

Fabiano Oss, Esp.

---

Helcio Hermes Hoffmann, Msc.

---

Lariana Luy Peixoto, Msc.

---

Pedro Edmundo Floriani, Msc.

---

Tércio Gaspar, Msc.

Blumenau, 23 de junho de 2018

## RESUMO

A sociedade vive em um período onde a segurança pública é imprescindível, sendo ela um dever do Estado para com seus cidadãos. Porém, com os altos índices de criminalidade no Brasil a população começa a se questionar se realmente estão seguros e acabam por buscar novas maneiras para garantir a própria segurança. Portanto o principal objetivo é contribuir com a segurança patrimonial, oferecendo um sistema de segurança domiciliar. A solução se baseia no uso de dispositivos Arduino, os quais serão sensores de movimentação e sirenes para instalação em domicílio que irão capturar dados e enviar para uma API *web service*, que terá a função de processar, armazenar e disponibilizar estes dados. Foi criada uma interface *mobile* conectada a API, esta interface possibilita acessar as informações e realizar o gerenciamento da rede de dispositivos de segurança.

**Palavras-chave:** Segurança. Nuvem. IoT. Arduino.

## **ABSTRACT**

The society lives in a period where public safety is essential, being a duty of the State towards its citizens. However, with high crime rates in Brazil the population begins to wonder if they are really safe and end up looking for new ways to ensure their own safety. Therefore, the main objective is to contribute to property security by offering a home security system. The solution is based on the use of Arduino devices, which will be motion sensors and sirens for home installation that will capture data and send to a web service API, which will have the function of processing, storing and making this data available. A mobile interface connected to the API was created, this interface allows access to the information and to manage the network of security devices.

**Keywords:** Security. Cloud. IoT. Arduino.

## LISTA DE FIGURAS

Figura 1 - Órgãos públicos que exercem a segurança pública.....	16
Figura 2 - Arduino Uno .....	18
Figura 3 - Exemplo de informações e produtos na nuvem .....	19
Figura 4 - Abrangência do modelo SaaS .....	20
Figura 5 - Imagem explicativa do web service.....	21
Figura 6 - Imagem explicativa do JSON.....	23
Figura 7 - Visão geral do sistema.....	29
Figura 8 - Módulo ESP8266 para se conectar com a Wi-Fi .....	30
Figura 9 – Conjunto de pilhas que irá energizar os dispositivos.....	31
Figura 10 - Botão WPS dos dispositivos .....	31
Figura 11 - Módulo de detecção de movimento .....	32
Figura 12 - Modelo do dispositivo de sensor de movimento .....	33
Figura 13 - Esquema do dispositivo de sensor de movimento .....	33
Figura 14 - Dispositivo de sensor de movimento montado.....	34
Figura 15 - Fluxo para enviar movimento detectado .....	34
Figura 16 – Sirene do dispositivo de sirene .....	35
Figura 17 - Modelo do dispositivo de sirene.....	35
Figura 18 - Esquema do dispositivo de sirene.....	36
Figura 19 - Dispositivo de sirene montado .....	36
Figura 20 - Fluxo para soar a sirene .....	37
Figura 21 - MER geral do projeto .....	38
Figura 22 - Fluxo para autenticar um usuário.....	39
Figura 23 - Corpo da requisição enviada para o web service .....	39
Figura 24 - Corpo do retorno da requisição do web service em caso de sucesso ....	40
Figura 25 - Corpo do retorno da requisição do web service em caso de erro .....	40
Figura 26 - Rota para cadastrar um novo usuário .....	41
Figura 27 - Exemplo de teste automatizado com o Postman .....	42
Figura 28 - Tela de Login do aplicativo .....	43
Figura 29 - Tela de Cadastro do aplicativo.....	44
Figura 30 - Lista de grupos do usuário no aplicativo .....	45
Figura 31 - Criando um grupo no aplicativo .....	46
Figura 32 - Novo grupo de dispositivo “Casa do campo” .....	47

Figura 33 - Editar/Remover um grupo do usuário no aplicativo.....	48
Figura 34 - Lista de dispositivos vinculados ao grupo .....	49
Figura 35 - Vinculando um dispositivo ao grupo.....	50
Figura 36 - Removendo um dispositivo do grupo .....	51
Figura 37 - Opção de ligar/parar alarme e convidar novas pessoas .....	52
Figura 38 - Convidar uma nova pessoa ao grupo .....	53
Figura 39 - Aceitando/Recusando o convite de um grupo.....	54
Figura 40 - Grupo de dispositivo convidado .....	55
Figura 41 - Notificação .....	56

## LISTA DE GRÁFICOS

Gráfico 1 - Número absoluto de roubos no Brasil.....	13
Gráfico 2 - Ocorrências de furto em Santa Catarina .....	14
Gráfico 3 - Registros de roubos em Blumenau.....	14
Gráfico 4 - Pesquisa de Internet das Coisas .....	24

## LISTA DE QUADROS

Quadro 1 - Histórias de usuários.....	26
Quadro 2 – <i>Spikes Web Service</i> .....	27
Quadro 3 - <i>Spikes</i> Dispositivos .....	27
Quadro 4 - <i>Spikes</i> Aplicativo .....	28

# SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>12</b>
1.1 Objetivos .....	12
<b>1.1.1 Objetivo Geral</b> .....	<b>13</b>
<b>1.1.2 Objetivos Específicos</b> .....	<b>13</b>
1.2 Justificativa .....	13
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>15</b>
2.1 Segurança .....	15
<b>2.1.1 Segurança pública</b> .....	<b>15</b>
<b>2.1.2 Segurança da informação</b> .....	<b>16</b>
2.2 Arduino .....	17
2.3 Nuvem .....	19
2.4 REST .....	21
<b>2.4.1 HTTP e HTTPS</b> .....	<b>21</b>
<b>2.4.2 Métodos de requisições</b> .....	<b>22</b>
<b>2.4.3 JSON</b> .....	<b>22</b>
2.5 Internet das coisas .....	23
<b>3 METODOLOGIA</b> .....	<b>25</b>
<b>4 DESENVOLVIMENTO</b> .....	<b>26</b>
4.1 Requisitos do Usuário .....	26
4.2 Requisitos do Sistema ( <i>Spikes</i> ) .....	27
4.3 Descrição do Projeto .....	28
4.4 Tecnologias .....	29
<b>4.4.1 Arduino</b> .....	<b>30</b>
<b>4.4.2 Web Service</b> .....	<b>37</b>
<b>4.4.3 Aplicativo para Android</b> .....	<b>42</b>
<b>4.4.4 Qualidade de software</b> .....	<b>57</b>
<b>4.4.5 Infraestrutura do Projeto</b> .....	<b>57</b>
<b>4.4.6 Estimativa de custo</b> .....	<b>57</b>
<b>4.4.7 Resultado</b> .....	<b>58</b>
<b>5 CONSIDERAÇÕES FINAIS</b> .....	<b>60</b>
<b>REFERÊNCIAS</b> .....	<b>62</b>
<b>APÊNDICE A – HISTÓRIAS DE USUÁRIO</b> .....	<b>65</b>

<b>APÊNDICE B – ESPECIFICAÇÕES DO BANCO DE DADOS .....</b>	<b>68</b>
--	-----------

## 1 INTRODUÇÃO

Segundo indicadores disponibilizados pela Secretaria de Segurança Pública de Santa Catarina (SSP/SC), na cidade Blumenau, em média, são registrados 356 furtos por mês. Esse número vem aumentando nos últimos anos e é possível ver um aumento de 11% de 2015 a 2017 (SSP/SC, 2018). Por conta desses problemas, os brasileiros passaram a procurar maneiras alternativas de proteger seus patrimônios.

Nota-se que muitas propriedades possuem um sistema de segurança de alarme para prevenção de furtos e roubos. Algumas pessoas não possuem o conhecimento tecnológico necessário para realizar a instalação de um sistema de segurança por conta própria em seu domicílio e acabam sendo obrigadas a contratar empresas especializadas nessa área de atuação.

Sendo assim, este trabalho traz uma proposta de oferecer um meio simples e funcional para quem deseja montar o próprio sistema de segurança através de dispositivos Arduino (sensor de movimento e sirene) conectados à nuvem, que se trata de uma abstração da infraestrutura da Internet (RUSCHEL; ZANOTTO; MOTA, 2010), e um aplicativo para celular que possibilita controlar e configurar o grupo desses dispositivos.

O conceito de Internet das Coisas (*Internet of Things*, IoT) foi aplicado com a utilização desses dispositivos. Segundo Silva (2017), é possível observar que IoT é um conceito que visa conectar dispositivos do mundo real com o mundo digital, sendo assim os dispositivos de segurança conectados diretamente com a Internet fazem parte da Internet das Coisas.

Para montagem dos dispositivos de segurança, será feito o uso do Arduino, uma plataforma de desenvolvimento de dispositivos eletrônicos com *hardware* livre que pode facilmente ser programada através de um computador e que possui uma grande quantidade de informações disponíveis na Internet.

### 1.1 Objetivos

Neste tópico serão apresentados os objetivos gerais do projeto, junto aos objetivos específicos.

### 1.1.1 Objetivo Geral

Desenvolver um sistema de segurança domiciliar, utilizando o Arduino para IoT, armazenando os dados na Nuvem e oferecendo um serviço SaaS (Software como um serviço).

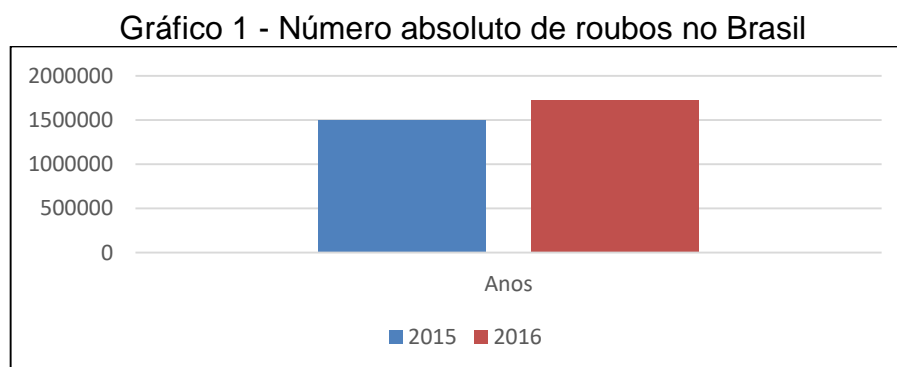
### 1.1.2 Objetivos Específicos

A seguir são descritos os objetivos específicos deste projeto:

- a) Elaborar um aplicativo para realizar o gerenciamento do sistema da rede de dispositivos de segurança do usuário;
- b) Elaborar um *web service* para receber, armazenar e disponibilizar dados
- c) Elaborar uma plataforma de detector de movimento para coletar dados de movimento e enviar ao *web service*;
- d) Elaborar uma plataforma de sirene para buscar informações do *web service*.

## 1.2 Justificativa

De acordo com o artigo 144 da Constituição da República Federativa do Brasil, o Estado tem o dever de oferecer segurança para a população e aos patrimônios, tanto do Estado quanto do cidadão (BRASIL, 1988), no entanto o índice de criminalidade no Brasil vem crescendo nos últimos anos. Isso pode ser visto no Anuário Brasileiro de Segurança Pública que indicou um aumento de 13,9% na quantidade de roubos no ano de 2015 para 2016 (FBSP, 2017).

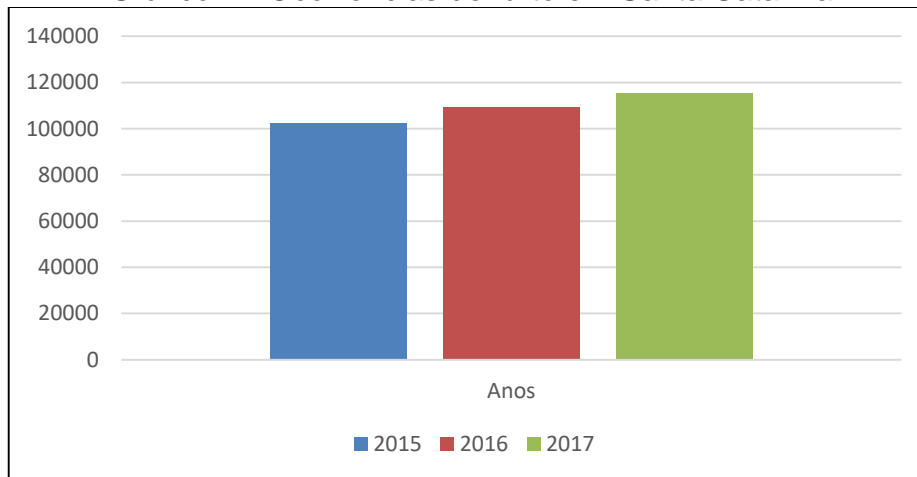


Fonte: Modificado pelos autores com base no Anuário Brasileiro de Segurança Pública (FBSP, 2017)

A Secretaria de Segurança Pública de Santa Catarina fornece dados de crimes ocorridos no estado. Analisando estes dados é possível perceber um aumento

de mais de 10% na quantidade geral de registros de ocorrência de furto de 2015 até 2017 (SSP/SC, 2018).

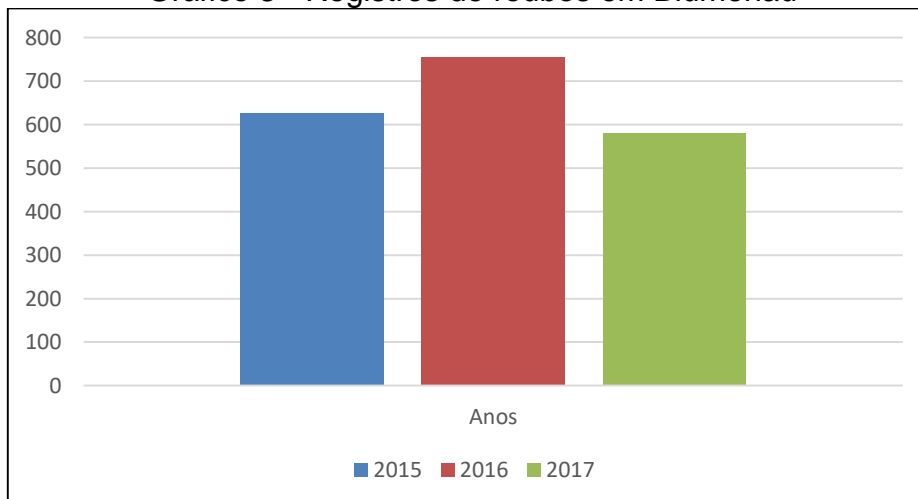
Gráfico 2 - Ocorrências de furto em Santa Catarina



Fonte: Modificado pelos autores com base na Secretaria de Segurança Pública de Santa Catarina (FBSP, 2018)

É possível verificar dados mais localizados, onde em média, 654 registros de roubos por ano na cidade de Blumenau, mais de um roubo ao dia.

Gráfico 3 - Registros de roubos em Blumenau



Fonte: Modificado pelos autores com base na Secretaria de Segurança Pública de Santa Catarina (FBSP, 2018)

Sendo assim, há uma oportunidade de negócio no ramo de segurança pois entende-se que há uma necessidade de inovação na área e propõe-se utilizar tecnologias atuais como IoT, Nuvem e Android para desenvolver um sistema de segurança domiciliar.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será fundamentado os conceitos gerais da pesquisa.

### 2.1 Segurança

De acordo com Cruz (2008), o tema segurança é algo muito abrangente e para se ter um entendimento melhor do que se trata a segurança ou até mesmo a segurança pública, é necessário o conhecimento de ordem pública que faz referência à tranquilidade e à salubridade pública.

