UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE
DEPARTAMENTO DE COMPUTAÇÃO

KEIFFER SCHMIDT

APLICAÇÃO MOBILE HÍBRIDA: SENTINELA

ALEGRE
2017

KEIFFER SCHMIDT

APLICAÇÃO MOBILE HÍBRIDA: SENTINELA VIANA

Trabalho de Conclusão de Curso de Sistema de Informação do Centro de Ciências exatas, naturais e da saúde da Universidade Federal do Espírito Santo, apresentado como requisito parcial para obtenção do Título de Bacharel em Sistema de Informação.

Orientador: prof. M.Sc. Valéria Alves da Silva.

ALEGRE
2017

KEIFFER SCHMIDT

APLICAÇÃO MOBILE HÍBRIDA: SENTINELA VIANA

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado em 15 de dezembro de 2017.

COMISSÃO EXAMINADORA

Prof.ª. M.Sc. Valeria Alves da Silva
Universidade Federal do Espírito Santo
Orientadora

Prof.ª. M.Sc. Fabricio Guedes Bissoli
Universidade Federal do Espírito Santo

Prof.ª. M.Sc. Jacson Rodrigues Correia da Silva
Universidade Federal do Espírito Santo

RESUMO

Este trabalho propôs apresentar um projeto criado para cidade de Viana, com intuito de ajudar a combater problemas vividos na cidade, tais como tráfico de drogas, crimes e acidentes de trânsito. Otimizando a busca e resolução dos mesmos através do uso da tecnologia em dispositivos mobiles, possibilitando denúncias pelo aplicativo ou pedidos de socorro para uma central de videomonitoramento e por meio do aplicativo poderá obter sua localização. Como o projeto é recente, ainda não podemos avaliar a contribuição do aplicativo no auxílio a segurança da população local.

Palavras-chave: *App, Javascript, Mobile, Cordova, Nodejs*.

ABSTRACT

This paper proposes to present a project created for the city of Viana, with the purpose of helping to combat problems lived in the city, such as drug trafficking, crime and traffic accidents. Optimizing the search and resolution of the same through the use of technology in mobile devices, allowing application denunciations or requests for help to a central video monitoring and through the application can obtain their location. As the project is recent, we still can not assess the contribution of the application to the safety of the local population.

Keywords: App, Javascript, Mobile, Cordova, Nodejs

LISTA DE FIGURAS

Figura 1 - Estrutura cordova....14
Figura 2 – exemplo uso plugin geolocation....15
Figura 3 - Exemplo de JavaScript....16
Figura 4 - Exemplo JQuery....16
Figura 5 – Página da UFES Alegre sem CSS....17
Figura 6 – Página UFES Alegre com CSS....17
Figura 7 - *Ajax method post*....18
Figura 8 - MDL....19
Figura 9 - Framework7....19
Figura 10 - Intel XDK....20
Figura 11 - Exemplo N*odejs*....21
Figura 12 - Exemplo *ExpressJS*....21
Figura 13 – lista *Json* enviada pelo servidor....22
Figura 14 - Ejs linguagem....22
Figura 15 - EJS resultado....22
Figura 16 - *Express upload*....23
Figura 17 - *Session*....24
Figura 18 - *Moment*....24
Figura 19 - *Sequelize define user*....25
Figura 20 - Ligação tabelas....26
Figura 21 - *PM2* gerenciador....26
Figura 22 - Comparação *mobiles*....28
Figura 23 – app de localização....29
Figura 24 - Caso de uso....32
Figura 25 - Esquema banco de dados....33
Figura 26 - Tela inicial de carregamento mobile....36
Figura 27 - Tela principal, opção de abrir menu ou alertar....36
Figura 28 - Ao clicar em qualquer opção, o aplicativo ligará para o número indicado na imagem....36
Figura 29 - Ao alertar irá para essa tela e será exibido opções de ocorrência....36
Figura 30 - Opção de escrever sobre a ocorrência e enviar o alerta!....37
Figura 31 – Tela inicial....37
Figura 32 - Menu....37

Figura 33 – tela Ocorrências.....38
Figura 34 – Tela ocorrência texto.....38
Figura 35 - Enviando.....38
Figura 36 – Sem conexão.....38
Figura 37 - Login de funcionários.....39
Figura 38 - Painel principal.....39
Figura 39 - Criar usuário funcionários.....40
Figura 40 - Listar usuários cadastrados.....40
Figura 41 - Lista ocorrências.....41
Figura 42 - Anotação sobre a ocorrência.....41

SUMÁRI

1. INTRODUÇÃO....10

1.1 O Problema e sua Importância....11

1.2 Objetivos....11

1.2.1 Objetivo Geral....11

1.2.2 Objetivos Específicos....11

2. REFERENCIAL TEÓRICO....13

2.1 CONCEITOS CLIENTE-SIDE....13

2.1.1 Cordova....13

2.1.4 *Google maps developer*....15

2.1.5 *JavaScript*....15

2.1.6 *JQuery*....16

2.1.7 CSS (*Cascading Style Sheets*)....17

2.1.8 AJAX....18

2.1.9 MDL(*getmdl.io*)....18

2.1.10 *Framework7*....19

2.1.11 Intel XDK....20

2.2 CONCEITOS SERVER-SIDE....20

2.2.1 *NodeJs*....20

2.2.2 *ExpressJS*....21

2.2.3 EJS....21

2.2.4 *Express-Fileupload*....22

2.2.5 *Express-session*....24

2.2.6 *Moment*....24

2.2.7 *MySql*....24

2.2.8 Mapeamento Objeto-Relacional(ORM)....25

2.2.9 *Sequelize*....25

2.2.10 *PM2*....26

2.2.11 *NPM*....26

2.2.12 *GitHub*....27

2.2.13 *GIT*....27

3. REVISÃO DE LITERATURA....28

4. METODOLOGIA.....30

4.1 Introdução.....30

4.2 Aplicação Mobile e Sistema de Gerenciamento.....30

4.2.1 Descrição das Aplicações (*front-end* e *server-side*).....30

4.2.2 Modelo Mobile.....31

5. RESULTADOS.....36

5.1 Breve descrição de uso *Android/IOS*.....39

5.2 Breve descrição de utilização do sistema.....42

5.3 Avaliação pelos Usuários.....42

5.3.1 Aplicação *Mobile*.....42

5.3.2 Sistema interno de controle Vídeomonitoramento.....42

6. CONCLUSÕES.....43

REFERÊNCIAS.....44

1. INTRODUÇÃO

Atualmente a insegurança vem causando preocupações para a população tanto em território nacional, como aqui no estado do Espírito Santo, seja essa insegurança relacionada a furtos, roubos, assassinatos, tráfico e até mesmo o trafego pela cidade.

> Os últimos números relacionados ao cenário nacional do tráfico de drogas no Brasil mantém o Espírito Santo como o líder nacional na taxa proporcional. Para cada 100 mil habitantes, 144 pessoas estão envolvidas do tráfico de entorpecentes, segundo dados do Anuário Nacional de Segurança Pública de 2014. Esses dados eram de 111 pessoas por 100 mil relacionadas ao crime. (ANUÁRIO BRASILEIRO DE SEGURANÇA PÚBLICA, 2017).

Muitos acidentes vêm ocorrendo ao longo dos anos seja por falta de investimento das prefeituras ou imprudência das pessoas.

> De acordo com o Detran-ES (2017) na Grande Vitória, aconteceram 26.801 acidentes de trânsito em 2015, sendo que 7.640, ou seja, 28,51% deles registraram vítimas. Dos acidentes com vítimas, 63,47% aconteceram durante o dia e 96,03% foram em área urbana. Grande parte são colisões/abalroamento, com 4.643 registros, seguido de atropelamentos, com 841 acidentes.

Uma maneira de contribuir seria minimizando o tempo de alerta sobre qualquer dessas ocorrências e a resposta para as mesmas, e com o avanço da tecnologia é possível utilizar aplicações mobile para denúncias ou pedidos de socorro para uma central de vídeomonitoramento onde serão registrados o local e o problema para uma tomada de decisão mais rápida possível junto às autoridades.

Este trabalho apresenta todo o processo para o desenvolvimento de uma aplicação híbrida e detalha algumas tecnologias e como foram utilizadas.

1.1 O Problema e sua Importância

Em todo o país observa-se a constante violência, furtos, assaltos e assassinatos. No Espírito Santo, por exemplo, a taxa de homicídios no ano de 2014 foi de 39,4 por 100 mil habitantes (ANUÁRIO BRASILEIRO DE SEGURANÇA PÚBLICA, 2017). Trata-se de uma taxa altíssima se comparada à de outros estados, situando o estado do Espírito Santo em oitavo lugar no Brasil, e sendo quase 4 vezes maior do que o índice aceitável pela Organização Mundial de Saúde (PEREIRA & GRASSI , 2012). Para lidar com a criminalidade é preciso do uso de todos os recursos possíveis e isso inclui a tecnologia. Uma possível estratégia seria a criação de um aplicativo (APP) mobile para denúncias de crime, tráfico entre outros problemas.