A ordem pública é mais fácil de ser sentida do que definida, mesmo porque ela varia de entendimento no tempo e no espaço. Aliás, nessa última hipótese, pode variar, inclusive dentro de um determinado país. Mas sentir-se-á a ordem pública segundo critérios de ordem superior, políticos, econômicos, morais e, até mesmo, religiosos. A ordem pública não deixa de ser uma situação de legalidade e moralidade normal, apurada por quem tenha competência para isso sentir e valorar. A ordem pública, em outras palavras, existirá onde estiver ausente a desordem, isto é, os atos de violência, de que espécie for, contra as pessoas, bens ou o próprio Estado. A ordem pública não é figura jurídica, embora se origine e tenha a sua existência formal (CRUZ, 2008, p.3, apud LAZZARINI, 1986).

Conforme foi citado por Cruz (2008, apud LAZZARINI, 1986) a ordem pública tem como finalidade estabelecer e manter uma ordem social, sendo ela aplicada para as pessoas e bens. Então quando é falado que algo é seguro ou que possui segurança, estará dizendo que recursos estão sendo utilizados para uma possível proteção de uma pessoa ou objeto.

#### 2.1.1 Segurança pública

Para Cruz (2008, apud MOREIRA NETO, 1987) a segurança pública se trata de um conjunto de regras, sendo elas regras formais e coativas que acabam regulando o escopo da sociedade para que seja possível estabelecer um clima de convivência pacífico e harmonioso.

Como foi citado anteriormente sobre o artigo 144 da constituição, o Estado tem o dever de manter sua sociedade segura e isso inclui, as pessoas e os patrimônios, tanto pessoal quanto do Estado. Para realizar essa ação o Estado investe em órgãos públicos que estão representados na Figura 1. Esses órgãos possuem à responsabilidade primordial de zelar pela efetiva aplicação de medidas que garantam a segurança dos cidadãos (CRUZ, 2008).

Figura 1 - Órgãos públicos que exercem a segurança pública



Fonte: Elaborado pelos autores com base no artigo 144 da Constituição da República Federativa do Brasil (BRASIL, 1988)

Segurança de patrimônio, que é uma vertente da segurança pública, se refere ao ato ou efeito de proteger os bens pessoais de um indivíduo, são os recursos que o Estado utiliza para melhorar a segurança dos patrimônios.

### 2.1.2 Segurança da informação

Outro tipo de segurança que deve ser priorizada é a segurança da informação, pois existem informações que são pessoais, e esse tipo de informação não podem ser acessadas por ninguém além da pessoa detentora da informação. Para entender sobre isso primeiro, deve-se entender o que é informação.

A Informação em Ciência e Tecnologia (ICT) é constituída de elementos simbólicos utilizados para comunicar o conhecimento científico e técnico, independente de seu caráter (numérico, textual, icônico etc.), dos suportes materiais, da forma de apresentação. Refere-se tanto à substância ou conteúdo dos documentos quanto à sua existência material (AGUIAR, 1991, p.2).

Analisando esta definição de Aguiar (1991) e analisando um sistema de segurança domiciliar pode-se entender que tipos de informações podem ser: Imagens, textos, valores, números entre vários outros dados que podem ser analisados.

“Segurança da informação está ligada à proteção de informações e dados com o intuito de preservar o valor que possuem para uma organização ou indivíduo”

(NASCIMENTO, 2011, p.1). Sendo assim, os *softwares* que armazenam dados pessoais, devem ter uma grande preocupação com a segurança da informação, pois uma vez que essas informações sejam acessíveis por outras pessoas, acabam tornando o sistema uma base de informações pessoais que podem ser utilizados por pessoas desonestas, e a situação pode ser ainda pior se houver o uso do conceito de Internet das Coisas, onde dispositivos estarão coletando dados de locais físicos, como por exemplo o domicílio da pessoa.

## 2.2 Arduino

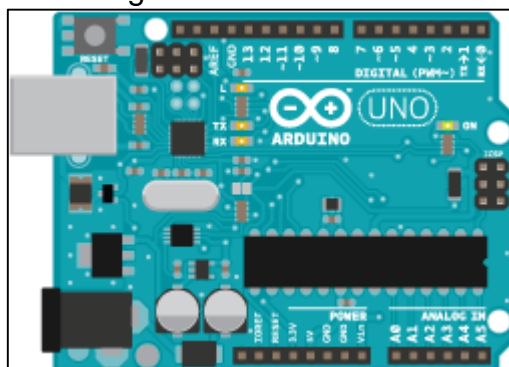
A respeito do Arduino, “o conceito Arduino surgiu na Itália, em 2005, com objetivo de criar um dispositivo de controle para protótipos construídos de forma menos dispendiosa do que outras soluções disponíveis” (FONSECA; VEGA, 2014, p.3), ou seja, o Arduino é um facilitador para construção de dispositivos de controle.

Pode-se entender que Arduino “é uma plataforma de computação física de fonte aberta, com base em uma placa simples de entrada/saída” (BANZI, 2011, p.1). Segundo Banzi (2011), o Arduino também pode ser utilizado para objetos interativos independentes, podendo realizar comunicações com *softwares* de um computador. Seguindo o pensamento semelhante ao de Banzi, o Souza (2011) também diz que a placa de Arduino é muito independente e versátil tendo assim um microcontrolador que é capaz de potencializar funções.

A placa Arduino é baseada num microcontrolador muito versátil que potencializa suas funções para além de uma simples interface passiva de aquisição de dados, podendo operar sozinha no controle de vários dispositivos e tendo assim aplicações em instrumentação embarcada e robótica. (SOUZA, et al, 2011, p.1).

Atualmente, o Arduino possui diversas placas de microcontroladores. Um exemplo é a placa Uno, que está representada na Figura 2, porém existem diversas outras placas como: Mega 2560, Due, Mega ADK, Nano, entre outras.

Figura 2 - Arduino Uno



Fonte: Site do Arduino (ARDUINO, 2018)

A programação no Arduino é realizada através de sua IDE (Ambiente Integral de Desenvolvimento) padrão utilizando a linguagem de programação C ou C++ para compilar o código fonte e transferir os dados para a plataforma através do cabo USB. Atualmente, o Arduino possui muitos módulos independentes, como por exemplo: câmera, sensor de movimento, sensor de presença, interruptores, receptores GPS, módulos Ethernet, entre inúmeros outros módulos.

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. (MCROBERTS, 2011, p.20).

Utilizando a IDE para criar um programa de computador, que segundo McRoberts “é um conjunto de instruções passo a passo, das quais você faz o *upload* para o Arduino. Seu Arduino, então, executará essas instruções, interagindo com o que estiver conectado a ele” (MCROBERTS, 2011, p.24). Na comunidade do Arduino, estes programas de computadores são conhecidos como *sketches* (esboço).

O que difere as placas Arduino das demais no mercado, segundo Banzi (2011), são os seguintes fatores:

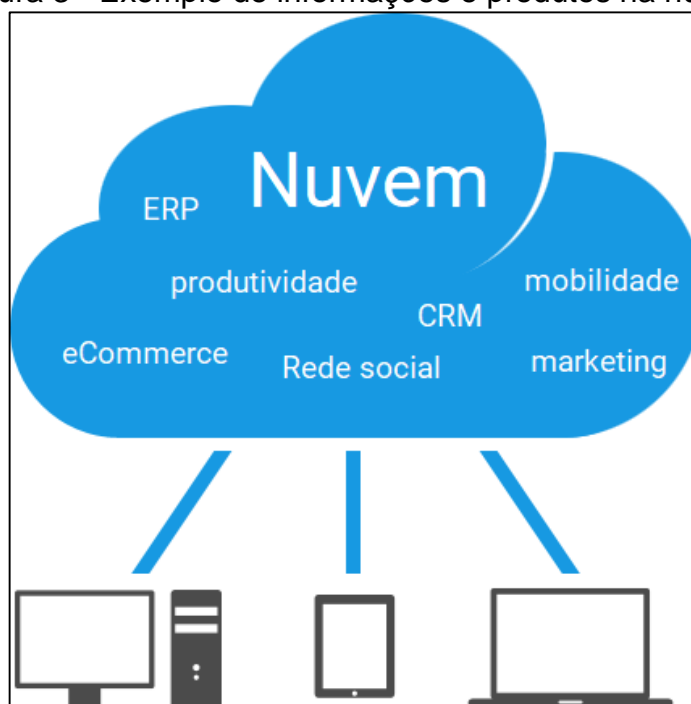
- Trata-se de um ambiente multiplataforma; ele pode ser executado no Windows, Macintosh e Linux.
- Tem por base o IDE de programação *Processing*, ambiente de desenvolvimento fácil de ser utilizado.
- Pode ser programado utilizando-se um cabo USB, sem necessidade de uma porta serial. Esse recurso é útil, uma vez que muitos computadores modernos não têm portas seriais.

- O *Arduino Project* foi desenvolvido em um ambiente educacional; portanto, é ideal para iniciantes que desejam resultados rápidos.

## 2.3 Nuvem

Pode-se entender que “a nuvem é uma representação para a Internet ou infra-estrutura de comunicação entre os componentes arquiteturais, baseada em uma abstração que oculta a complexidade de infra-estrutura” (RUSCHEL; ZANOTTO; MOTA, 2010, p.3), ou seja, quando é enviado informação para nuvem essas informações são armazenadas em servidores, os quais podem ser acessados através da Internet, por conta disso a nuvem é conhecida como o local onde pode-se ter acesso à informação de qualquer lugar do mundo. Na Figura 3 há exemplos de informações e produtos contidos na nuvem.

Figura 3 - Exemplo de informações e produtos na nuvem



Fonte: Elaborado pelos autores com base no Google Cloud (GOOGLE CLOUD, 2018)

Dentre todas as informações e produtos na nuvem, existe um conceito de serviços na Internet chamado *web service*, “*web service* é um sistema de *software* projetado para apoiar interações máquina-para-máquina interoperáveis pela rede, fornecendo uma interface descrita em um formato processável por máquina” (OLIVEIRA, 2011, p.9).

O serviço consiste num módulo de software instalado numa plataforma computacional com acesso à rede e oferecido pelo provedor de serviços. “Um serviço existe para ser usado por um consumidor, podendo funcionar também como um cliente de outro serviço” (OLIVEIRA, 2011, p.9).

Uma forma de serviço que utiliza nuvem é o *SaaS* (*Software as a Service* ou *Software como um serviço*). Segundo a Azure (2018), uma plataforma da *Microsoft* que possui *SaaS* integrado, este tipo de serviço permite a conexão de usuários e uso de aplicativos através do uso da Internet, como por exemplo *Microsoft Office 365*, *Gmail* e *Google Drive*. Na Figura 4 é demonstrada toda abrangência do serviço *SaaS*.

Figura 4 - Abrangência do modelo SaaS



Fonte: Artigo da Microsoft Azure falando sobre o que é SaaS (AZURE, 2018)

Este tipo de serviço possui várias vantagens e facilidades tanto para o desenvolvimento quanto para o usuário final, como descartar a necessidade de criar toda uma infraestrutura para alocar um servidor onde a aplicação vai executar, poupando tempo e diminuição de custos com implantação. Os serviços disponibilizados como Azure (Microsoft), *Google Cloud Platform* (Google) são exemplos de plataformas que são consolidadas e garantem o serviço hospedado vinte e quatro horas *online*, podendo ser acessado de qualquer lugar no mundo, conforme descrito pela Azure (2018) e Cloud (2018).

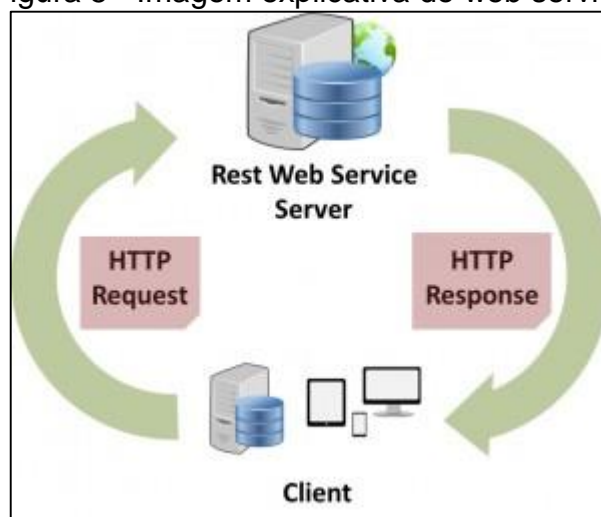
O modelo *SaaS* oferece um preço dinâmico, conforme a demanda, pois têm uma política de pague apenas o que usar, então para aplicações menores não há necessidade de montar uma grande infraestrutura de alto custo. Conforme for surgindo a necessidade de ampliação da infraestrutura, basta apenas incrementar o plano oferecido pela plataforma escolhida em poucos cliques.

Uma forma muito comum da criação de um modelo SaaS é fazer a construção de um *web service* que esteja fazendo uso de REST.

## 2.4 REST

Um modelo de arquitetura utilizada para realizar comunicações com um *web service* é o REST, o termo REST significa *Representational State Transfer* (FIELDING, 2000, tradução nossa). O modelo de arquitetura REST pode ser exemplificado quando existir um dispositivo que está se comunicando com um *web service* que esteja utilizando o modelo REST. Na Figura 5 pode-se visualizar como é feita a comunicação entre o cliente e um *web service*.

Figura 5 - Imagem explicativa do web service



Fonte: *Web Service at a Glance* (HELICAL, 2015)

De acordo com Fielding (2000), o REST trata-se de um modelo de arquitetura que utiliza o protocolo HTTP (*HyperText Transfer Protocol*) para fazer suas comunicações, fazendo uso de uma notação para transferência de dados, sendo ela o XML e o JSON.

### 2.4.1 HTTP e HTTPS

Segundo Souza Silva e Silva (2009, apud STALLINGS 2005) HTTP significa *Hypertext Transfer Protocol*, sendo um protocolo da *World Wide Web* mais popularmente conhecido como WWW. O HTTP pode ser utilizado em qualquer aplicação que seja no formato de cliente/servidor e que envolva a utilização de

hipertexto nos arquivos, por se tratar de cliente/servidor o HTTP é muito utilizado na Internet.

O HTTP foi desenvolvido para ser usado na Internet, embora seu desenho definitivo permita que ele possa ser usado em outras formas de aplicação. Por essa razão, são aceitas operações chamadas método diferente da simples solicitação de uma página da web (SOUZA SILVA; SILVA, 2009 apud TANENBAUM, 2003, p.3).

Analisando o que Souza Silva e Silva (2009) citaram, toda requisição realizada na Internet é por meio do HTTP, ou seja, para se ter acesso a informações de algum *site* é necessário fazer o uso do protocolo HTTP como forma de comunicação.

De acordo com Souza Silva e Silva (2009), o HTTP possui uma variação de protocolo que é nomeado por HTTPS (*Hypertext Transfer Protocol Secure*), que se trata de fazer o uso do protocolo SSL (*Secure Sockets Layer*) que é uma tecnologia de segurança desenvolvida pela Netscape em 1994.

HTTPS é um protocolo que tem por finalidade garantir a segurança na transmissão de dados, em aplicações que envolvam dados sigilosos, como o *e-commerce*, transação bancária, senhas, informações privadas e tantas outras (SOUZA SILVA; SILVA, 2009), ou seja, quando é possível danificar a integridade dos dados do usuário que está fazendo uso da aplicação é recomendável que seja utilizado o protocolo HTTPS.

#### 2.4.2 Métodos de requisições

O REST faz uso do protocolo HTTP para o envio das requisições, essas requisições utilizam métodos que segundo Moro et al. (2009), são utilizadas para determinada tarefa. Estes métodos são: *GET* para recuperar uma representação de um recurso, *PUT* para criar um novo recurso ou modificar um existente, *DELETE* para deletar um recurso e *POST* comumente utilizado para criação de um novo recurso (MORO et al. 2009).