1.2 Objetivos

1.2.1 Objetivo Geral

Este trabalho tem por objetivo aplicar o uso de aplicativo *mobile* para contribuição da redução de crimes e tempo resposta a eventuais problemas, desde acidentes de trânsito a roubos na cidade de Viana. Criando uma aplicação para auxiliar a comunicação da sociedade com as autoridades competentes.

1.2.2 Objetivos Específicos

- Definir tecnologia para criação de servidor e APP *Mobile* (*back-end* e *front-end*);
- Realizar a modelagem do banco;
- Interligar o banco com aplicativo;
- Construir a aplicação para visualização e tratamento no centro de monitoramento

- Realizar os testes de usabilidade com o usuário.

Este trabalho está divido da seguinte forma: o Capítulo 2 apresenta um referencial teórico; o Capítulo 3 detalha a metodologia a ser seguida; o Capítulo 4 possui trabalhos relacionados; o Capítulo 5 apresenta os resultados e finalizando com o capítulo 6 que apresenta as conclusões do projeto.

2. REFERENCIAL TEÓRICO

Para criação do sistema de gerenciamento e o APP *Mobile* que serão explicados em tópicos posteriores, foi necessária uma pesquisa sobre como desenvolvê-los, e é possível encontrar muitos módulos, *frameworks* e tecnologias pra auxiliar em sua construção, há muitas alternativas para criação de *back-end* como exemplo, php, c# ,java, nodejs (javascript), e o mesmo se aplica para o *front-end* tendo inúmeros *frameworks* para tal finalidade. Nesse tópico será explicado as várias tecnologias que foram usadas ao longo do projeto.

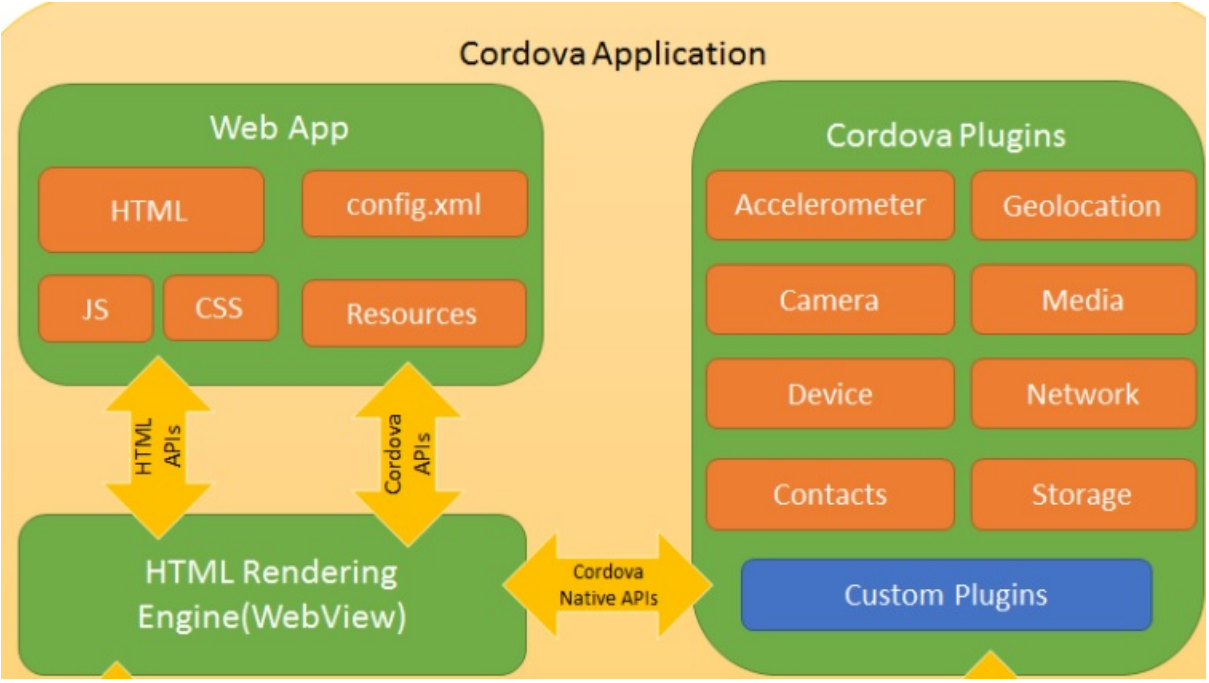
2.1 CONCEITOS CLIENTE-SIDE

Para criação da aplicação foi preciso escolher algum *framework* para desenvolvimento que se adequasse melhor as necessidades do problema e do desenvolvedor, há inúmeras possibilidades, como *Xamarin*, *java*, *cordova* entre outros, todos tem sua vantagem e desvantagem porém o que mais foi valorizado ao fazer uma escolhe foi o tamanho da comunidade e sua participação e se era multi-plataforma (híbrido) ou não. O *Xamarin Forms* apesar de ser híbrido não tem uma comunidade muito participativa, após uma análise empírica pelo desenvolvedor em fóruns dos mesmos, foi decidido pelo uso do *Cordova*.

2.1.1 Cordova

O *Cordova* foi doado para a Apache, pelo Adobe e teve sua origem do *Phonegap* sendo na prática a mesma coisa, é um *framework* para desenvolvimento mobile híbrido, ou seja, pode-se utilizar recursos de qualquer aparelho seja Android, IOS, *Windows Phone* e outros. Como todo aparelho celular vem com um componente denominado *Webview* ele funciona como se fosse um navegador mínimo com vários recursos. Logo o *Cordova* utiliza desse recurso para acessar as funcionalidades nativas do aparelho celular e também interpretar, interagir e exibir sua aplicação simplesmente construída com HTML, Javascript e CSS (CORDOVA, 2017). Na Figura 1 representa a estrutura de como é feita essa comunicação com cordova.

Figura 1 - Estrutura cordova



Fonte: Cordova(2017)

2.1.2 *WebView*

Através do *WebView* é possível fazer a comunicação de recursos nativos do celular com sua aplicação híbrida utilizando a linguagem WEB e também exibir no celular como se fosse uma página web porém sem abas de navegação e itens padrões de um navegador (CORDOVA, 2017).

2.1.3 PLUG-INS

Os *plug-ins* são fundamentais para o *Cordova* pois com estes é possível utilizar recursos nativos do aparelho celular como câmeras, GPS, Armazenamento entre outros. Na Figura 2 um exemplo da utilização do recurso do GPS após adicionar o *plugin Geolocation*, para funcionamento do *google maps* da aplicação (CORDOVA, 2017).

Figura 2 – Exemplo uso *plugin geolocation*

```
var mapOptions = {
        zoom: 5,
        mapTypeId: google.maps.MapTypeId.ROADMAP,
};
navigator.geolocation.watchPosition(function(position){
    map = new google.maps.Map(document.getElementById("map"), mapOptions);
    map.setCenter(new google.maps.LatLng(position.coords.latitude, position.coords.longitude));
    var latLong = new google.maps.LatLng(position.coords.latitude, position.coords.longitude);

    var marker = new google.maps.Marker({
        position: latLong
    });

    marker.setMap(map);
    map.setZoom(15);
    map.setCenter(marker.getPosition());
    },onError);
```

Fonte - O autor

2.1.4 *Google maps developer*

O *google* dispõe de uma API para uso de seu mapa com várias personalizações e funcionalidades basta criar um cadastro e gerar uma chave de uso para sua aplicação (GOOGLE,2017).

2.1.5 *JavaScript*

Javascript é linguagem de programação *front-end* e também *back-end,* sendo possível criar interações e modificações nas páginas por determinados eventos e também comunicações utilizando *Ajax* para endereços externos como de um servidor possui uma biblioteca denominada *Jquery* para minimizar código ao ser utilizado. A Figura 3 é um exemplo de trecho de código utilizando o *JavaScript* (ICTEA, 2017)..