#### 2.4.3 JSON

O JSON é um acrônimo para “*JavaScript Object Notation*”, que se trata de um formato de texto para a serialização de dados estruturados (FONSECA, 2007), ou seja, é um texto que contém os dados de um objeto de uma linguagem de programação.

Figura 6 - Imagem explicativa do JSON

```
1. {
2.   "funcionarios":[
3.     {
4.       "Nome": "Pedro",
5.       "Idade": "40",
6.       "Cargo": "Gerente"
7.     },
8.     {
9.       "Nome": "Ana",
10.      "Idade": "35",
11.      "Cargo": "Diretora"
12.    },
13.    {
14.      "Nome": "Carla",
15.      "Idade": "23",
16.      "Cargo": "Recepcionista"
17.    }
18.  ]
19. }
```

Fonte: Artigo da Matera falando sobre a utilização de Json e jQuery (MATERA, 2013)

Segundo Fonseca (2007), o JSON possui quatro tipos primários para seus atributos, sendo eles *strings*, números, booleanos e nulos, e mais dois tipos estruturados que são os objetos e vetores.

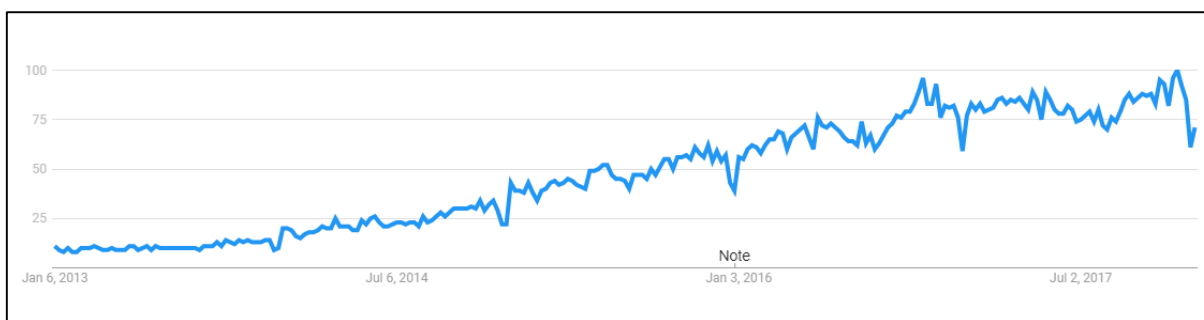
Um objeto é uma coleção não ordenada de zero ou mais pares nome/valor, onde o nome é uma *string* e o valor é uma *string*, número, booleano, nulo, objeto ou vetor (FONSECA, 2007). Na Figura 6 onde é mostrado um exemplo de um objeto com um vetor de “funcionários” onde dentro desse vetor possui outros objetos contendo atributos como nome, idade e cargo do funcionário.

Concluindo, para Fonseca (2007), o JSON foi criado com a finalidade de ser simples, portátil, textual e um subconjunto do JavaScript.

## 2.5 Internet das coisas

Um assunto que vem ganhando repercussão nos últimos anos é Internet das Coisas, do inglês *Internet of Things* (IoT), tendo um aumento de aproximadamente 9 vezes nas buscas do Google de 2013 para 2017 (Google Trends, 2018), conforme no Gráfico 4.

Gráfico 4 - Pesquisa de Internet das Coisas



Fonte: Google Trends (GOOGLE TRENDS, 2018)

GUBBI et al. (2013) trazem a seguinte visão sobre a Internet das Coisas, dizendo que a computação deve ir além dos dispositivos móveis e *tablets*, e deve conectar os objetos que convivemos no dia-a-dia para melhorar nosso ambiente. Como descrito por Teixeira (2014), o *Internet of things* são “coisas” físicas ligadas à virtualização.

A Internet das Coisas (IoT –Internet of Things) é uma infraestrutura de rede dinâmica e global com capacidades de auto configuração, baseada em protocolos de comunicação padronizados e interoperáveis, onde “coisas” físicas e virtuais têm identidades, atributos físicos e personalidades virtuais. (TEIXEIRA, et al., 2014, p.1).

Analisando esta definição de Teixeira (2014), é possível entender a Internet das Coisas, que se trata de uma grande rede de dispositivos físicos e virtuais que estão conectados a uma grande rede de servidores, podendo ser a Internet.

### 3 METODOLOGIA

Para abordagem inicial foi utilizado o formato de pesquisa exploratória com base em mídias digitais, artigos científicos e livros, procurando autores com projetos semelhantes e que possam agregar valor para a aplicação.

No gerenciamento de projeto foi escolhido o modelo ágil para definição das tarefas e do desenvolvimento. Como ferramenta visual, foi escolhido o Trello (*kanban* online) que subdivide as tarefas em cinco listas: *Backlog* (tarefas que estão prontas para serem desenvolvidas), *Impediments* (tarefas que não podem ser desenvolvidas, pois depende de outra(s) tarefa(s) ser(em) antes finalizada(s)), *Doing* (tarefas que estão em andamento na *sprint*), *Testing* (tarefas que já estão desenvolvidas e necessitam ser testadas pelos integrantes) e *Done* (tarefas que foram finalizadas e cumpriram todos os critérios de aceitação dos integrantes e foram marcadas como concluídas).

Para o levantamento de requisitos, seguindo o modelo ágil, foi adotada a elaboração de *User Stories* (histórias que contemplam a interação do usuário com a funcionalidade do sistema) e as *Spikes* (histórias que detalham a parte técnica do sistema).

## 4 DESENVOLVIMENTO

Durante a pesquisa realizada, foi possível entender como será desenvolvido um sistema de segurança domiciliar, utilizando tecnologias como arquitetura web, Arduino e dispositivos móveis.

### 4.1 Requisitos do Usuário

As histórias de usuário estão listadas de maneira simplificada no Quadro 1 conforme os quadros completos no Apêndice A.

Quadro 1 - Histórias de usuários

Histórias	Histórias de usuário
E1	Eu como usuário gostaria de ter a possibilidade de criar uma conta para mim
E2	Eu como usuário gostaria de ter a possibilidade de alterar os dados da minha conta
E3	Eu como usuário gostaria de ter a possibilidade de me autenticar pela conta criada diretamente no aplicativo
E4	Eu como usuário gostaria de poder cadastrar um grupo de dispositivos
E5	Eu como usuário gostaria de vincular um dispositivo com um dos meus grupos de dispositivos, havendo uma tela que realiza a leitura de <i>QR Codes</i> dos dispositivos
E6	Eu como usuário gostaria de convidar um usuário para ser participante do meu grupo de dispositivos
E7	Eu como usuário gostaria de poder aceitar ou recusar um convite de participação de um grupo de dispositivos
E8	Eu como usuário gostaria de visualizar na tela principal os grupos de dispositivos relacionados a mim
E9	Eu como usuário gostaria de visualizar os dispositivos dos grupos de dispositivos que sou administrador ou participante
E10	Eu como usuário administrador de um grupo de dispositivos gostaria de ligar/desligar o alarme do grupo
E11	Eu como usuário gostaria de receber uma notificação de alerta do alarme
E12	Eu como usuário gostaria de poder identificar na lista de dispositivos quais dispositivos estão sem acesso à Internet

## 4.2 Requisitos do Sistema (*Spikes*)

As *spikes* do sistema estão listadas no Quadro 2 (referente ao *web service*), no Quadro 3 (referente aos dispositivos) e no Quadro 4 (referente ao aplicativo).

Quadro 2 – *Spikes Web Service*

Histórias	Épico 2: <i>Web service</i>
E1	O sistema deve conter um <i>web service</i> na nuvem
E2	O <i>web service</i> deve realizar sua comunicação utilizando o modelo <i>RESTful</i>
E4	O <i>web service</i> deve receber e retornar dados criptografados utilizando uma criptografia criada
E5	O <i>web service</i> deve realizar sua comunicação através do protocolo HTTPS
E6	O <i>web service</i> deve ter um banco de dados relacional para armazenamento dos dados
E7	Deve ser possível controlar o cadastro de usuários do sistema
E8	Deve ser possível controlar os grupos de dispositivos
E9	Deve ser possível controlar os dispositivos
E10	Deve ser possível controlar os dados dos dispositivos de detecção de movimento
E11	Deve ser possível controlar os dados dos dispositivos de sirene
E12	Deve enviar uma notificação do alarme para o aplicativo

Quadro 3 - *Spikes Dispositivos*

Histórias	Épico 3: Dispositivos de segurança
E1	Os dispositivos devem ser construídos utilizando a placa Arduino UNO
E2	Os dispositivos devem se conectar com a rede Wi-Fi através do módulo ESP8266 do Arduino
E3	Os dispositivos de segurança devem estar conectados a Internet através de uma rede Wi-Fi de responsabilidade do usuário
E4	Os dispositivos devem se conectar com a rede Wi-Fi utilizando o padrão de segurança de rede WPS ( <i>Wi-Fi Protected Setup</i> )
E5	Os dispositivos devem enviar dados para o <i>web service</i> para informar que estão conectados a Internet

E6	O dispositivo de detecção de movimento deve enviar dados para o <i>web service</i> quando detectar um movimento
E7	Os dispositivos devem ser energizados por pilhas
E8	O dispositivo de sirene deve verificar o <i>web service</i> para saber se deve soar

Quadro 4 - *Spikes* Aplicativo

Histórias	Épico 4: Aplicativo
E1	O aplicativo deve ser construído no modelo <i>MVP (Model View Presenter)</i>
E2	Realizar requisições <i>HTTPS</i> ao <i>web service</i> utilizando o modelo <i>RESTful</i>
E3	O banco de dados será implementado no modelo relacional para armazenar os dados consumidos do <i>web service</i>
E4	Para processos que possam envolver custo de processamento e tempo de resposta utilizar conceitos de programação reativa

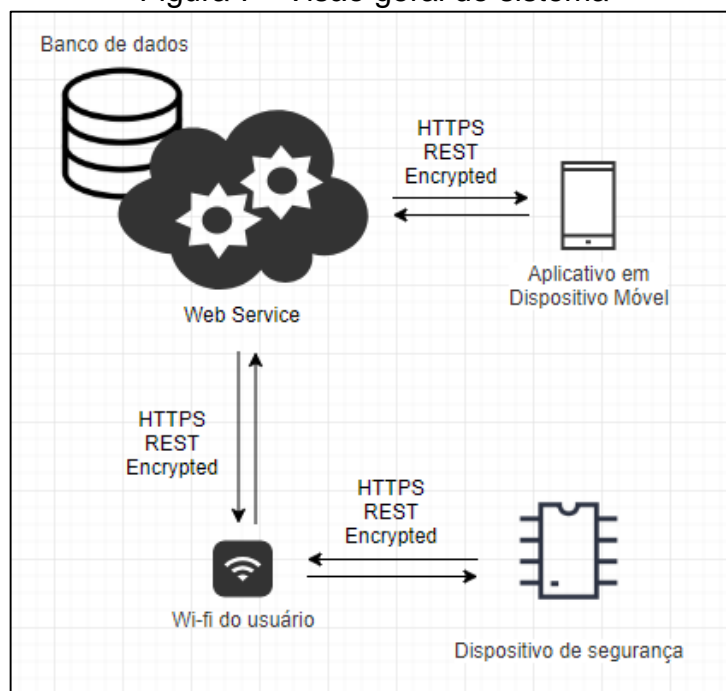
### 4.3 Descrição do Projeto

O sistema é focado em uma solução para segurança domiciliar utilizando o modelo SaaS, IoT, Nuvem e Arduino. Em geral a solução que permite o usuário realizar a instalação de dispositivos de segurança em sua residência e realizar o controle através de um aplicativo para dispositivos móveis.

São dois dispositivos de segurança: Sensor de movimento e sirene. Cada dispositivo tem o seu papel na segurança, o sensor de movimento serve para, quando o alarme estiver ligado, captar movimentos e ativar o alarme, notificando o usuário e acionando as sirenes. As sirenes têm o papel de alertar a vizinhança que há um invasor na residência.

Os dispositivos se conectam na rede interna do usuário pelo Wi-Fi através do botão WPS, após isso os dispositivos já estão conectados à Internet e podem ser vinculados no aplicativo. Para o usuário controlar o dispositivo pelo aplicativo ele deve realizar a leitura de um QR *CODE* que está junto ao dispositivo, a partir de então o usuário pode controlar o dispositivo e compartilhar com outros usuários.

Figura 7 - Visão geral do sistema



Fonte: Elaborado pelos autores com base nos requisitos do sistema (2018)

Nessa arquitetura, demonstrada na Figura 7, o *web service* tem como função ser o intermediador de comunicação entre dispositivos de segurança e o aplicativo, além disso o *web service* é responsável pela comunicação com o banco de dados para armazenamento dos dados cruciais. Os dados trafegados são através do protocolo HTTPS utilizando modelo REST e criptografia dos dados.

Para realizar o versionamento do projeto foi utilizada a plataforma Git que, com base em GitHub (2018), é uma ferramenta que possibilita o desenvolvimento do sistema com vários desenvolvedores trabalhando simultaneamente no mesmo código fonte, aumentando a produtividade. Todo o código fonte gerado pelos desenvolvedores é armazenado em um repositório na nuvem, conhecido como *origin*. A ferramenta online do Git escolhido foi o Bitbucket, sendo uma ferramenta sem custos e permite o repositório ser privado, ou seja, outros desenvolvedores que não possuem permissão ao repositório do sistema, não podem visualizar e nem realizar *download* do sistema.

#### 4.4 Tecnologias

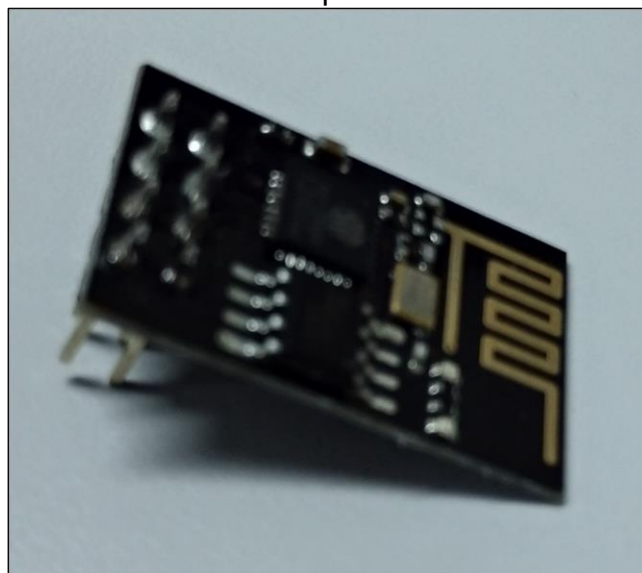
As tecnologias utilizadas no desenvolvimento desse projeto foram segmentadas de acordo com a necessidade e o domínio de cada parte.

#### 4.4.1 Arduino

O Arduino foi escolhido justamente por ter um baixo custo, por possuir um desenvolvimento mais simples para implementação das funcionalidades dos microprocessadores e por deter vários módulos que possuem funcionalidades distintas e que sejam passíveis de aplicação do conceito de IoT, como por exemplo, o módulo sensor de movimento e sirene que foram utilizados nesse projeto.

Para conseguir aplicar o conceito de IoT nos dispositivos é necessário possuir conexão com a Internet, para isso foi utilizado o módulo ESP8266 que trata de realizar a conexão com o Wi-Fi.

Figura 8 - Módulo ESP8266 para se conectar com a Wi-Fi



Fonte: Elaborado pelos autores (2018)

Este módulo ESP8266, que está representado na Figura 8, foi atrelado aos dispositivos (Sensor de movimento e sirene) para fazer com que os mesmos consigam enviar e buscar dados do *web service*.

Para fazer com que os dispositivos funcionem sem a necessidade de estarem conectados à energia, foi adicionado um conjunto de 5 pilhas recarregáveis da marca Rayovac, cada pilha possui 1,5 volts totalizando 7,5 volts. Na saída do fio positivo foi adicionado um switch para ligar e desligar os dispositivos.

Figura 9 – Conjunto de pilhas que irá energizar os dispositivos



Fonte: Elaborado pelos autores (2018)

Utilizando essas pilhas faz com que seja possível acoplar o dispositivo em qualquer lugar da casa e também com elas possibilita manter os módulos em funcionalidade.

Para realização da conexão do dispositivo com a Internet, foi feito o uso da tecnologia WPS (*Wi-Fi Protect Setup*), adicionando um botão para realizar o pareamento com a rede.

Figura 10 - Botão WPS dos dispositivos

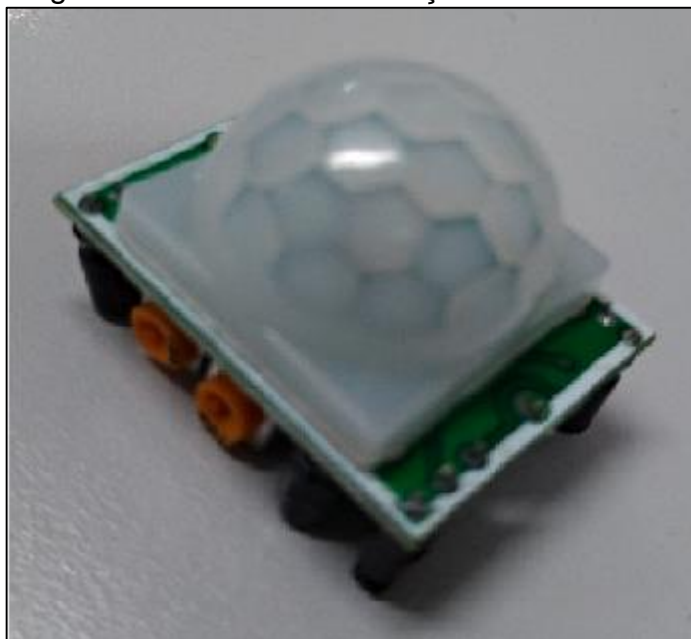


Fonte: Elaborado pelos autores (2018)

A conexão com o WPS funcionará da seguinte maneira, será necessário o usuário pressionar o botão WPS que existe em seu roteador e em seguida o botão representado na Figura 10 que estará em cada dispositivo, dessa forma o dispositivo irá se conectar na rede do usuário.

Para realizar a montagem do dispositivo de sensor de movimento foi utilizado o módulo PIR DYP-ME003 do Arduino que têm a finalidade de capturar movimentações realizadas que são com base na variação de luz infravermelha emitida pela radiação do corpo humano, este sensor consegue detectar movimentos em uma área de até 7 metros.

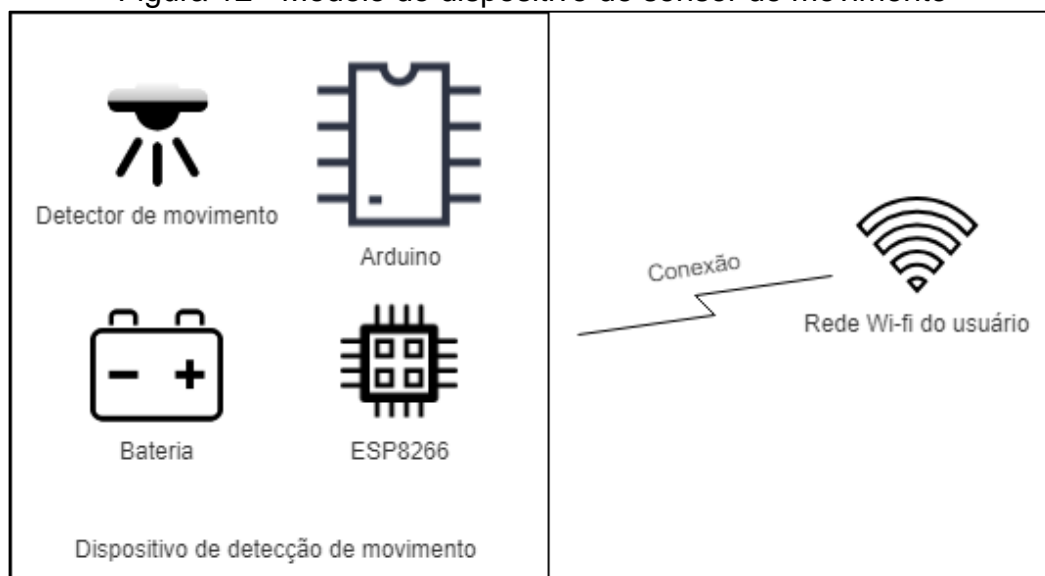
Figura 11 - Módulo de detecção de movimento



Fonte: Elaborado pelos autores (2018)

Para o funcionamento correto do dispositivo de sensor de movimento, foram interligados os seguintes componentes, Arduino, ESP8266 (Wi-Fi), Pilhas e o módulo de sensor de movimento.

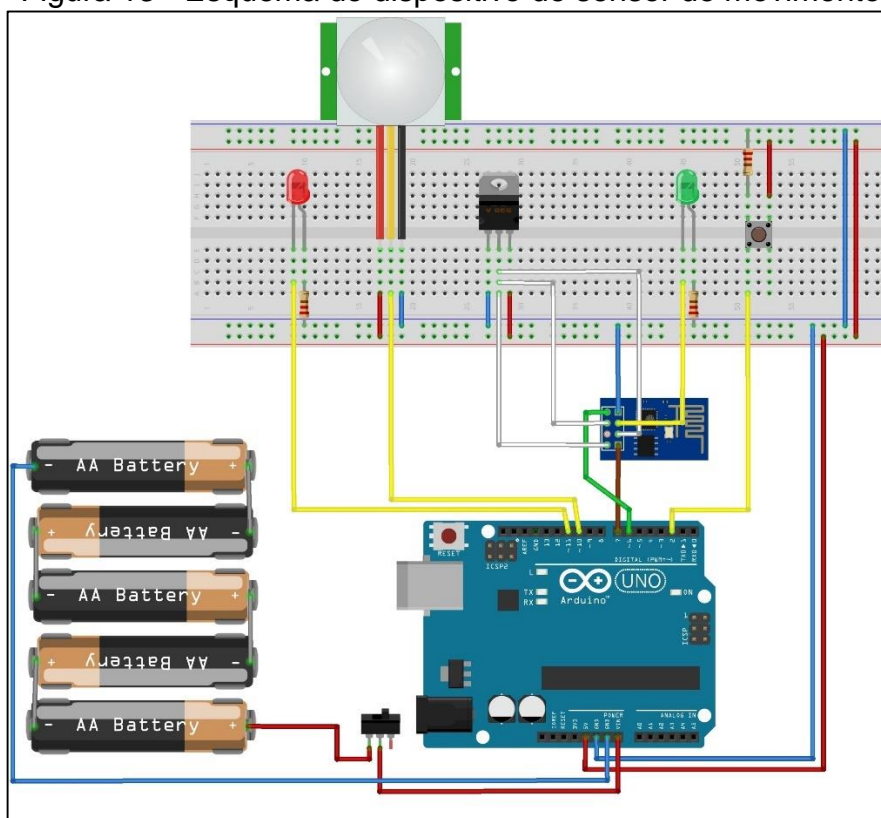
Figura 12 - Modelo do dispositivo de sensor de movimento



Fonte: Elaborado pelos autores (2018)

Tendo em vista o modelo da Figura 12, foi montado um esquema do dispositivo de sensor de movimento ilustrado na Figura 13.

Figura 13 - Esquema do dispositivo de sensor de movimento

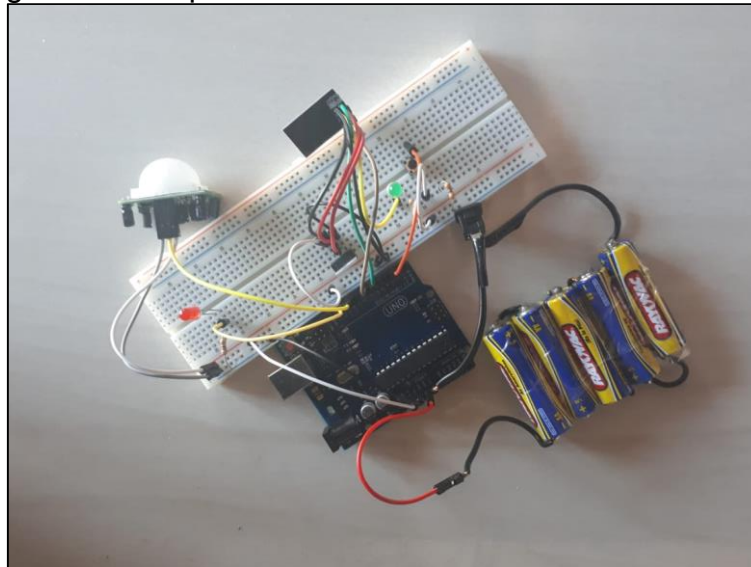


Fonte: Elaborado pelos autores (2018)

E com o esquema da Figura 13 definido o dispositivo foi construído conforme a Figura 14, utilizando 5 pilhas de 1,5v, um interruptor, uma placa Arduino UNO, uma

placa ESP8266, dois leds, um botão sem retenção, 3 resistores 220 ohms, um regulador de tensão LM1117T 3.3v e um modulo de sensor de movimento PIR.

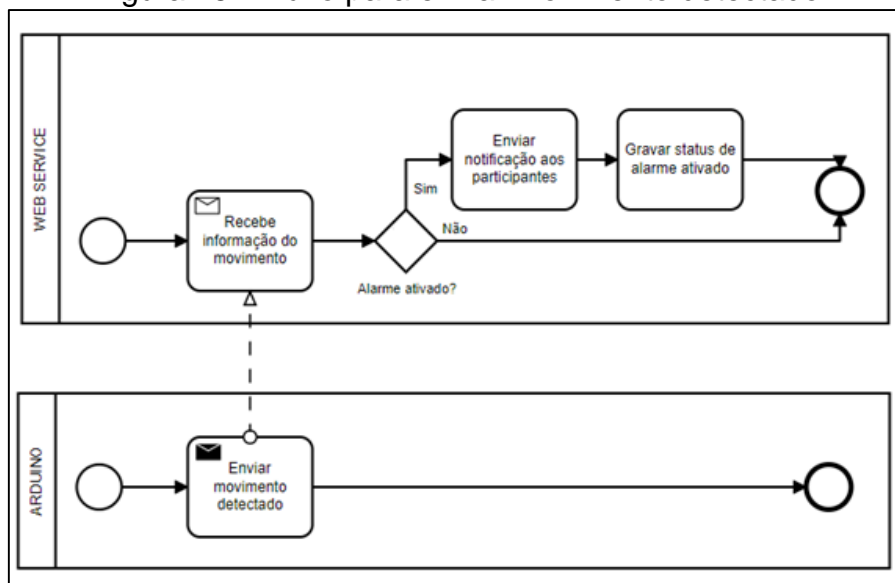
Figura 14 - Dispositivo de sensor de movimento montado



Fonte: Elaborado pelos autores (2018)

O dispositivo de sensor de movimento funciona da seguinte maneira: sempre que detectar um movimento será enviado este dado para o *web service* através do módulo ESP8266. Também será enviado um sinal para o *web service* a cada 1 minuto para informar que o dispositivo está conectado à Internet.

Figura 15 - Fluxo para enviar movimento detectado



Fonte: Elaborado pelos autores (2018)

Por fim, a única finalidade do dispositivo de sensor de movimento será notificar o *web service* caso houver uma detecção de movimento.

Por último, o dispositivo de sirene, que terá como finalidade apenas fazer barulho quando soar o alarme. Para montar esse dispositivo de sirene foi utilizado uma sirene de 12V que possui um índice de 120dB conforme a Figura 16.

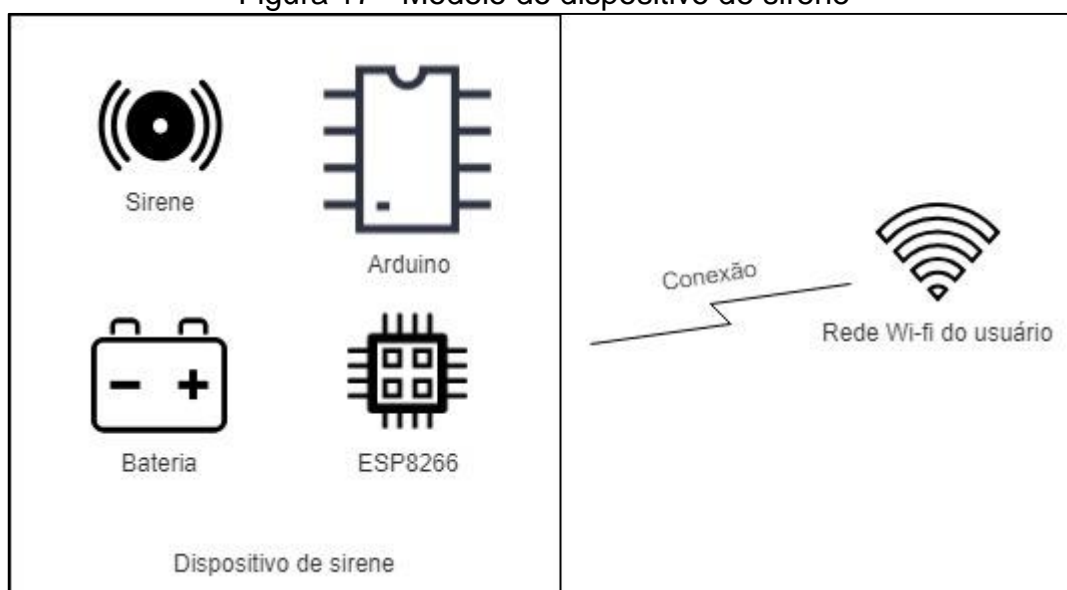
Figura 16 – Sirene do dispositivo de sirene



Fonte: Elaborado pelos autores (2018)

Para o funcionamento correto do dispositivo de sirene, foram interligados os seguintes componentes, Arduino, ESP8266 (Wi-Fi), Pilhas e a sirene.

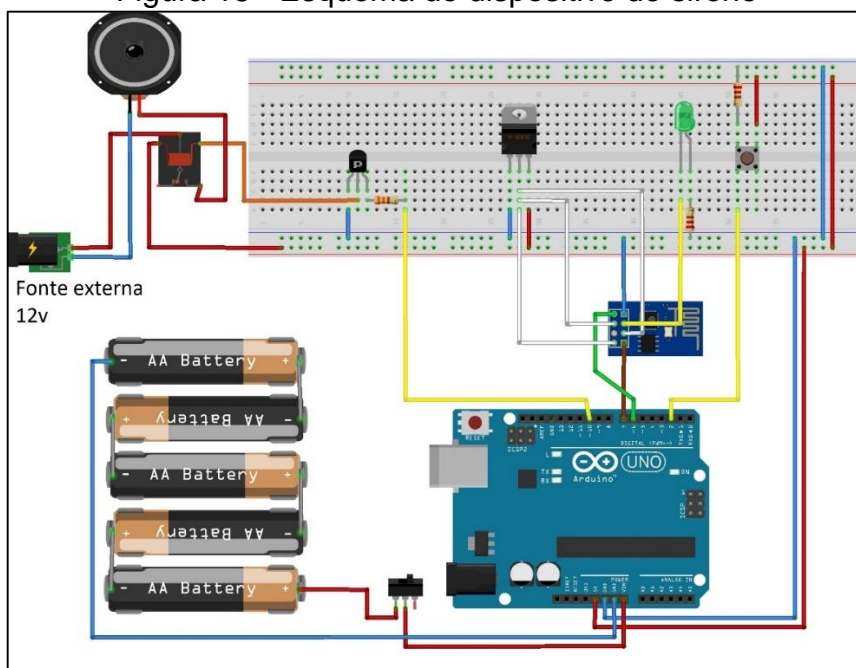
Figura 17 - Modelo do dispositivo de sirene



Fonte: Elaborado pelos autores (2018)

Tendo em vista o modelo da Figura 17, foi montado um esquema do dispositivo de sirene ilustrado na Figura 18.