Figura 3 - Exemplo de JavaScript.

```
1  <html>
2    <body>
3
4      <h2>titulo</h2>
5
6      <p id="exemploId">paragrafo a ser modificado</p>
7
8      <button type="button" onclick="myFunction()">botao</button>
9
10     <script>
11         function myFunction() {
12             document.getElementById("exemploId").innerHTML = "paragrafo alterado";
13         }
14     </script>
15
16   </body>
17 </html>
```

Fonte: O autor

2.1.6 *JQuery*

É uma biblioteca criada para um uso mais prático do *Javascript,* foi desenvolvida para deixar mais simples e pequenos os códigos. Existe a possibilidade de importar a biblioteca *JQuery* para seu projeto (JQUERY, 2017). A Figura 4 mostra um exemplo de *JQuery*:

Figura 4 - Exemplo JQuery.

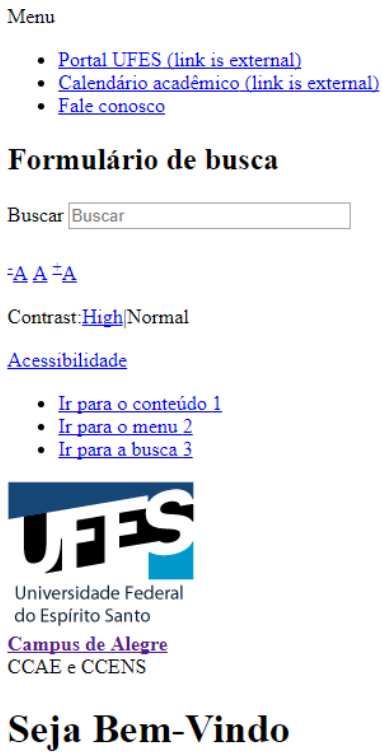
```
1  <html>
2    <head>
3      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
4    </head>
5    <body>
6
7      <h2>titulo</h2>
8
9      <p id="exemploId">paragrafo a ser modificado</p>
10
11     <button type="button" onclick="myFunction()">botao</button>
12
13     <script>
14         function myFunction() {
15             $("#exemploId").html("paragrafo alterado");
16         }
17     </script>
18
19   </body>
20 </html>
```

Fonte: O autor

2.1.7 CSS (*Cascading Style Sheets*)

CSS é uma linguagem de estilos, através dela é possível criar e aplicar *layouts* adicionando cores, tamanhos, formas entre outros no HTML (MDN WEB DOCS, 2017). A Figura 5 e 6 mostram um exemplo de CSS:

Figura 5 – Página da UFES Alegre sem CSS.



Fonte: O autor

Figura 6 – Página UFES Alegre com CSS.

Fonte: O autor

2.1.8 AJAX

AJAX (*Asynchronous JavaScript e XML*). É utilizado para comunicação do *client-side* para *server-side*, envia e recebe informações em diferentes formatos, podendo ser JSON, XML, HTML, e até mesmo no formato de arquivos de texto. (MDN WEB DOCS, 2017). Na Figura 7 podemos ver um exemplo.

Figura 7 - *Ajax method post*.

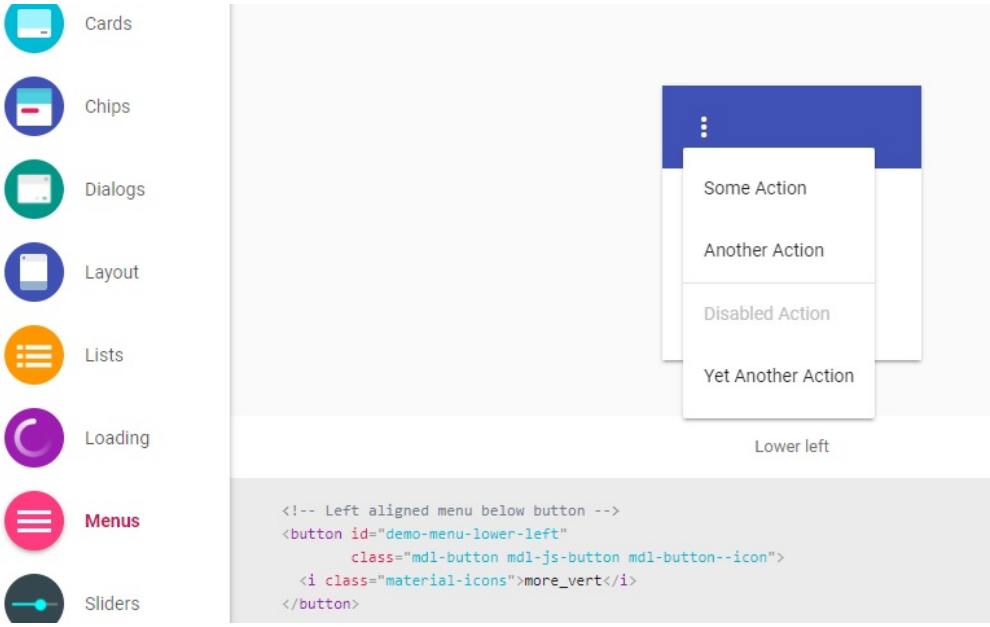
```
$("button").click(function(){
    $.post("demo_test_post.asp",
    {
        name: "Donald Duck",
        city: "Duckburg"
    },
    function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
});
```

Fonte: w3schools(2017)

2.1.9 MDL(*getmdl.io*)

O *Material Design Lite* (MDL) é um *framework front-end* para linguagem WEB, com ele é possível simplificar o trabalho para criação de interfaces. Não depende de *frameworks JavaScript* ou bibliotecas (GETMDL, 2017). A Figura 8 é um exemplo de componente do MDL:

Figura 8 - MDL

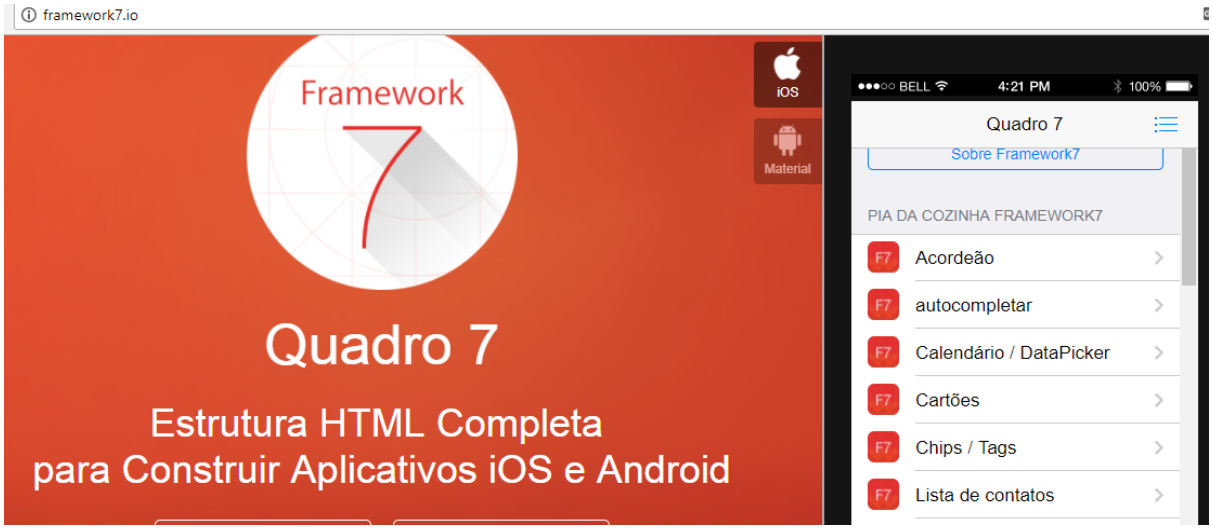


Fonte: Mdl.io(2017)

2.1.10 *Framework7*

Framework7 é um *framework* para *front-end* com uma grande variedade de componentes como botões, *cards*, menus próprios para interface Android e IOS (FRAMEWORK7, 2017). A Figura 9 mostra a logo do Framework7.

Figura 9 - Framework7.