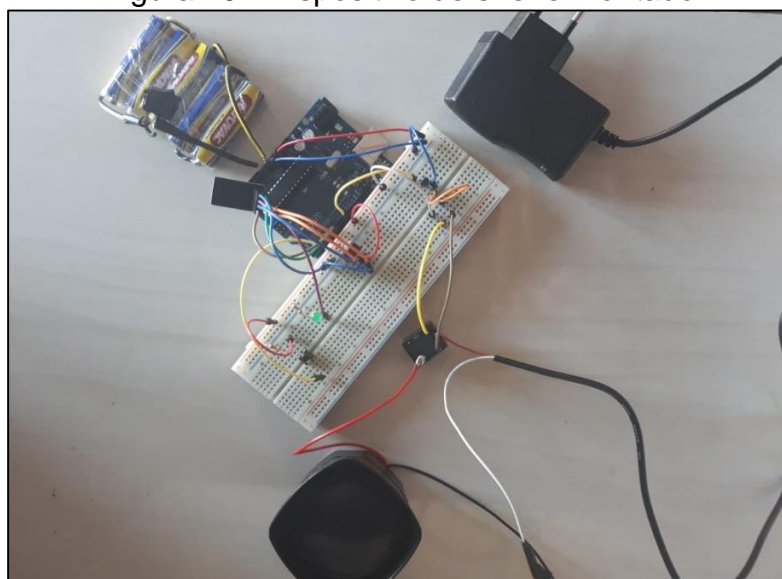
Figura 18 - Esquema do dispositivo de sirene



Fonte: Elaborado pelos autores (2018)

E com o esquema da Figura 18 definido o dispositivo foi construído conforme a Figura 19, utilizando 5 pilhas de 1,5v, um interruptor, uma placa Arduino UNO, uma placa ESP8266, um led, um botão sem retenção, dois resistores 220 ohms, um resistor 4700 ohms, um regulador de tensão LM1117T 3.3v, um transistor 2SC458, um *relay*, uma sirene e uma fonte externa de 12v.

Figura 19 - Dispositivo de sirene montado

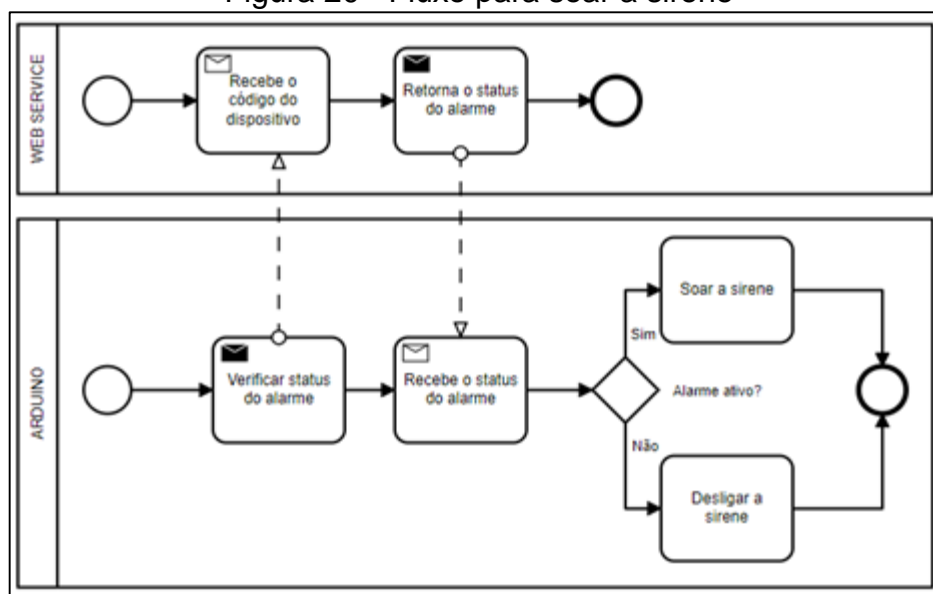


Fonte: Elaborado pelos autores (2018)

O dispositivo de sirene funciona da seguinte maneira, a cada minuto o dispositivo irá consultar o *web service* através do módulo ESP8266 para saber se o

alarme está ativado, ou seja, se enquanto o alarme estivesse ligado e o sensor de movimento detectasse uma presença iria ativar o alarme, então devendo soar a sirene.

Figura 20 - Fluxo para soar a sirene



Fonte: Elaborado pelos autores (2018)

A única finalidade do dispositivo de sirene será verificar o status do alarme pelo *web service* e caso esteja ativado, soar o alarme como ilustrado na Figura 20.

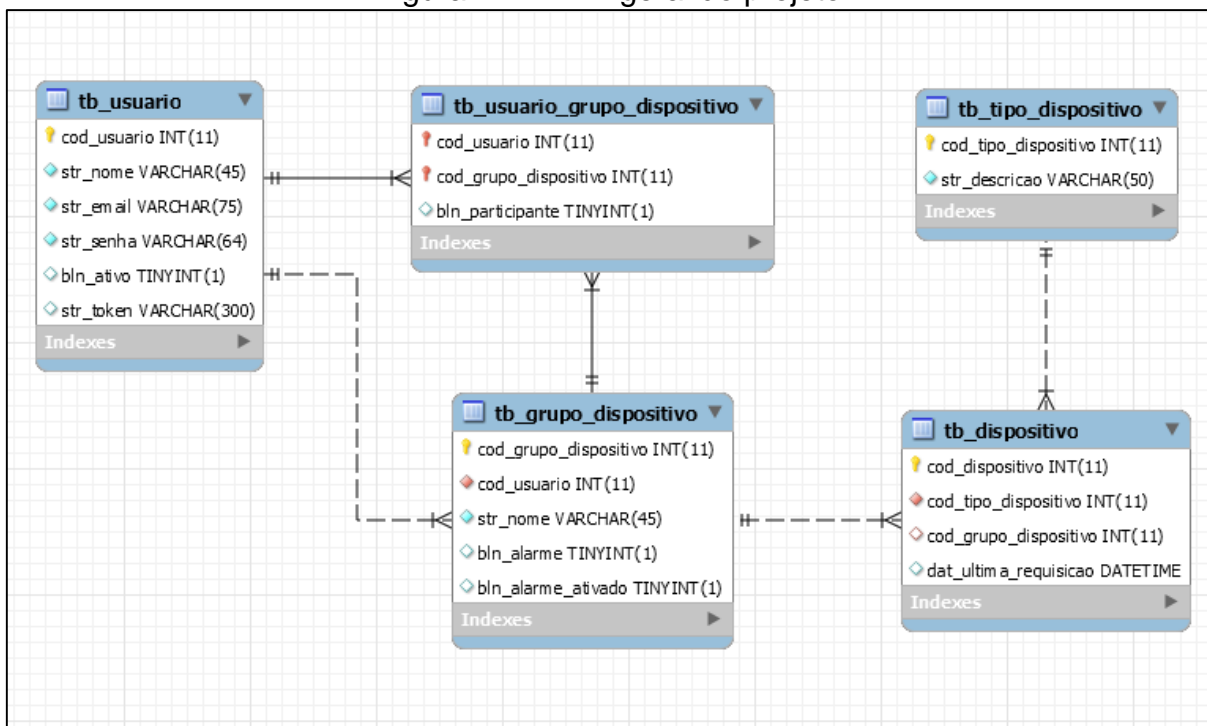
#### 4.4.2 Web Service

O principal recurso desse projeto são os dispositivos, e estes dispositivos foram montados para trabalhar de forma isolada, uma das formas para realizar isso é passar as regras de negócio do sistema para dentro do próprio dispositivo, mas isso foi descartado por ser inviável para a proporção desse sistema, pois no mesmo momento que fosse encontrado uma falha ou necessitasse de alguma melhoria teria que recolher todos os dispositivos já em funcionamento e alterar seu código fonte, por esse motivo não seria nada viável armazenar a regra de negócio em cada dispositivo. Para sanar esse problema, foi construído um *web service*, que será o responsável por possuir toda a regra de negócio do sistema, dessa forma os dispositivos só irão ter a função de coletar dados e mandar para o *web service*, que irá receber esses dados e processar, aplicando a regra de negócio estabelecida.

O *web service* trabalhará da seguinte maneira, com o esquema de rotas e com a persistência de dados, para persistir os dados, ou seja, para armazenar, foi

utilizado o banco de dados MySQL, nele foi construído uma estrutura de tabelas para atender a demanda.

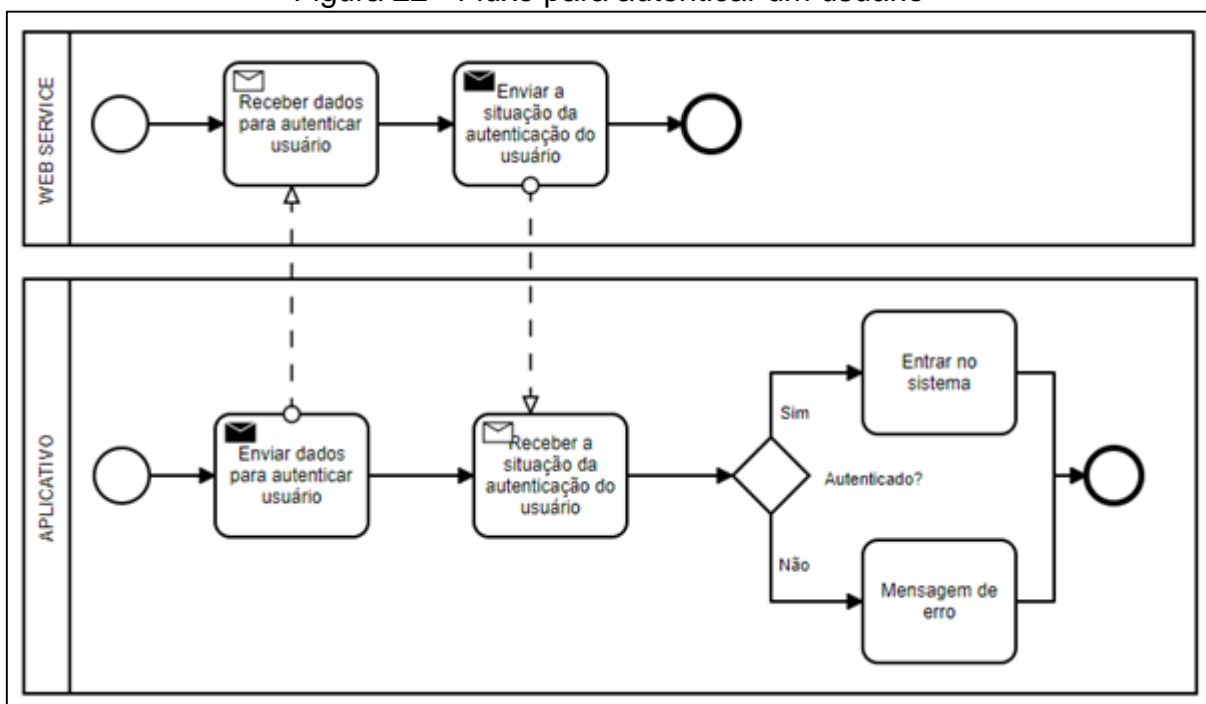
Figura 21 - MER geral do projeto



Fonte: Elaborado pelos autores (2018)

No esquema de rota foi aplicado o conceito RESTful para separar o tipo da regra de negócio como por exemplo, buscas, inserções, alterações e exclusões, assim deixando mais estruturado o *web service*. As rotas significam o recurso que será utilizado através da URL que é passada, como por exemplo, <https://meudominio/webservice/v1/usuario/login>, nesta URL está especificando que será utilizado o recurso de login da entidade usuário.

Figura 22 - Fluxo para autenticar um usuário



Fonte: Elaborado pelos autores (2018)

Para realizar essa comunicação com o *web service*, primeiramente é necessário criptografar todos os dados para aumentar a segurança e a sigilidade dos dados do usuário.

Toda requisição para o *web service*, deve seguir a seguinte regra, converter em JSON os dados que serão transmitidos, após a conversão deve ser aplicado a criptografia nesse JSON e por fim mandar tudo que foi criptografado em um atributo chamado *data*.

Figura 23 - Corpo da requisição enviada para o web service

```
{
  data: "WzZ2bFw1PG5bNWd8Z1tFeVs1VXNmNkV5ZjVvM2..."
}
```

Fonte: Elaborado pelos autores (2018)

Dessa forma, caso a requisição seja capturada por um terceiro, estará com seu conteúdo criptografado, dificultando o acesso a estes dados.

Da mesma maneira que é criptografado para enviar para o *web service*, serve para o retorno da requisição.

Figura 24 - Corpo do retorno da requisição do web service em caso de sucesso

```
{
  res: true,
  msg: "Av22bF51PG5bNWd8sldFeVs1VXNmNkV5ZjVvM2..."
}
```

Fonte: Elaborado pelos autores (2018)

Diferente do envio, agora é retornado um atributo chamado “res”, que faz referência ao resultado da execução do recurso utilizado, no caso do exemplo da Figura 24 tudo ocorreu de forma correta, mas caso houver um erro será diferente o retorno.

Figura 25 - Corpo do retorno da requisição do web service em caso de erro

```
{
  res: false,
  msg: "Não foi possível estabelecer conexão com o banco de dados"
}
```

Fonte: Elaborado pelos autores (2018)

Neste caso, o resultado volta como falso, dizendo que o recurso não conseguiu ser executado com sucesso, havendo um erro durante sua execução, e o atributo “msg” informa a mensagem de qual erro ocorreu no *web service*.

Cada rota no *web service* é responsável por realizar uma ação, seja ela uma inserção, alteração, exclusão ou uma busca de informação, como por exemplo o cadastro de usuário.

Figura 26 - Rota para cadastrar um novo usuário

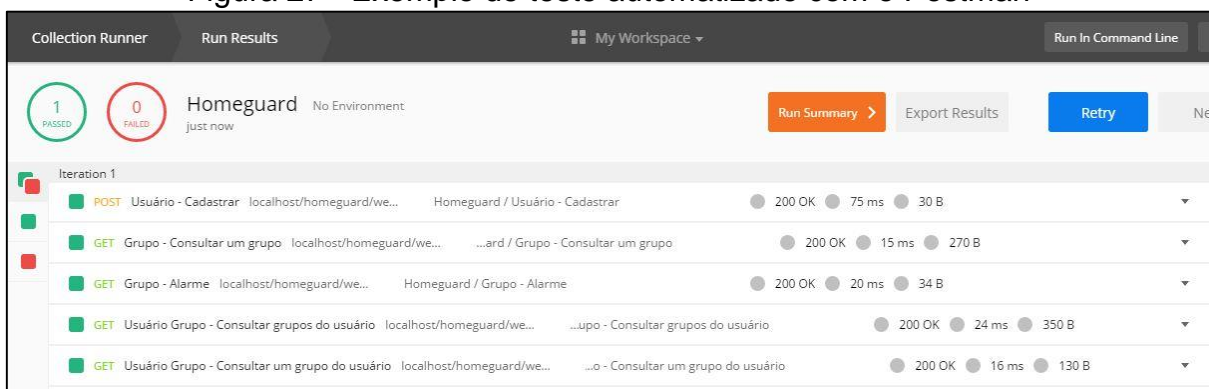
```
$app->post("", function($data) use ($app) {  
    $app->attrData($data, "nome", "");  
    $app->attrData($data, "email", "");  
    $app->attrData($data, "senha", "");  
    $app->attrData($data, "token", "");  
  
    $app->notNull($data, ["nome", "email", "senha"]);  
  
    if (!filter_var($data->email, FILTER_VALIDATE_EMAIL)) {  
        throw new Exception("Email inválido");  
    }  
  
    $sql = "SELECT cod_usuario FROM tb_usuario WHERE str_email = '{$data->email}'";  
  
    if ($app->selectOne($sql)) {  
        throw new Exception("Email já cadastrado");  
    }  
  
    $usuario = new Usuario();  
    $usuario->setStrNome($data->nome);  
    $usuario->setStrEmail($data->email);  
    $usuario->setStrSenha($data->senha);  
    $usuario->setBlnAtivo(1);  
    $usuario->setStrToken($data->token);  
  
    $usuario->insert();  
  
    $app->response($usuario->getCodUsuario());  
});
```

Fonte: Elaborado pelos autores (2018)

Conforme a Figura 26, este é um exemplo de uma rota para cadastrar usuário, ou seja, para utilizar o recurso dessa rota, deve ser informado uma URL que corresponde a essa rota, tendo que informar campos, como nome, e-mail, senha e *token*.