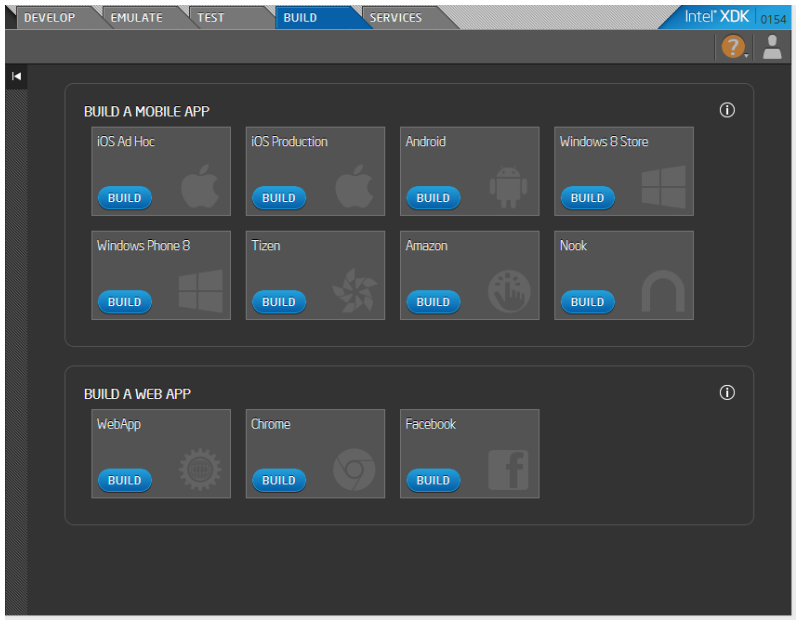


Fonte: Framework7(2017)

2.1.11 Intel XDK

O Intel Xdk é um IDE desenvolvido pela Intel facilitando o desenvolvimento para *mobile* web utilizando *cordova*, sendo possível *buildar apks* para *Android IOS*, *Windows phone* entre outros, com a possibilidade de simular sua aplicação sem a necessidade de *buildar* (ESTEVARENGO, 2016). Na Figura 10 uma representação do Intel Xdk.

Figura 10 - Intel XDK



Fonte: O autor

2.2 CONCEITOS SERVER-SIDE

2.2.1 *NodeJs*

Em 2009, Ryan Dahl, com o apoio de uma equipe, desenvolveu o *NodeJs*, possuindo um padrão inovador, é totalmente não bloqueante, tendo como vantagem o aproveitamento máximo de processamento e não cria gargalos por ser assíncrono. A linguagem utilizada para desenvolvimento é *Javascript* sendo orientado a eventos ou também chamado rotas (NODEJS, 2017). Também é possível simplificar, assim como em *Javascript* que tem a biblioteca *Jquery*, o *NodeJs* possui um *framework* para simplificação e facilitação de sua utilização, denominado *ExpressJS*. Na Figura 11 um trecho de código em *NodeJs*.

Figura 11 - Exemplo N*odejs.*

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello Node.JS!');
}).listen(8080);
console.log('Server running at http://localhost:8080/');
```

Fonte: O autor

2.2.2 *ExpressJS*

O ExpressJs é um *framework* para *NodeJS* com ele é possível simplificar e facilitar a utilização dos recursos e métodos HTTP e *middlewares* (EXPRESSJS, 2017). Na figura 12 um exemplo de trecho de código:

Figura 12 - Exemplo *ExpressJS*.

```
var express = require('express');
var app = express();

app.get('/', function (req, res) {
  res.send('Hello World!');
});

app.listen(3000, function () {
  console.log('Example app listening on port 3000!');
});
```

Fonte: ExpressJS(2017)

2.2.3 EJS

O EJS *(embedded javascript)* é uma linguagem utilizada no *cliente-side* com ele é possível injetar dados enviados pelo *server-side* e manipular no HTML apenas utilizando tags “<%= %>” (BITOVI, 2017). Nas Figuras 13 e 14 são exemplos de seu uso.

Figura 13 – lista *Json* enviada pelo servidor

```
{ título:        'Produtos de limpeza'
  suprimentos:   [ 'mop' , 'vassoura' ,
                 'duster' ]                }
```

Fonte: EJS(2017)

Figura 14 - Ejs linguagem.

```
<ul>
<% for(var i=0; i<supplies.length; i++) {%>
   <li><%= supplies[i] %></li>
<% } %>
</ul>
```

Fonte:EJS

Figura 15 - EJS resultado.

- mop
- broom
- duster

The result is shown here

Fonte: Ejs(2017)

2.2.4 *Express-Fileupload*

Middleware para carregar arquivos com ele é possível anexar fotos (NPMJS, 2017). Na Figura 16 um exemplo de uso.

Figura 16 - *Express upload*.

```
app.post('/upload', function(req, res) {
  if (!req.files)
    return res.status(400).send('No files were uploaded.');

  // The name of the input field (i.e. "sampleFile") is used to retrie
  let sampleFile = req.files.sampleFile;

  // Use the mv() method to place the file somewhere on your server
  sampleFile.mv('/somewhere/on/your/server/filename.jpg', function(er
```

Fonte: Brian White(2017)

2.2.5 *Express-session*

Com o *express-session* é possível criar sessões para os usuários, permitindo após login, manterem-se conectados por tempo limitado de acordo com a configuração (GITHUB- EXPRESSJS, 2017). Na figura 17 um exemplo de seu uso.

Figura 17 - *Session.*

```
var app = express()
app.set('trust proxy', 1) // trust first proxy
app.use(session({
  secret: 'keyboard cat',
  resave: false,
  saveUninitialized: true,
  cookie: { secure: true }
}))
```

Fonte: ExpressSession(2017)

2.2.6 *Moment*

É uma biblioteca *JavaScript* que possibilita manipular e exibir data e hora de várias maneiras. Na Figura 18 um exemplo de seu uso (MOMENTJS, 2017).

Figura 18 - *Moment.*

```
moment().format('MMMM Do YYYY, h:mm:ss a'); // September 2nd 2017, 12:58:55 pm
moment().format('dddd');                    // Saturday
moment().format("MMM Do YY");               // Sep 2nd 17
moment().format('YYYY [escaped] YYYY');     // 2017 escaped 2017
moment().format();                          // 2017-09-02T12:58:55-03:00
```

Fonte: Moment(2015)

2.2.7 *MySql*

Mysql é um sistema de gerenciamento banco de dados e possui como linguagem o SQL, através deste é possível criar esquemas de tabelas para organização, inserção e alteração de dados do sistema (ORACLE, 2017).

2.2.8 Mapeamento Objeto-Relacional(ORM)

Quando se fala em sistemas não se pode esquecer do banco de dados tanto sua criação quanto sua segurança.

No trabalho de (MARQUES e JUNIOR, 2014), é citado (BAUER e KING, 2014) garantindo o beneficio da utilização de uma ORM pelas seguintes características: produtividade, segurança, manutenibilidade e independência do fornecedor do banco de dados, garantindo a abstração do banco de dados, redução de código, e cria uma camada a mais de filtros e validações antes de se comunicar com o banco de dados e por último deve garantir a facilidade e reuso do código caso troque o banco de dados.

ORM é um processo de mapeamento objeto relacional que faz uma relação dos modelos com os dados que os mesmos constituem. Alguns exemplos de ORM são o *doctrine, hibernate, sequelize* entre outros.

2.2.9 *Sequelize*

Sequelize é uma ORM baseada em promessas para *Node.js* v4 e posteriores. Ele suporta os dialetos *PostgreSQL, MySQL, SQLite* e *MSSQL* e possui suporte sólido de transações, relações, replicação de leitura e muito mais (SEQUELIZE, 2017). Na Figura 19 e 20 tem um exemplo de uso:

Figura 19 - *Sequelize define user.*

```
module.exports = function(sequelize, DataTypes) {
var Usuario= sequelize.define("Usuario", {
  login: DataTypes.STRING,
  nome: DataTypes.STRING,
  senha: DataTypes.STRING,
  cpf: DataTypes.STRING,
  matricula: DataTypes.STRING,
  email: DataTypes.STRING,
  password_digest: DataTypes.STRING
},
```

Fonte: Sequelize(2017)

Figura 20 - Ligação tabelas.

```
classMethods: {

  associate: function(models) {
   Usuario.hasMany(models.Historico),
   Usuario.hasMany(models.Detalhe),
   Usuario.belongsTo(models.Perfil, {
      onDelete: "CASCADE",
      foreignKey: {
        allowNull: false
      }
    })
 }
```

Fonte: Sequelize(2017)

2.2.10 *PM2*

O *PM2* é um gerenciador de aplicação *Nodejs*, serve para manter o funcionamento do servidor mesmo ao fechar o terminal (PM2, 2017). Na Figura 21 uma representação do mesmo.

Figura 21 - *PM2* gerenciador.