Para garantir que todos os recursos do *web service* estejam funcionando corretamente, foi implementado testes unitários através do software *Postman*. Dessa forma sempre que for necessário mandar uma nova atualização para o repositório do projeto na Nuvem será realizado a execução desse teste automatizado.

Figura 27 - Exemplo de teste automatizado com o Postman



Fonte: Elaborado pelos autores (2018)

Após executar todas as rotas do *web service*, é informado a situação desse teste, mostrando se houve algum erro e em qual rota houve esse erro, caso não haja, o *web service* está pronto para funcionar em produção.

O *web service* então ficou sendo um intermediador entre dispositivos e aplicativo, tornando essa comunicação mais simples por possuir toda a regra de negócio e a persistência de dados em um só local que pode ser facilmente acessado de qualquer lugar por qualquer dispositivo.

#### 4.4.3 Aplicativo para Android

Com os dispositivos coletando os dados e repassando para o *web service* resta gerenciar essa informação e para realizar esse gerenciamento será utilizado um aplicativo construído para plataforma Android com a linguagem de programação Kotlin.

O primeiro contato que o usuário irá ter após baixar e instalar o aplicativo, será com a tela login, onde usuário pode se autenticar para entrar no sistema ou escolher a opção para criar uma nova conta.

Figura 28 - Tela de Login do aplicativo

A screenshot of a login screen for an application. At the top center is a black silhouette of a house. Below it, the word "Email" is followed by a text input field containing "marcelo.kafels@hbsis.com.br". Underneath, the word "Senha" is followed by a password input field with three dots. A grey button labeled "ENTRAR" is positioned below the password field. At the bottom, there is a link that says "Não possui acesso? Crie sua conta!".

Email  
marcelo.kafels@hbsis.com.br

Senha  
...

ENTRAR

Não possui acesso? Crie sua conta!

Fonte: Elaborado pelos autores (2018)

Para realizar o login, o usuário deve informar o campo de e-mail e senha, e após confirmar, será realizada a autenticação como consta no fluxo que está na Figura 28.

Caso o usuário não tenha uma conta e opte por criar uma, ele deverá acessar a tela de cadastro de usuário que fica localizada após clicar no botão de criar nova conta localizado na tela de login (Figura 28). Assim, a tela de cadastro de usuário será aberta, como mostra a Figura 29.

Figura 29 - Tela de Cadastro do aplicativo



A imagem mostra a interface de usuário para o cadastro de um novo usuário. No topo, há uma barra azul com um ícone de seta para trás e o título "Cadastrar usuário". Abaixo, há quatro campos de entrada de texto:

- Nome: "Kafels"
- Email: "marcelo.kafels@hbsis.com.br"
- Senha: "\*\*\*\*\*"
- Confirmar senha: "\*\*\*\*\*"

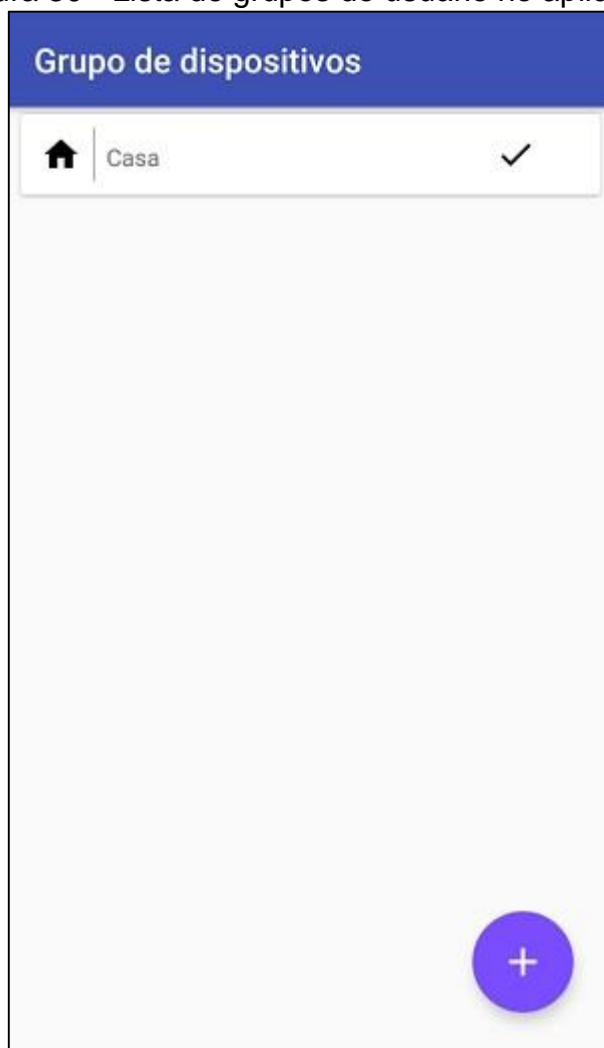
Na base do formulário, há um botão cinza com o texto "CADASTRAR".

Fonte: Elaborado pelos autores (2018)

Para fazer o cadastro de uma nova conta, o usuário deve informar os seguintes campos: nome, e-mail, senha e confirmar a senha para então confirmar seu cadastro.

Após ter realizado o login ou ter criado uma nova conta, o usuário é automaticamente direcionado para tela inicial administrativa, onde estará listando todos os grupos de dispositivos que o usuário possui, representado na Figura 30. Todos os grupos que foram criados pelo usuário possuem uma imagem representativa de uma casa, caso o grupo for de outro usuário (por meio de convites), é representado com uma imagem de duas pessoas.

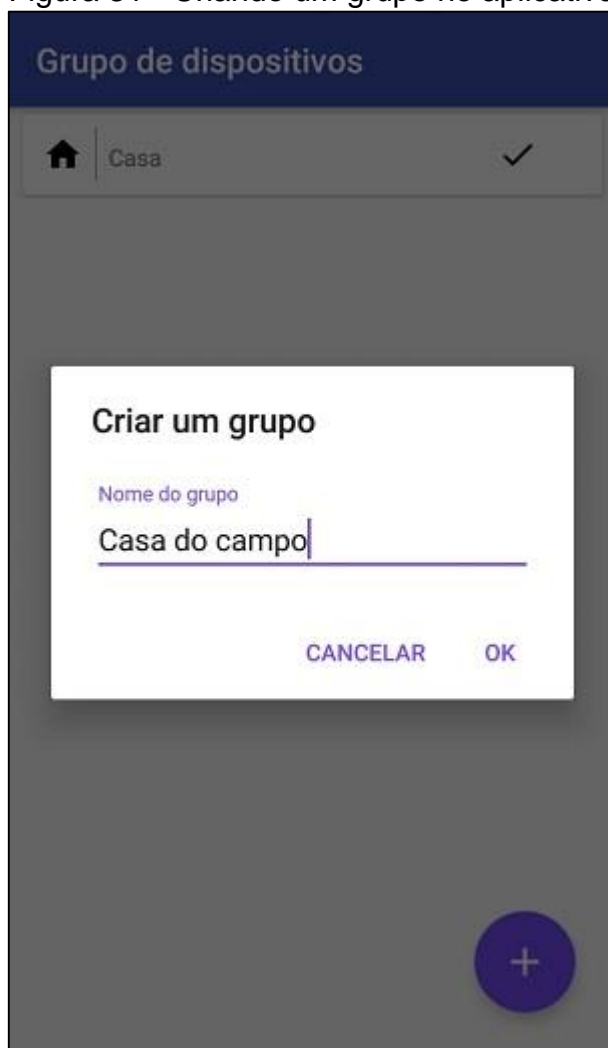
Figura 30 - Lista de grupos do usuário no aplicativo



Fonte: Elaborado pelos autores (2018)

Ao clicar no botão flutuante, uma *popup* é aberta para possibilitar a criação de um novo grupo de dispositivos, conforme a Figura 31.

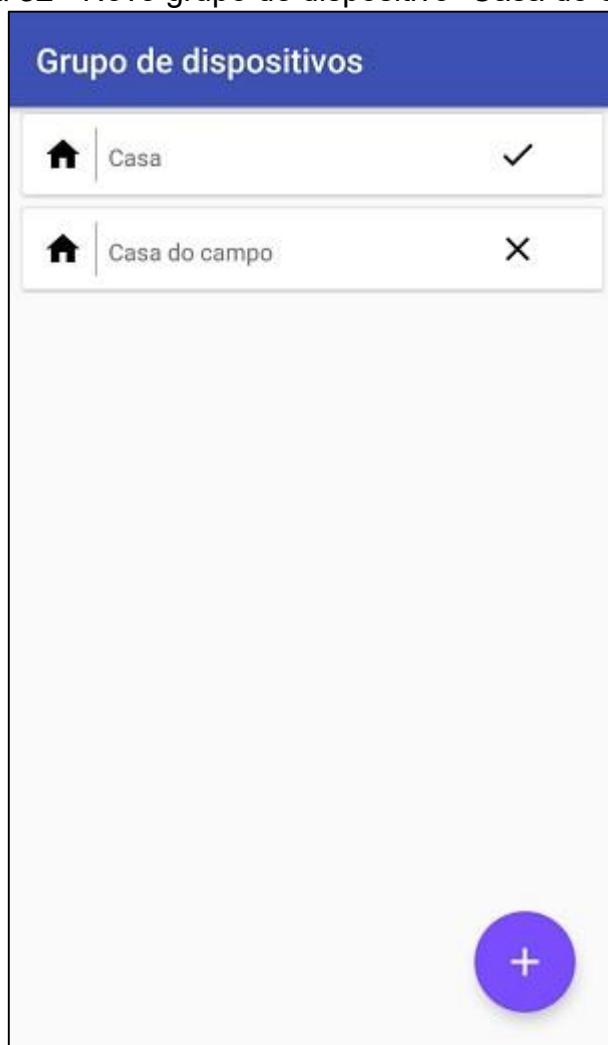
Figura 31 - Criando um grupo no aplicativo



Fonte: Elaborado pelos autores (2018)

Após criar um grupo, o mesmo será listado na tela de grupos de dispositivo do usuário, conforme demonstrado na Figura 32.

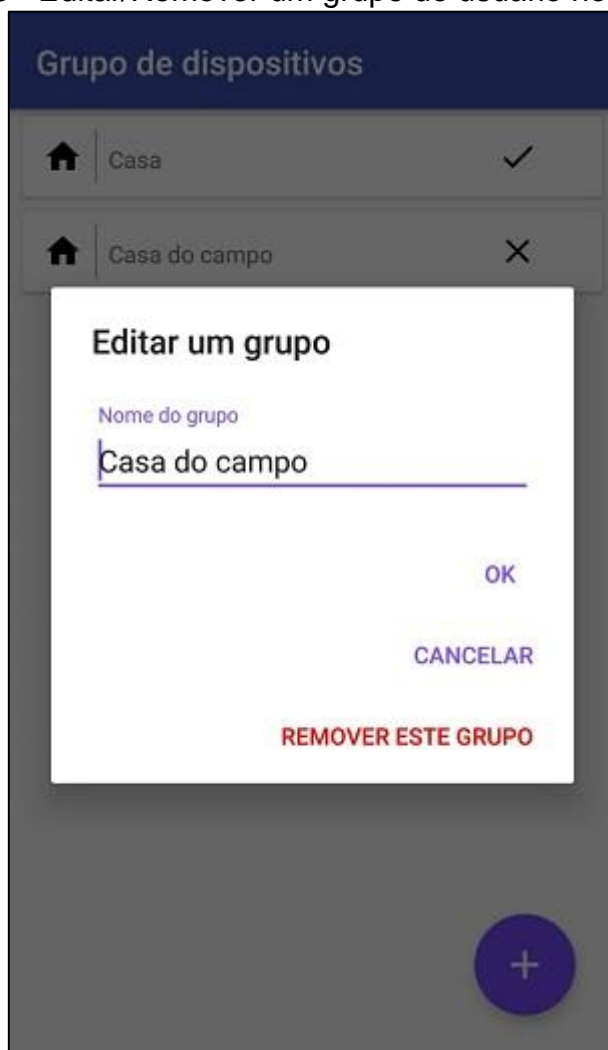
Figura 32 - Novo grupo de dispositivo “Casa do campo”



Fonte: Elaborado pelos autores (2018)

Ao clicar e segurar em um grupo de dispositivo, uma *popup* é aberta dando a possibilidade tanto de editar o nome do grupo como remover o mesmo, caso o usuário que estiver realizando a ação ser o administrador do grupo, conforme a Figura 33. Se o usuário não for administrador do grupo, a *popup* será aberta dando apenas a possibilidade de desvincular o grupo da conta do usuário.

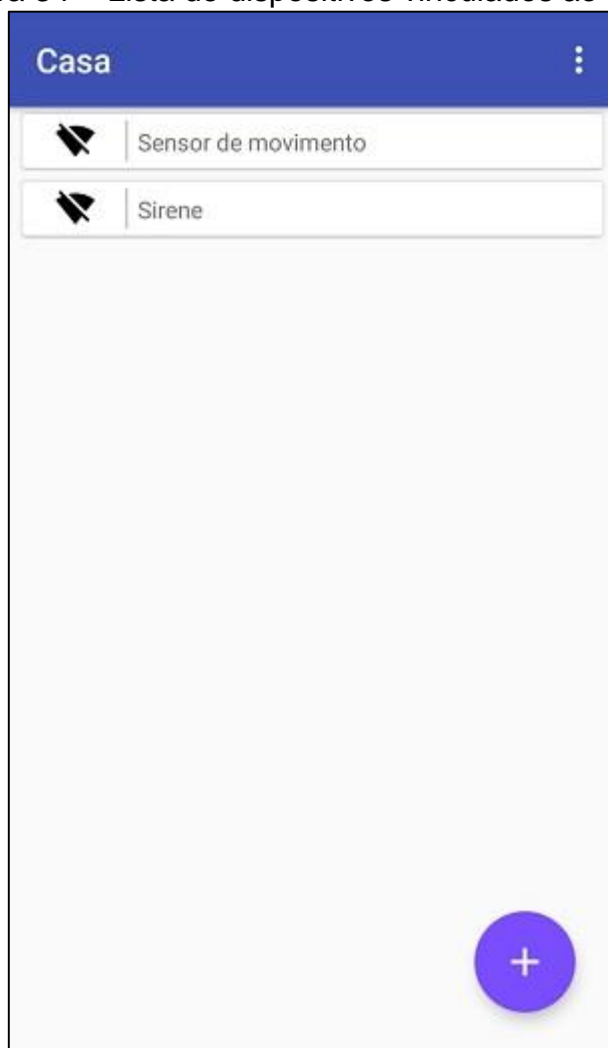
Figura 33 - Editar/Remover um grupo do usuário no aplicativo



Fonte: Elaborado pelos autores (2018)

Quando o usuário clicar em um grupo de dispositivo, será aberta a tela de dispositivos vinculados ao grupo, conforme demonstrado na Figura 34.