```
→ ping.js git:(master) pm2 list
```

App name	id	mode	pid	status	restart	uptime	memory	watching
ping	0	fork	2090	online	0	16s	23.465 MB	disabled

Fonte:pm2(2017)

2.2.11 *NPM*

NPM é um gerenciador de pacotes do *NodeJS*, qualquer componente, biblioteca que necessite utilizar no projeto você pode simplesmente importar através dele (PEREIRA, 2013).

2.2.12 *GitHub*

É uma plataforma de hospedagem de código-fonte que utiliza o *Git*. Promove projetos *open-source* pois permite contribuição de outros desenvolvedores em vários projetos e utilização dos mesmos (GITHUB, 2017).

2.2.13 *GIT*

É utilizado para gerenciar versões de aplicações, e compartilhamento de código (GIT, 2017).

3. REVISÃO DE LITERATURA

No trabalho de (PREZOTTO e BONIATI, 2014) utilizou-se *Cordova* para seu desenvolvimento, onde foi criado um aplicativo de controle de presença de sala de aula, na aplicação desenvolvida é possível cadastrar alunos e turmas, para em seguida criar uma aula e marcar os alunos presentes. No cadastro dos alunos é possível tirar uma foto do mesmo para uma melhor identificação durante o processo da chamada. Para obter acesso às funcionalidades da aplicação o professor deve primeiramente fazer o *login* na aplicação, este acesso ficara disponível a ele até que seja feito o *logoff*.

Foi utilizado para criação do APP, linguagens como *HTML, CSS* e *JavaScript*. Além disso para armazenamento simples foi utilizado *Web SQL* e *LocalStorage*. Sendo utilizado um *framework* híbrido foi testado em 4 plataformas: *Android, Firefox OS*, *Windows Phone e Blackberry* (PREZOTTO e BONIATI, 2014).

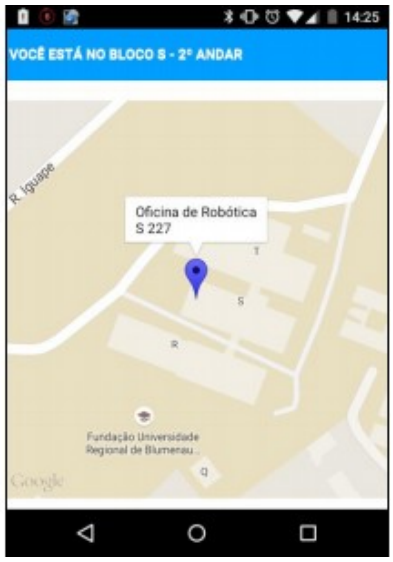
A conclusão do projeto foi que houve poucas mudanças na interface entre uma plataforma e outra, porém deve-se pensar em utilizar outra base de dados, pois o *Windows phone* não oferece suporte ao *Web Sql* (PREZOTTO e BONIATI, 2014). Na Figura 22 uma representação de seu aplicativo em diferentes sistemas.

Figura 22 - Comparação *mobiles*.

Fonte: Prezotto(2014)

No trabalho de (TONDIM,2015), foi criado um aplicativo para localização interna do usuário na Universidade FURB, onde foi utilizado o *PhoneGap* para utilização de linguagem web e funcionalidades *Geolocation* do celular, permitindo aos usuários obterem informações em sua localidade e facilitando a locomoção no mesmo. Na Figura 23, temos o resultado de seu trabalho:

Figura 23 – app de localização



Fonte: TONDIM(2015)

Seus resultados foram positivos, após testes com usuários foi possível ter o retorno dos mesmos quanto a usabilidade e 30% dos usuários não acharam a aplicação intuitiva contra 70% que não tiveram grandes dificuldades.

4. METODOLOGIA

4.1 Introdução

Inicialmente houve um estudo para criar um APP e quais tecnologias seriam de maior utilidade aos requisitos. Adicionada a curiosidade do desenvolvedor para criação de uma aplicação híbrida que seria desenvolvida tanto para Android como para IOS. Foi necessário a utilização do *Cordova* para acesso de funções nativas do celular, tais como, GPS, câmeras, chamadas, entre outros. Também foi feito a utilização de uma ferramenta que oferecesse rápido desenvolvimento, um emulador que simula-se alguns modelos de celulares e funções nativa. O *Intel XDK.*

Como descrito anteriormente houve a necessidade de um sistema interno para controle, criação de relatórios e visualização dos alertas. Foi desenvolvido um sistema web utilizando também *HTML*, *JavaScript, JQuery, CSS*.

O servidor que gerencia o *APP Mobile* e o sistema web foi desenvolvido utilizando *NodeJS*, aplicando tecnologias como *ExpressJS* para criação de rotas, uma ORM e também foi utilizado uma estrutura esqueleto para melhor organização e boas práticas do sistema e outras dependências.

Em todas as etapas do trabalho houve um controle de versões utilizando o *GIT* e também compartilhamento com o supervisor do projeto na prefeitura de Viana para análise e aprovações do mesmo.

4.2 Aplicação Mobile e Sistema de Gerenciamento

4.2.1 Descrição das Aplicações (*front-end* e *server-side*)

Para criar uma aplicação de denúncia de algum crime, seja tráfico de drogas até alguma briga ou roubo, inicialmente foi criado uma versão apenas para Android porém foi necessário compilar uma nova versão para IOS, pois houve demanda. Outra necessidade surgida foi o desenvolvimento de um sistema para gerenciamento das ocorrências enviadas pelo mobile.

4.2.2 Modelo Mobile

Para construção do mobile houve uma pesquisa para os requisitos e recursos encontrados, foi utilizado alguns *plugins,* como GPS, conexão 3g e *wifi* para utilização da API do *Google Maps* e por fim, um *plugin* chamado *whitelist* para autorizar o envio de dados através de *Ajax* ao servidor. Utilizando a linguagem *WEB* para desenvolvimento onde está incluso *JavaScript/JQuery* para manipulação do *HTML* e *CSS* para o design. Para melhor produção foi feito uma pesquisa por um *framework* para *front-end* com componentes prontos. Decidiu-se a utilização do *Material Design Lite* por sua simplicidade e facilidade, possui Menus, Listas, *cards*, botões, *loads* entre outros. Apesar de não ter sido desenvolvido como um *framework* com a intenção de ser usado propriamente para *mobile* e sim para páginas *web*, serviu para o propósito do aplicativo para *Android*. Também foi preciso de um ambiente de desenvolvimento Integrado (IDE) para emular e compilar o aplicativo e testar em diferentes plataformas, foi utilizado então a plataforma Intel XDK.

4.2.2.1 Requisitos Funcionais

Após diálogo com os *stackholders*, foi coletado 5 requisitos. No Quadro 1 uma representação dos mesmos.

Quadro 1 - Requisitos Funcionais

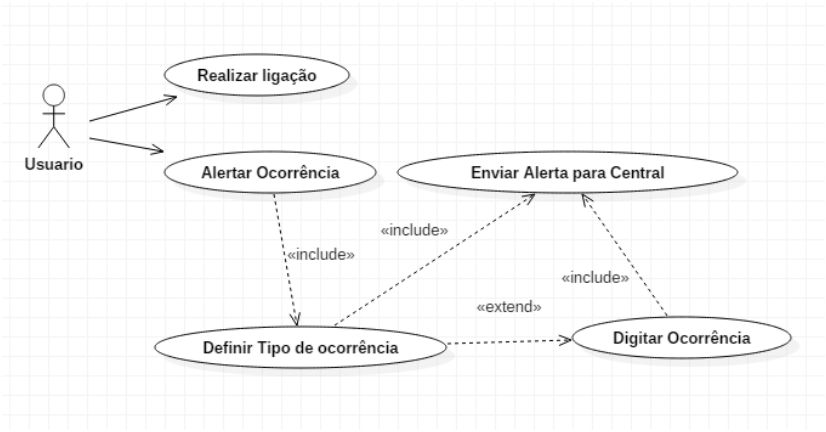
ID	NOME	DESCRIÇÃO
RF001	Realizar Ligação	Ao abrir o menu lateral pode selecionar uma opção para ligação
RF002	Alertar Ocorrência	O aplicativo deve capturar latitude e longitude do usuário
RF003	Definir tipo Ocorrência	Será possível armazenar um tipo de ocorrência
RF004	Digitar ocorrência	O aplicativo possibilita digitar um texto sobre a ocorrência
RF005	Enviar alerta para Central	Será enviado os dados capturados para central

Fonte: O autor