Figura 34 - Lista de dispositivos vinculados ao grupo



Fonte: Elaborado pelos autores (2018)

Para vincular novos dispositivos, é necessário clicar no botão flutuante, onde será aberta a câmera para realizar a leitura do *QR CODE* do dispositivo, conforme a Figura 35. Ao vincular o dispositivo, o mesmo será listado na tela de dispositivos vinculados ao grupo (Figura 34).

Figura 35 - Vinculando um dispositivo ao grupo



Fonte: Elaborado pelos autores (2018)

Para remover um dispositivo vinculado a um grupo, é necessário clicar e segurar em um dispositivo, então uma *popup* será aberta para confirmação de exclusão, conforme a Figura 36.

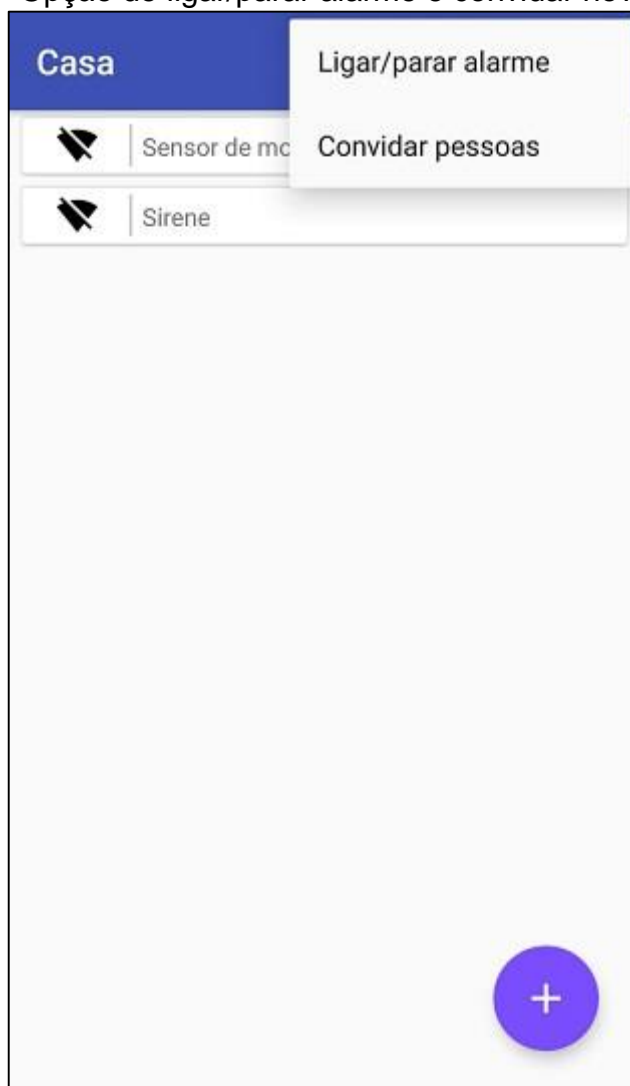
Figura 36 - Removendo um dispositivo do grupo



Fonte: Elaborado pelos autores (2018)

Ao clicar nos três pontos na barra superior da tela de lista de dispositivos (Figura 34), permite o usuário ligar/parar o alarme e convidar novas pessoas ao grupo de dispositivo, conforme a Figura 37. Se for clicado no botão "Ligar/parar alarme", o status do grupo do dispositivo (habilitado/desabilitado) será alterado. A visualização do status de habilitado/desabilitado pode ser visualizado na tela de grupo de dispositivos (Figura 32).

Figura 37 - Opção de ligar/parar alarme e convidar novas pessoas



Fonte: Elaborado pelos autores (2018)

Caso for clicado no botão “Convidar pessoas”, uma *popup* será aberta permitindo escrever o e-mail de outro usuário, representado na Figura 38.

Figura 38 - Convidar uma nova pessoa ao grupo



Fonte: Elaborado pelos autores (2018)

Quando um usuário for convidado, ao acessar a tela de grupo de dispositivos uma *popup* é aberta mostrando os grupos em que o usuário foi convidado. É mostrada a informação do nome do grupo, o nome do usuário administrador do grupo e as opções de aceitar/recusar, conforme a Figura 39.

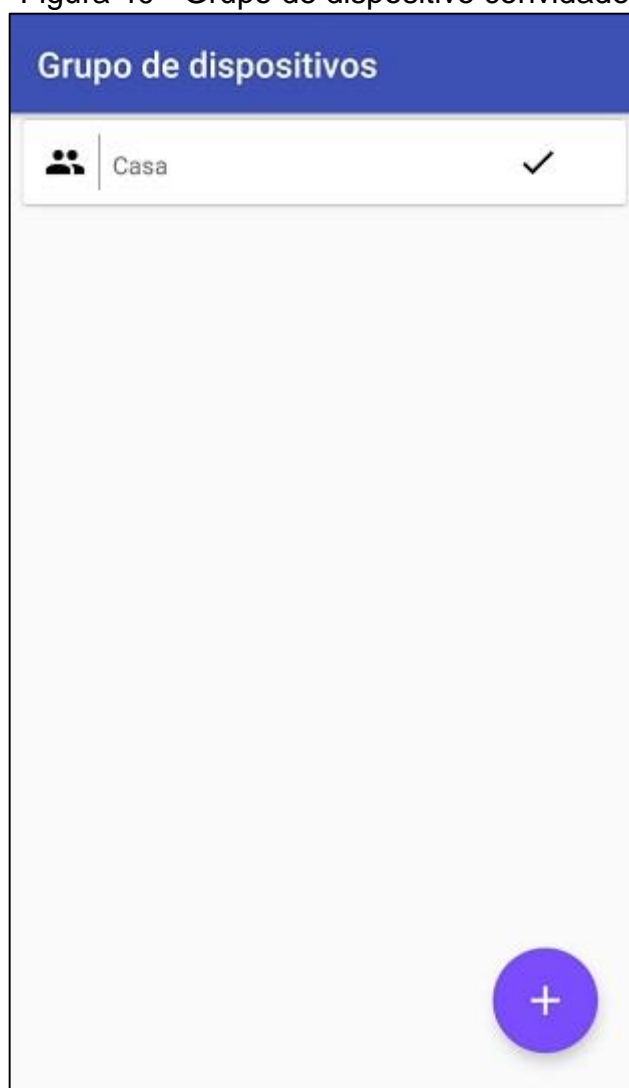
Figura 39 - Aceitando/Recusando o convite de um grupo



Fonte: Elaborado pelos autores (2018)

Quando o usuário aceita entrar em um grupo na qual ele foi convidado o grupo de dispositivo fica disponível para ele com um ícone de duas pessoas, como na Figura 40, e pode gerenciar a segurança do grupo a partir de então.

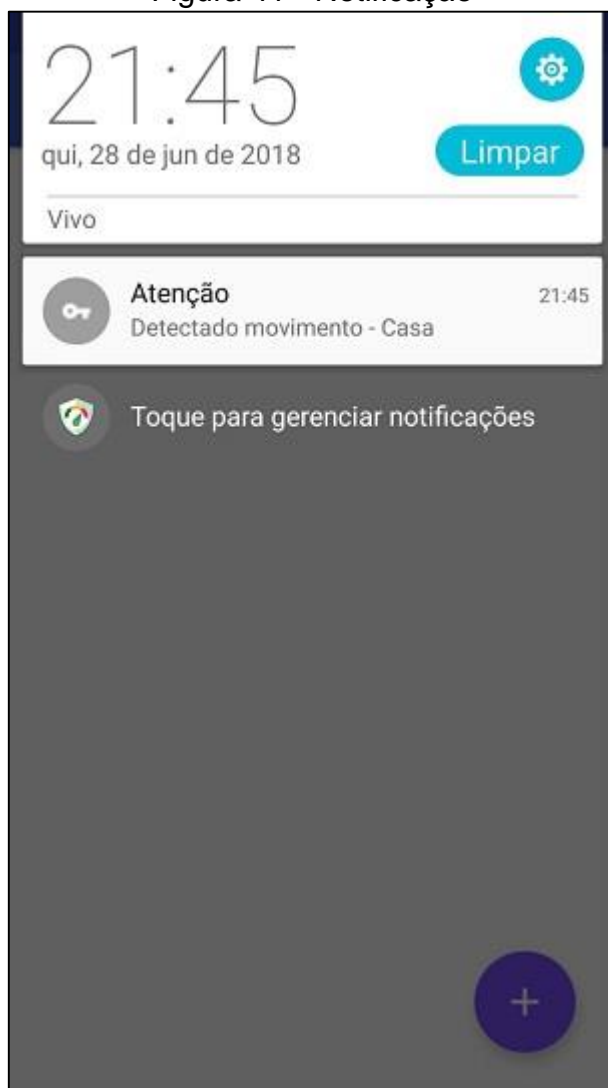
Figura 40 - Grupo de dispositivo convidado



Fonte: Elaborado pelos autores (2018)

Quando o webservice envia uma mensagem para o aplicativo, é apresentado um ícone de notificação no topo do celular com o título “Atenção” e descrição “Detectado movimento - (Nome do grupo do dispositivo)”, conforme a Figura 41.

Figura 41 - Notificação



Fonte: Elaborado pelos autores (2018)

Caso o usuário clicar na notificação, será aberto o aplicativo na lista de grupos de dispositivos (Figura 32).

#### 4.4.4 Qualidade de software

Para garantir a integridade das funcionalidades do projeto foram utilizados testes automatizados na API através do software *Postman*. Conforme Bernardo (2008), atualmente o que se espera de um produto é que ele esteja funcionando corretamente para o cliente.

Controlar a qualidade de sistemas de software é um grande desafio devido à alta complexidade dos produtos e às inúmeras dificuldades relacionadas ao processo de desenvolvimento, que envolve questões humanas, técnicas, burocráticas, de negócio e políticas. Idealmente, os sistemas de software devem não só fazer corretamente o que o cliente precisa, mas também fazê-lo de forma segura, eficiente e escalável e serem flexíveis, de fácil manutenção e evolução (BERNARDO, 2008, p.1).

Foram implementados testes unitários na API que são executados sempre que for necessário enviar uma nova atualização para o repositório no *Bitbucket*.

#### 4.4.5 Infraestrutura do Projeto

Para que o projeto funcione corretamente, é necessário que haja no mínimo uma certa infraestrutura, como por exemplo, para o Arduino é necessário que seja utilizado um módulo de Wi-Fi e que haja disponibilidade de uma rede sem fio, pois o Arduino precisa enviar e receber os dados do *web service*. Além da conexão com a Internet o Arduino também necessita de um conjunto de 5 pilhas recarregáveis Rayovac de 7,5v para ter energia, permitindo a execução dos outros módulos como, sensor de movimento e sirene.

Para o *web service* é necessário um servidor que contenha *Apache* para executar a linguagem de programação base do *web service*, que foi construído em *PHP*, e para o banco de dados deve conter o *MySQL*.

#### 4.4.6 Estimativa de custo

Tendo a infraestrutura e o desenvolvimento do projeto definido, foi possível fazer o levantamento da estimativa de custo.

Para infraestrutura, foi necessário o uso de um *web service*, que foi realizado através do serviço da *Umbler*, uma plataforma de serviço em nuvem, que oferece poder computacional de forma escalável. Para o projeto foi considerado a utilização do plano *Cloud Sites PP* que contempla um servidor *Apache* para compilação dos

códigos fontes na linguagem *PHP*, um banco relacional *MySQL*, certificação *SSL/TLS*, 1 GB de memória *RAM* entre outros serviços de gerenciamento do servidor, por um total de R\$ 60,00 reais ao mês, mais R\$ 35,00 reais do domínio do *site*.

O valor final do dispositivo de sensor de movimento, foi: R\$ 50,00 do *Arduino*; R\$ 10,00 das pilhas; R\$ 55,00 do módulo *ESP8266* para conexão com a *Wi-Fi*; R\$ 10,00 do módulo de sensor de movimento infravermelho. Já o valor final do dispositivo de sirene, que utiliza os seguintes itens, foi: R\$ 50,00 do *Arduino*; R\$ 10,00 das pilhas; R\$ 55,00 do módulo *ESP8266*; R\$ 15,00 da sirene; R\$ 8,00 da fonte de 12V. Esses valores podem ser menores em caso de compra através de atacado.

Em relação aos custos de desenvolvimento e suporte, seria considerado o custo de uma prestação de serviço mensal para o desenvolvimento do projeto e suportes a clientes.

Portanto, a estimativa do dispositivo de sensor de movimento fica por conta de R\$ 125,00, já o dispositivo de sirene fica R\$ 138,00 e para a infraestrutura um total de R\$ 95,00.

#### 4.4.7 Resultado

O desenvolvimento do projeto foi realizado de forma multitarefa, ou seja, dispositivos *Arduino*, *web service* e aplicativo foram feitos todos ao mesmo tempo, dessa forma conseguindo ter uma maior produtividade e agilidade no desenvolvimento do sistema.

O projeto se mostrou uma ótima oportunidade para obter novos conhecimentos com algumas tecnologias. Nenhum integrante da equipe possuía consciência de por onde começar o desenvolvimento dos dispositivos, então foi buscado o entendimento nas linguagens *C/C++* para programação das placas e em elétrica para montagem dos circuitos. Mesmo quando a equipe encontrou dificuldades ela conseguiu encontrar pessoas dispostas a ajudar.

Pode ser citado como exemplo a dificuldade da equipe para entender como era feito a comunicação entre a placa *Arduino UNO* e a placa *ESP8266*, onde recebeu ajuda em um fórum chamado *Stack Overflow*, ou o conhecimento em elétrica necessário para montar os dispositivos que a equipe encontrou ajuda em uma loja de componentes eletrônicos de *Blumenau*.

O *web service* teve uma grande importância para o sistema, pois com ele foi possível realizar a comunicação entre dispositivos e aplicativo. Durante a construção

do *web service* não houve nenhum problema ou dificuldade, pois foi utilizada uma biblioteca própria para elaboração do *web service*. A finalidade do *web service* é basicamente realizar a persistência dos dados e também disponibilizá-los, seja para o dispositivo Arduino ou para o aplicativo, por exemplo, quando o dispositivo de sensor de movimento detecta, o Arduino informa ao *web service* o qual é responsável por armazenar essa informação, assim o aplicativo pode consultar o *web service* e saber que o sensor de movimento detectou algo.

Realizar testes automatizados para o *web service* foi de grande ajuda, pois garantiu a funcionalidade correta das rotas e também ajudou muito com a necessidade de ajustes.

O aplicativo teve como desafio o uso do Kotlin como linguagem de programação, onde a equipe não havia conhecimento do uso e o fato do aplicativo ser totalmente online. No início houve uma pequena dificuldade com a comunicação com o *web service* devida à criptografia que foi utilizada, pois enquanto não estivesse correta, a comunicação com o *web service* era inviabilizada. Para solucionar este problema, foi depurado tanto o *web service* quanto o aplicativo ao mesmo tempo, linha por linha até descobrir onde estava errado. Esse problema promoveu o trabalho em equipe para buscar a solução e permitir que fosse dado o andamento ao aplicativo. Já o uso do Kotlin, todas as dúvidas que foram surgindo, novamente foi bastante pesquisado no Stack Overflow sobre como implementar uma funcionalidade e quais eram as melhores formas. Não só no Stack Overflow, como também no próprio site do Kotlin, onde disponibilizam toda a documentação para a linguagem de programação.

Com esses desafios cumpridos, tornou-se possível a finalização do aplicativo, da qual sua finalidade é mostrar todos os dados do usuário e alertá-lo quando um movimento for detectado por um dispositivo vinculado a um grupo e também a função do gerenciamento dos grupos e dispositivos de segurança vinculados.