4.2.2.2 Diagrama de Caso de Uso

Após a definição dos requisitos foi possível gerar um diagrama de caso de uso conforme a Figura 24 mostra.

Figura 24 - Caso de uso



Fonte: O autor

4.2.3 Sistema de Gerenciamento

Muitas perguntas foram feitas antes de começar seu desenvolvimento e uma delas foi " como a equipe de segurança da prefeitura teria facilidade e organização para seu uso? " Após várias conversas foi possível desenvolver uma base sólida e começar a sua criação.

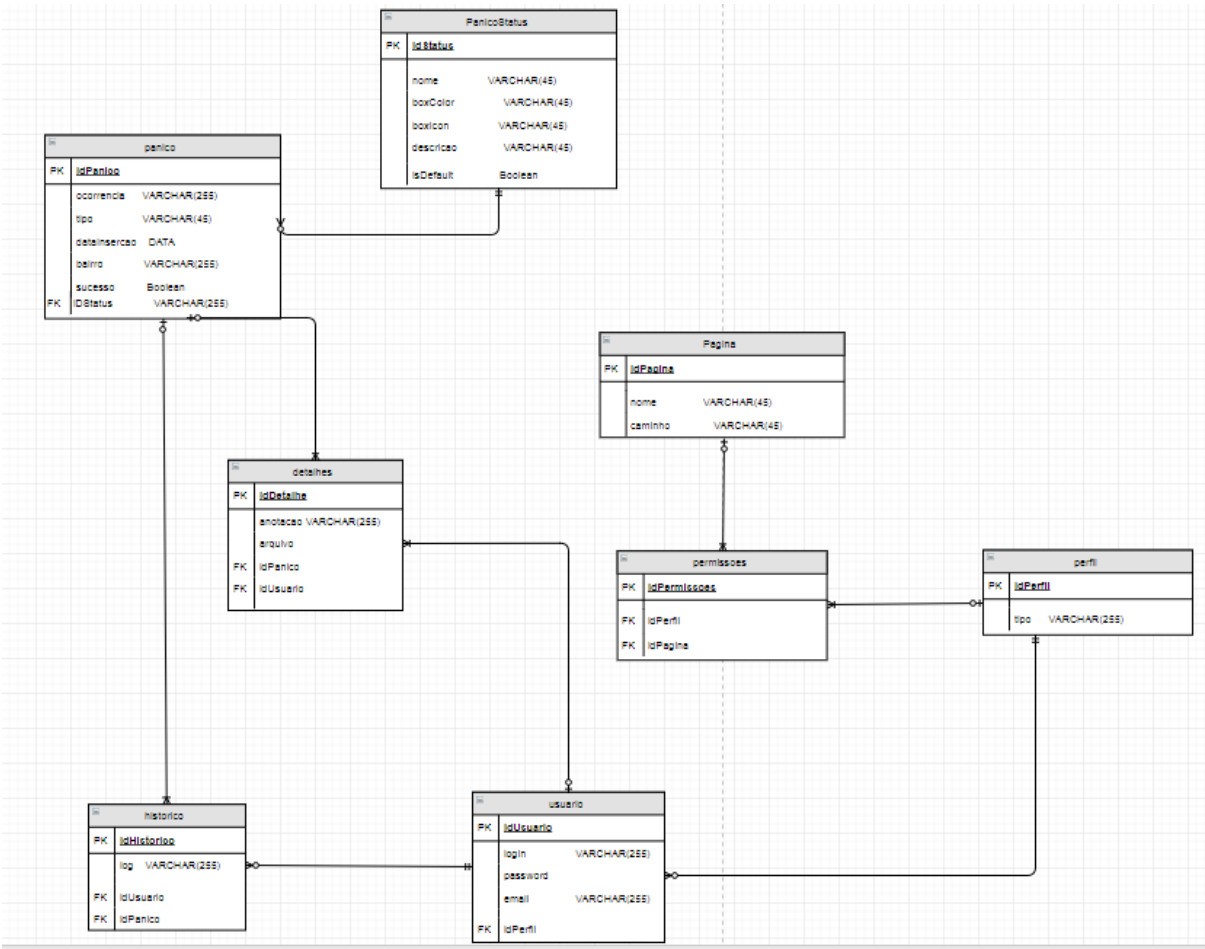
Foi utilizado o mesmo *server-side* e banco de dados do mobile, e seu *front-end* foi criado com *HTML, Javascript/Jquery, EJS, CSS* e um *framework* de auxílio na criação (*Admlte*) que possibilitou uma rápida produção e interface limpa e intuitiva ao usuário.

Após a criação do *fron-end mobile* foi preciso de uma busca sobre possíveis linguagens *server-side*, inicialmente foi pensado no uso do PHP por questões de conhecimento, porém, após algumas pesquisas e dicas de outro desenvolvedor foi visto uma nova linguagem usando *javascript,* possuindo uma grande comunidade (*stackoverflow, github*), por questão de curiosidade e entusiasmo foi então decidido o uso do *nodeJS*.

Assim, surgiu a necessidade de um entendimento básico sobre seu funcionamento e praticar inúmeros exemplos básicos. Após aprendizado foi necessário utilizar vários módulos do *Github* referentes à *NodeJs* para aumento de produção. Mesmo com isso o projeto *server-side* estava ficando desorganizado, então foi preciso criar ou encontrar um esqueleto para o projeto definindo padrões, logo, foi utilizado um esqueleto que envolvia *NodeJS-express* e uma ORM nomeada *Sequelize*, depois de definido uma estrutura e qual ORM seria utilizada precisava de um banco de dados e foi definido e instalado o *Mysql*.

A ORM dá uma liberdade de futuramente, se necessário, fazer a troca do banco de dados sem grandes problemas. O próximo passo agora era a criação do modelo do banco, na Figura 25 mostra o modelo relacional, onde existem 8 tabelas que serão descritas no tópico 4.2.3.1.

Figura 25 - Esquema banco de dados.



Fonte: o autor

4.2.3.1 Descrição Modelo Relacional

Tabela *usuário*

Um usuário possui os seguintes campos:

- *login* – *login* para acesso ao sistema.
- *password* - senha para acesso ao sistema.
- *email* – email do usuário.

Tabela *perfil*

Um usuário possui um perfil:

- Tipo: pode ser perfil “Administrador” ou “funcionário”.

Tabela *permissão*

Um perfil possui permissão para uma ou mais paginas

- *idPerfil* – identificação do perfil
- *idPagina* – identificação da página

Tabela *pagina*

Uma página tem acesso por um ou mais perfis

- *Nome* – nome da página exemplo, “cadastro usuário”.
- *Caminho* – caminho da rota da página exemplo, “/usuário/cadastro”

Tabela *panico*

Uma ocorrência possui os seguintes campos:

- *Ocorrência* – texto enviado pelo usuário de uso do mobile
- *Tipo* – assalto, incêndio, agressão, acidente, sequestro, tráfico de drogas e outros.
- *dataInsercao*- data de envio da ocorrência.