## 5 CONSIDERAÇÕES FINAIS

Atualmente a segurança é primordial e de certa forma, todos sentem ou desejam estar seguros. Conforme os dados apresentados, no Brasil o índice de criminalidade tende a crescer a cada ano, portanto há a necessidade da criação e aprimoramento de tecnologias voltadas à segurança.

O projeto teve como função criar dois dispositivos que possam auxiliar na segurança domiciliar, que foram os dispositivos de sensor de movimento e sirene. Esses dispositivos se comunicam com o *web service* que tem como função a comunicação entre o aplicativo e os dispositivos. O diferencial entre um dispositivo de alarme comum e o desenvolvido neste projeto, está o uso do IoT nos dispositivos, através da comunicação direta ao *web service*. Também a facilidade da implantação pelo usuário final com o uso do botão WPS, assim não havendo a necessidade de um técnico especializado para instalação. Portanto, com a instalação realizada e os dispositivos conectados à internet, permite agora o usuário gerenciar esses dispositivos diretamente no aplicativo, vinculando-os em grupos e também dando a possibilidade do compartilhamento de grupos com demais usuários, como por exemplos familiares.

Temos quatro implementações futuras que acreditamos ser importantes para a continuação do projeto. A primeira é a adição de um novo dispositivo para complementar a solução. Este novo dispositivo será uma câmera que possibilita os usuários visualizarem as imagens em tempo real, além de realizar gravações quando os alarmes soarem.

Outra implementação futura muito importante é a definição da mensalidade e a criação de uma forma de cobrança dos clientes com base na quantidade utilizada de dispositivos.

Para melhorar em questão de segurança uma importante implementação que deve ser feita é a funcionalidade dos dispositivos *off-line*. Hoje nossos dispositivos não possuem tratamento para continuar funcionando sem conexão com a Internet, porém é algo que tem que ser tratado para aumentar a segurança do usuário.

Como última implementação futura o sistema deve possuir uma integração com a polícia militar para avisar sobre invasões. Essa integração funcionaria enviando para a polícia dados da localização e do dono do domicílio, assim que for solicitado pelo usuário através do aplicativo.

Em primeiro momento, não há um planejamento para venda de dados dos usuários, para garantir o sigilo de dados dos mesmos.

É importante destacar que todos os objetivos propostos neste trabalho foram atendidos.

## REFERÊNCIAS

AGUIAR, Afrânio Carvalho. **Informação e atividades de desenvolvimento científico, tecnológico e industrial: tipologia proposta com base em análise funcional**. Ciência da Informação, v. 20, n. 1, 1991.

ARDUINO, 2018. **What is arduino?**. Disponível em <<https://www.arduino.cc/>>. Acesso em 09 ago. 2018.

BANZI, Massimo, 2011. **Primeiros passos com o Arduino**. Disponível em: <[https://issuu.com/novateceditora/docs/cap\\_\\_tulo\\_9788575222904](https://issuu.com/novateceditora/docs/cap__tulo_9788575222904)>. Acesso em: 30 jun. 2018.

BERNARDO, Paulo Cheque e KON, Fabio, 2008. **A importância dos testes automatizados**. Disponível em: <<https://www.ime.usp.br/~kon/papers/EngSoftMagazine-IntroducaoTestes.pdf>>. Acesso em: 30 de jun. 2018.

BRASIL. Constituição (1988). **Constituição da República Federativa do Brasil: promulgada em 5 de outubro de 1988**. Organização do texto: Juarez de Oliveira. 4. Ed. São Paulo: Saraiva, 1990. 168p. (Série Legislação Brasileira).

CAVALCANTE, Michelle M.; et al., 2014. **A plataforma Arduino para fins didáticos: estudo de caso com recolhimento de dados a partir do PLX-DAQ**. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2014/0037.pdf>>. Acesso em: 30 de jun. 2018.

CRUZ, Tércia Maria Ferreira da, 2008. **Mídia e segurança pública: a influência da mídia na percepção da violência**. Disponível em: <<https://lumina.ufjf.emnuvens.com.br/lumina/article/view/173/168>>. Acesso em: 30 de jun. 2018.

HELICAL, 2015. **Web Service at a glance**. Disponível em <<https://www.helicaltech.com/web-service-at-a-glance/>>. Acesso em 08 ago. 2018.

MORO, Tharcis dal; DORNELES, Carina F. e REBONATTO, Marcelo Trindade, 2009. **Web services WS-\* versus Web Services REST**. Disponível em: <<http://seer.ufrgs.br/index.php/reic/article/view/22140/12928>>. Acesso em: 30 de jun. 2018.

SOUZA, Anderson R. de; et al., 2011. **A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC**. Disponível em: <<http://sbfisica.org.br/rbef/pdf/331702.pdf>>. Acesso em: 30 de jun. 2018.

SOUZA SILVA, André Luis de Souza e SILVA, Regina Célia Marques Freitas, 2009. **Protocolo HTTP X Protocolo HTTPS**. Disponível em: <<http://www.nucleus.feituverava.com.br/index.php/nucleus/article/view/146/275>>. Acesso em: 30 de jun. 2018.

Desenvolvimento ágil. **SCRUM.** Disponível em: <<http://www.desenvolvimentoagil.com.br/scrum>>. Acesso em 04 de abril de 2018.

NASCIMENTO, Edvaldo Ferreira do, 2011. **Segurança da informação: um estudo sobre ferramentas de segurança presentes no Windows Server.** Disponível em: <<http://revista.fatectq.edu.br/index.php/interfacetecnologica/article/view/55/52>>. Acesso em: 30 de jun. 2018.

MATERA. 2013. **Template HTML - Utilizando JSON e JQuery.** Disponível em <<http://www.matera.com/blog/post/template-html-utilizando-json-e-jquery>>. Acesso em 08 ago. 2018.

FIELDING, Roy Thomas, 2000. **Architectural styles and the design of network-based software architectures.** Disponível em: <<http://jpkc.fudan.edu.cn/picture/article/216/35/4b/22598d594e3d93239700ce79bce1/7ed3ec2a-03c2-49cb-8bf8-5a90ea42f523.pdf>>. Acesso em 08 ago. 2018.

FONSECA, Erika G. P. da; VEGA, Alexandre S. de la, 2011. **TUTORIAL SOBRE INTRODUÇÃO A PROJETOS UTILIZANDO O KIT DE DESENVOLVIMENTO ARDUINO.** Disponível em: <[http://www.telecom.uff.br/pet/petws/downloads/artigos/cobenge\\_2011/COBENGE\\_art1677\\_2\\_ARDUINO\\_2011\\_08\\_30.pdf](http://www.telecom.uff.br/pet/petws/downloads/artigos/cobenge_2011/COBENGE_art1677_2_ARDUINO_2011_08_30.pdf)>. Acesso em 08 ago. 2018.

FONSECA, Rúben e SIMÕES, Alberto, 2007. **Alternativas ao XML: YAML e JSON.** Disponível em: <<http://alfarrabio.di.uminho.pt/~albie/publications/xmlyamljson07.pdf>>. Acesso em: 30 de jun. 2018.

FÓRUM BRASILEIRO DE SEGURANÇA PÚBLICA, 2017. **Anuário brasileiro de segurança pública. 2017.** Disponível em: <[http://www.forumseguranca.org.br/wp-content/uploads/2017/12/ANUARIO\\_11\\_2017.pdf](http://www.forumseguranca.org.br/wp-content/uploads/2017/12/ANUARIO_11_2017.pdf)>. Acesso em 17 mar. 2018.

GITHUB, Platform. **Built for developers.** Disponível em: <<https://github.com>>. Acesso em: 03 mai. 2018.

GOOGLE CLOUD. **Computação em nuvem, serviços de hospedagem e APIs do Google.** Disponível em: <<https://cloud.google.com/?hl=pt-br>>. Acesso em: 18 abr. 2018.

GOOGLE TRENDS. 2018. **Exploração do termo Internet of Things.** Disponível em <<https://trends.google.com/trends/explore?date=2013-01-01%202017-12-31&q=%2Fm%2F02vnd10>>. Acesso em 02 abr. 2018.

GUBBI, Jayavardhana; et al., 2013. **Internet of Things (IoT): A vision, architectural elements, and future directions.** Disponível em: <<https://arxiv.org/ftp/arxiv/papers/1207/1207.0203.pdf>>. Acesso em: 30 de jun. 2018.

MCROBERTS, Michael, 2011. **Arduino básico.** Disponível em: <[http://adjutojunior.com.br/arduino/arduino\\_b%C3%A1sico\\_Michael\\_McRoberts.pdf](http://adjutojunior.com.br/arduino/arduino_b%C3%A1sico_Michael_McRoberts.pdf)>. Acesso em: 30 de jun. 2018.

MICROSOFT AZURE. **O que é o SaaS?**. Disponível em: <<https://azure.microsoft.com/pt-br/overview/what-is-saas/>>. Acesso em: 06 abr. 2018.

OLIVEIRA, Ricardo Ramos de, 2011. **Desenvolvimento de Web Services RESTful e SOAP-WSDL utilizando as Implementações de referência JAX-RS e JAX-WS**. Disponível em: <<http://garapa.intermedia.icmc.usp.br/mediawiki/images/5/57/Curso.pdf>>. Acesso em: 30 de jun. 2018.

RUSCHEL, Henrique; ZANOTTO, Mariana Susan; MOTA, Wélton Costa Da, 2010. **Computação em nuvem**. Disponível em: <<https://www.ppgia.pucpr.br/~jamhour/RSS/TCCRSS08B/Welton%20Costa%20da%20Mota%20-%20Artigo.pdf>>. Acesso em: 17 mar. 2018.

SILVA, Leandro Jamir, 2017. **Internet das Coisas**. Disponível em: <<https://www.riuni.unisul.br/bitstream/handle/12345/3940/TCC%20FINAL%20LEANDRO%20JAMIR%20SILVA.pdf?sequence=1&isAllowed=y>>. Acesso em: 30 de jun. 2018.

TEIXEIRA, Fernando A; et al., 2014. **Siot – Defendendo a Internet das Coisas contra Exploits**. Disponível em: <<http://roitier.pro.br/wp-content/uploads/2017/09/ST14-1.pdf>>. Acesso em: 30 de jun. 2018.

## APÊNDICE A – HISTÓRIAS DE USUÁRIO

E1	Eu como usuário gostaria de ter a possibilidade de criar uma conta para mim
A1	Deverá ser informado os seguintes campos: <b>Nome, Email, Senha e Confirmar senha</b>
A2	Ao clicar em <b>Criar conta</b> será criado a conta do usuário
A3	Ao salvar esta conta, deverá automaticamente realizar a autenticação do usuário

E2	Eu como usuário gostaria de ter a possibilidade de alterar os dados da minha conta
A1	Deverá poder alterar os seguintes campos: <b>Nome, Nova senha e Confirmar nova senha</b>
A2	Ao clicar em <b>Salvar</b> será salvo os dados do usuário

E3	Eu como usuário gostaria de ter a possibilidade de me autenticar pela conta criada diretamente no aplicativo
A1	Na tela login estarão apresentados os campos de <b>email/senha</b> para preenchimento
A2	Ao clicar no botão de <b>entrar</b> será feita uma requisição ao <i>web service</i> para validar este usuário

E4	Eu como usuário gostaria de poder cadastrar um grupo de dispositivos
A1	Na tela de cadastro de grupo de dispositivos, deve ter os seguintes campos: <b>Nome do grupo</b>
A2	Ao clicar no botão de <b>salvar</b> será feita uma requisição ao <i>web service</i> para criar esse grupo

E5	Eu como usuário gostaria de vincular os dispositivos com a minha conta, havendo uma tela que realiza a leitura de QR Codes (que deverão estar colados aos dispositivos adquiridos)
A1	Ao abrir a tela um leitor de QR Code começará a escanear
A2	Após escanear o QR Code do dispositivo, é será feita uma requisição ao webservice para vincular o dispositivo ao grupo escolhido

E6	Eu como usuário gostaria de convidar um usuário para ser participante do meu grupo de dispositivos
A1	Na tela de dispositivos, no topo do menu possuirá um botão para convidar novos usuários ao grupo de dispositivos
A2	Ao clicar no botão, uma <i>popup</i> será aberta contendo o seguinte campo: <b>email</b> . Ao clicar em OK, será enviado ao webservice o email do usuário convidado
A2	Caso o email digitado não existir, deve apresentar uma mensagem de erro

E7	Eu como usuário gostaria de poder aceitar ou recusar um convite de participação de um grupo de dispositivos
A1	Na tela de convites de grupo de participante, deve ser listado os grupos que o usuário foi convidado
A2	Após clicar no botão <b>aceitar</b> , o usuário será incluso ao grupo de dispositivo, e o grupo será removido da lista
A3	Após clicar no botão <b>recusar</b> , o usuário não fará parte do grupo, e o grupo será removido da lista

E8	Eu como usuário gostaria de visualizar na tela principal os grupos de dispositivos relacionados a mim
A1	Ao abrir a tela principal do aplicativo, mostrar em uma <b>Grid</b> os grupos de dispositivos vinculados ao usuário
A2	Ao clicar em um grupo, abrirá uma nova tela mostrando os dispositivos que estão vinculados para aquele grupo em uma <b>Grid</b>

E9	Eu como usuário gostaria de visualizar os dispositivos dos grupos de dispositivos que sou participante
A1	Na tela de grupo deverá ser listado todos os grupos a qual o usuário for participante

E10	Eu como usuário administrador de um grupo de dispositivos gostaria de ligar/desligar o alarme do grupo
A1	Na tela de dispositivos deverá ter um botão para ligar/desligar o alarme

E11	Eu como usuário gostaria de receber uma notificação de alerta do alarme
A1	Após o sensor detectar um movimento, uma notificação será enviada ao servidor e posteriormente ao aplicativo
A2	Deverá aparecer uma notificação no topo do aplicativo informando sobre a movimentação
A3	Ao clicar na notificação, abrirá a tela do grupo do dispositivo que foi detectado o movimento

E12	Eu como usuário gostaria de poder identificar na lista de dispositivos quais dispositivos estão sem acesso à Internet
A1	Na <b>Grid</b> de dispositivos, apresentará um ícone que não há conexão no dispositivo caso o mesmo não estiver comunicando com o servidor

## APÊNDICE B – ESPECIFICAÇÕES DO BANCO DE DADOS

Tabela tb\_dispositivo: Armazena os dispositivos

cod_dispositivo	Identificador único do dispositivo
cod_tipo_dispositivo	Identificador único do tipo do dispositivo
cod_grupo_dispositivo	Identificador único do grupo de dispositivo que o dispositivo está vinculado
dat_ultima_requisicao	Data e hora da última requisição realizada pelo dispositivo

Tabela tb\_tipo\_dispositivo: Armazena os tipos de dispositivo

cod_tipo_dispositivo	Identificador único do tipo de dispositivo
str_descricao	Descrição do tipo de dispositivo

Tabela tb\_grupo\_dispositivo: Armazena os grupos de dispositivos criados pelo usuário.

cod_grupo_dispositivo	Identificador único do grupo de dispositivos
cod_usuario	Identificador único do usuário que criou o grupo
str_nome	Nome do grupo de dispositivos
bln_alarme	Informa se o alarme está ligado ou não
bln_alarme_ativado	Informa se o alarme está ativado (tocando) ou não

Tabela tb\_usuario: Armazena os usuários

cod_usuario	Identificador único do usuário
str_nome	Nome do usuário
str_email	E-mail do usuário
str_senha	Senha do usuário criptografada
bln_ativo	Informa se o usuário está ativo ou não. (Exclusão lógica)
str_token	Identificador temporário do usuário. (Serve para envio de notificações)

Tabela tb\_usuario\_grupo\_dispositivo: Armazena o os usuários convidados para os grupos

cod_grupo_dispositivo	Identificador único do grupo de dispositivo
cod_usuario	Identificador único do usuário convidado
bln_participante	Informa se o participante já aceitou ou não o convite