- *Bairro* – bairro e rua de envio da ocorrência.
- *Latitude* – latitude da ocorrência.
- *Longitude* – longitude da ocorrência.

Tabela *panicoStatus*

Uma ocorrência possui status com os seguintes campos:

- *Nome* – nome do status, “Novo”, “Analisando”, “Concluído”.
- *boxColor* – cor da box .
- *boxIcon* – ícone da box.
- *Descrição* – descrição da box.
- *isDefault* – se é valor inicial da ocorrência.

Tabela detalhes

Uma ocorrência pode ter anotações por parte dos usuários e possui os seguintes campos:

- *anotação* - relatório sobre a ocorrência.
- *arquivo* - caminho da imagem anexado pelo usuário.

Tabela histórico

Possui histórico relacionado com usuário e a ocorrência contendo os seguintes campos:

- *log* – campo com descrição da ação do usuário exemplo, “fez anotação da ocorrência 25 “, “marcou como concluída a anotação”
- *idUsuario* – identificação do usuário.
- *idPanico* – identificação da ocorrência.

5. RESULTADOS

As Figuras 26 ao 30 mostram o *APP mobile Android* já pronto e publicado:

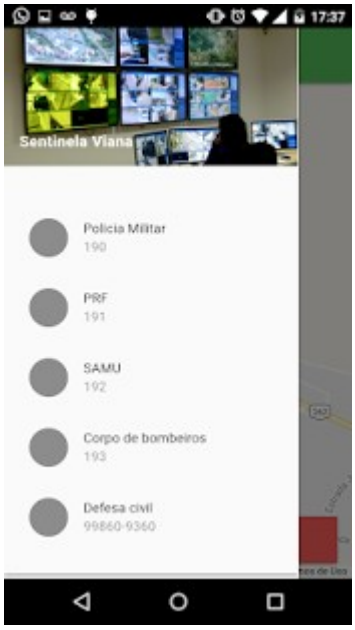
Figura 26 - Tela inicial de carregamento mobile.

Fonte: O autor.

Figura 27 - Tela principal, opção de abrir menu ou alertar.

Fonte: O autor.

Figura 28 - Ao clicar em qualquer opção, o aplicativo ligará para o número indicado na imagem.

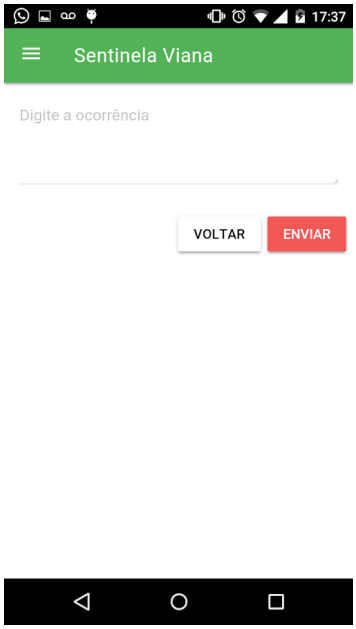


Fonte: O autor.

Figura 29 - Ao alertar irá para essa tela e será exibido opções de ocorrência.

Fonte: O autor.

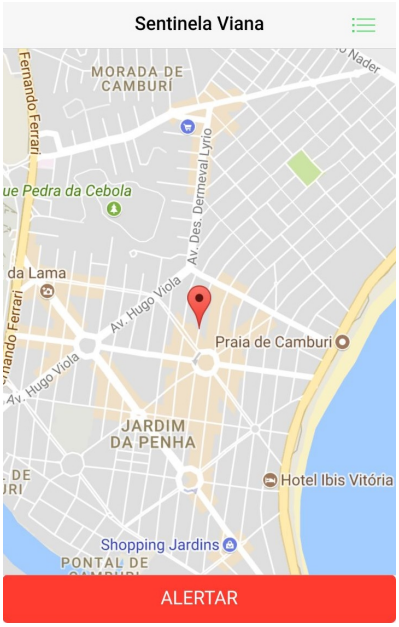
Figura 30 - Opção de escrever sobre a ocorrência e enviar o alerta!



Fonte: O autor.

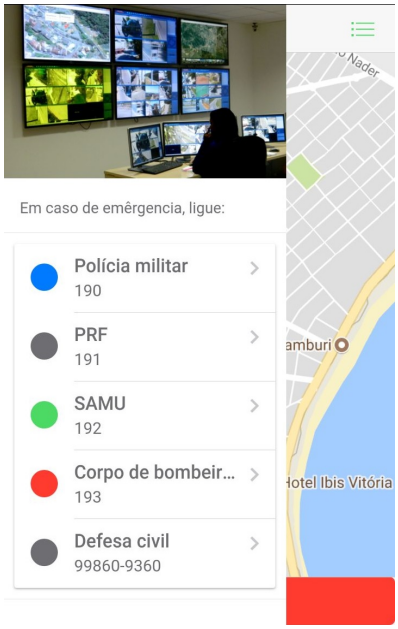
As figuras 31 ao 36 mostram o *APP mobile IOS* já pronto:

Figura 31 – Tela inicial.



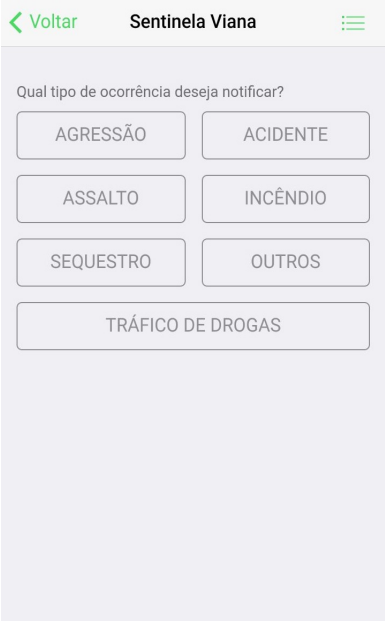
Fonte: O autor.

Figura 32 - Menu.



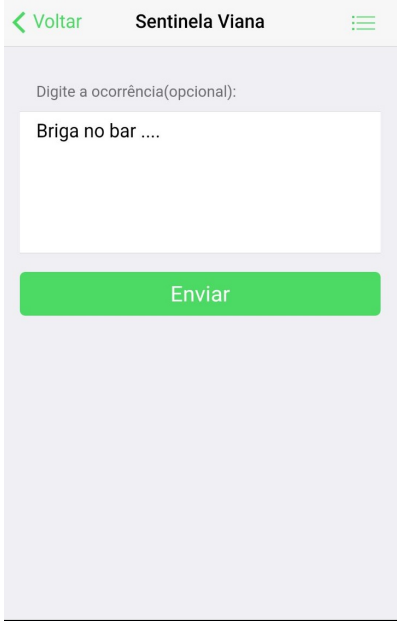
Fonte: O autor.

Figura 33 – tela Ocorrências.



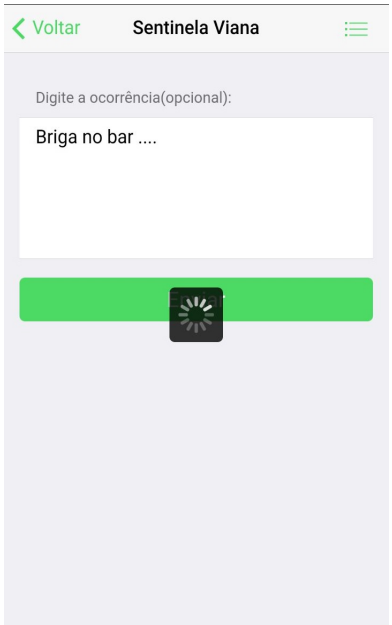
Fonte: O autor.

Figura 34 – Tela ocorrência texto.



Fonte: O autor.

Figura 35 - Enviando.



Fonte: O autor.

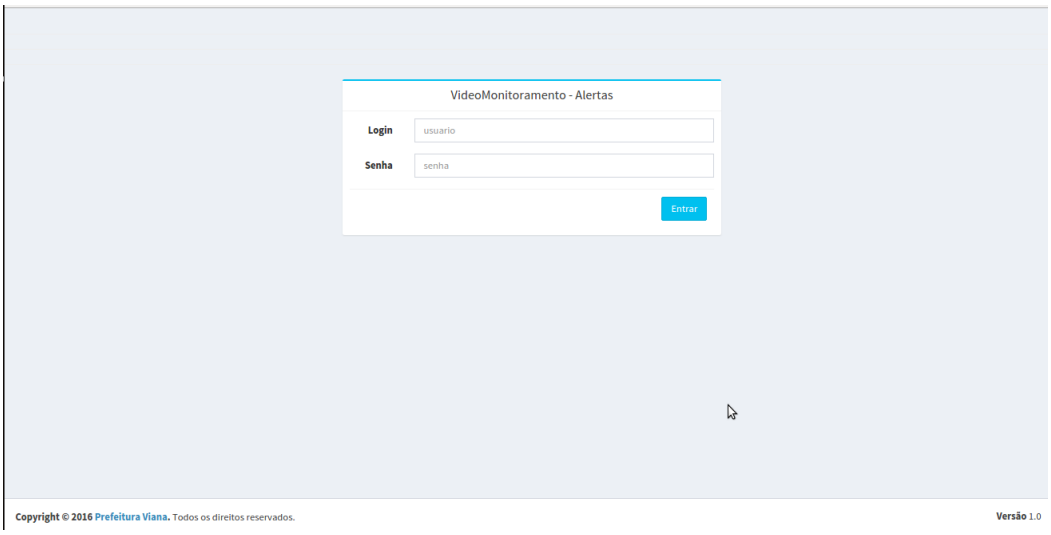
Figura 36 – Sem conexão.

Fonte: O autor.

5.1 Breve descrição de uso *Android/IOS*

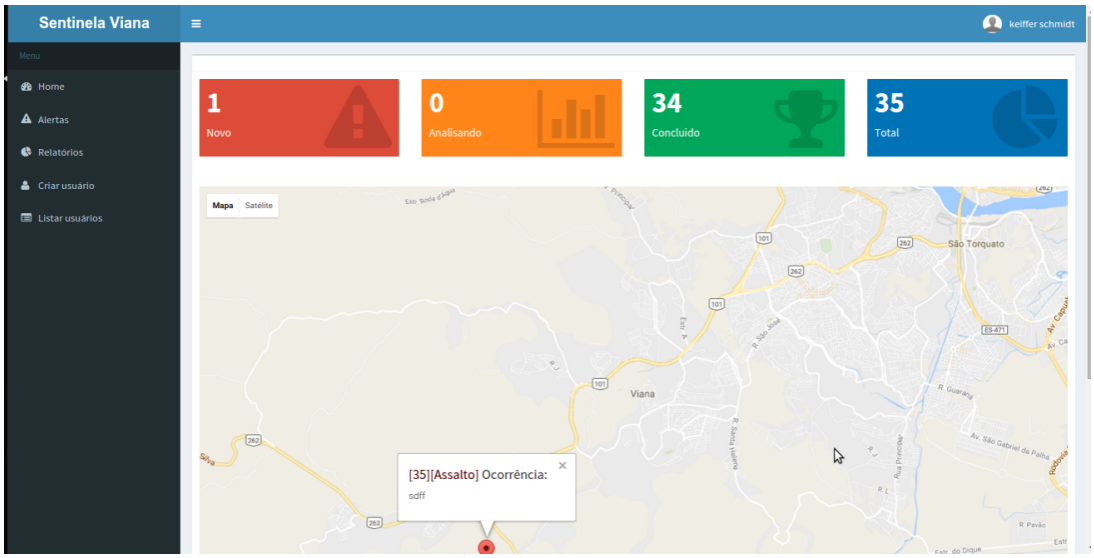
O usuário ao abrir o APP mobile, terá marcado a localização do mesmo através do GPS e *Google Maps* (figura 24), ele tem a opção de abrir o menu (figura 25) ou clicar em “alertar” que seguirá para a figura 26, deve-se então escolher uma opção e em seguida irá para figura 27, o usuário também poderá escrever sobre a ocorrência ou apenas enviar, se houver conexão exibirá “notificação enviada com sucesso” caso contrário avisará o usuário que não foi possível enviar. As Figuras de 37 a 42 mostram o sistema interno do vídeomonitoramento:

Figura 37 - Login de funcionários.



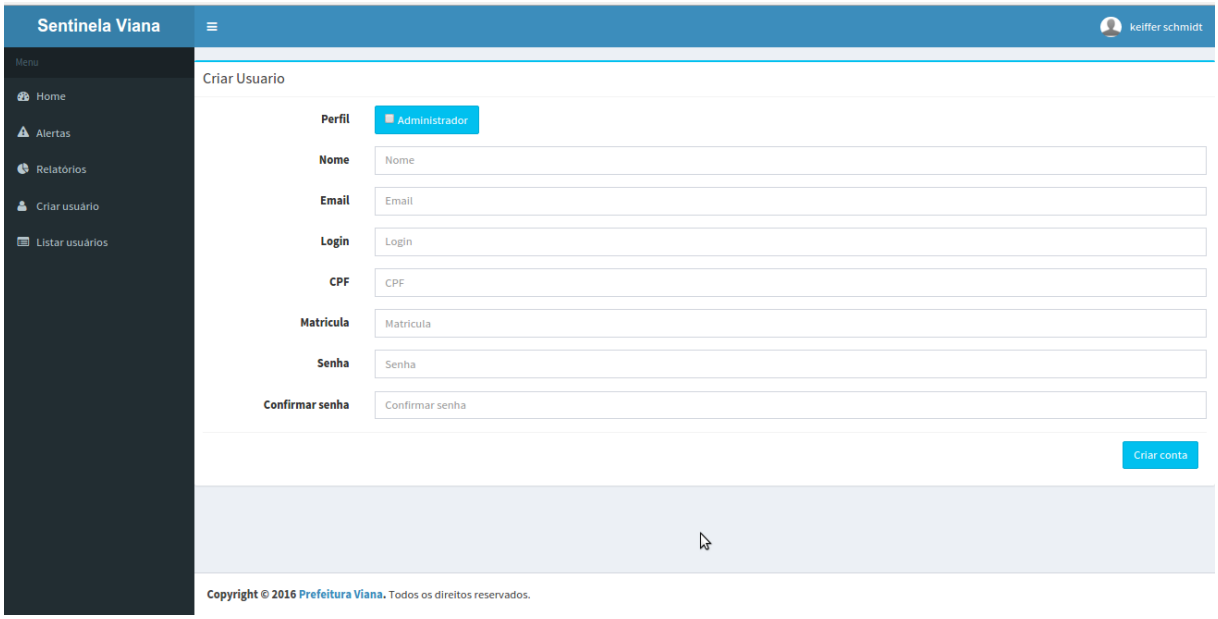
Fonte: O autor.

Figura 38 - Painel principal.



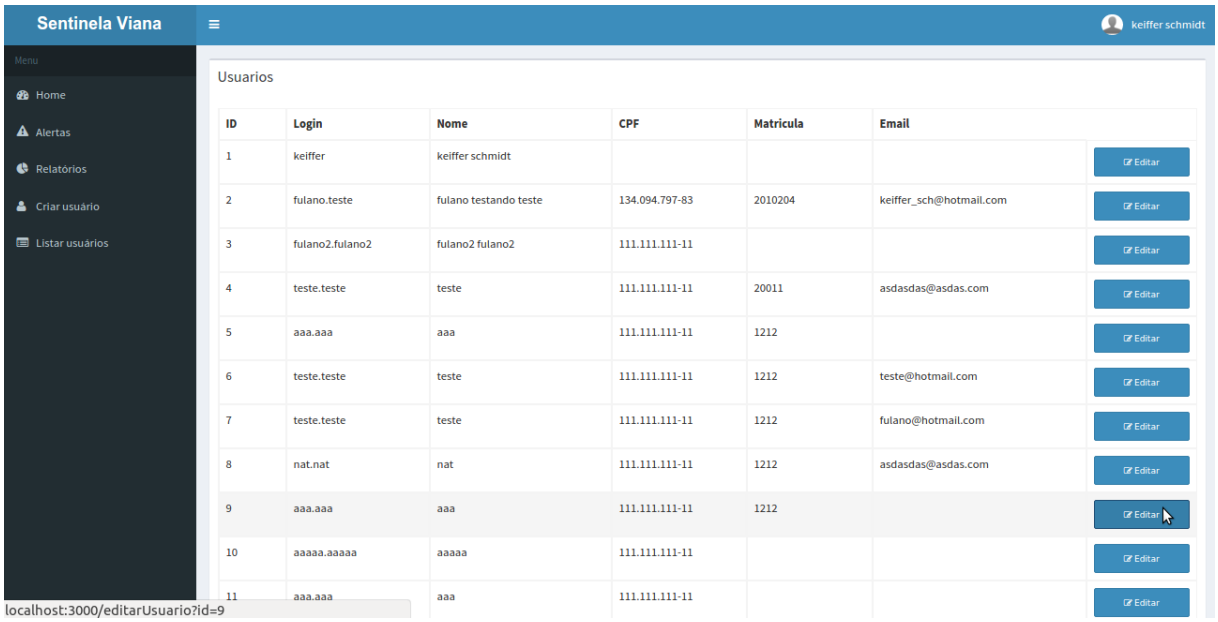
Fonte: O autor.

Figura 39 - Criar usuário funcionários.



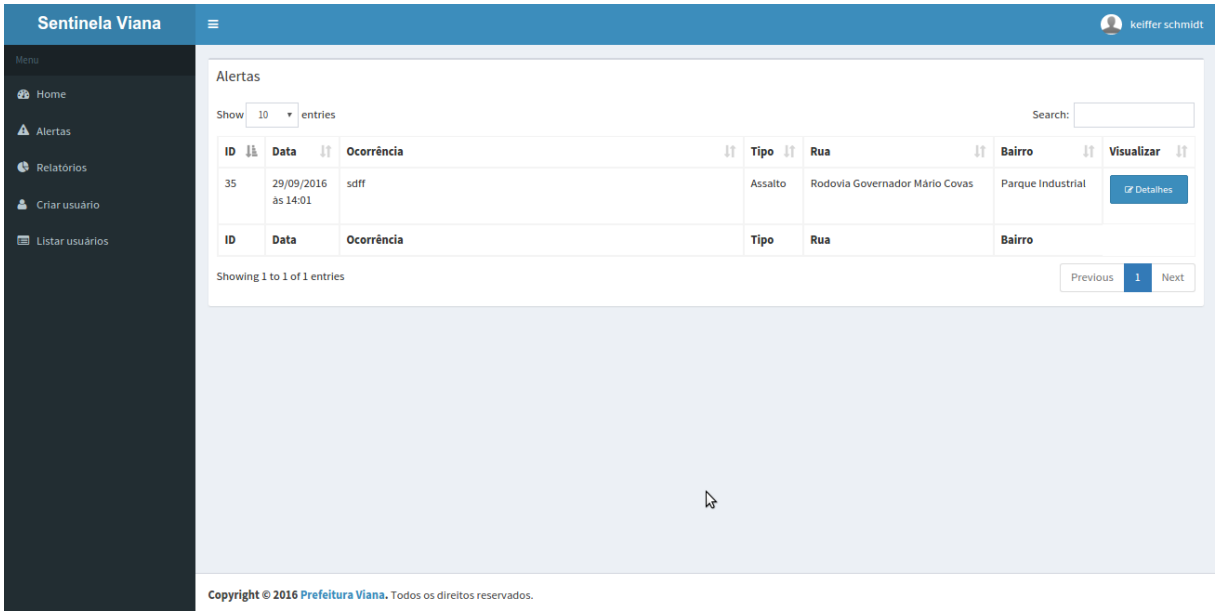
Fonte: O autor.

Figura 40 - Listar usuários cadastrados.



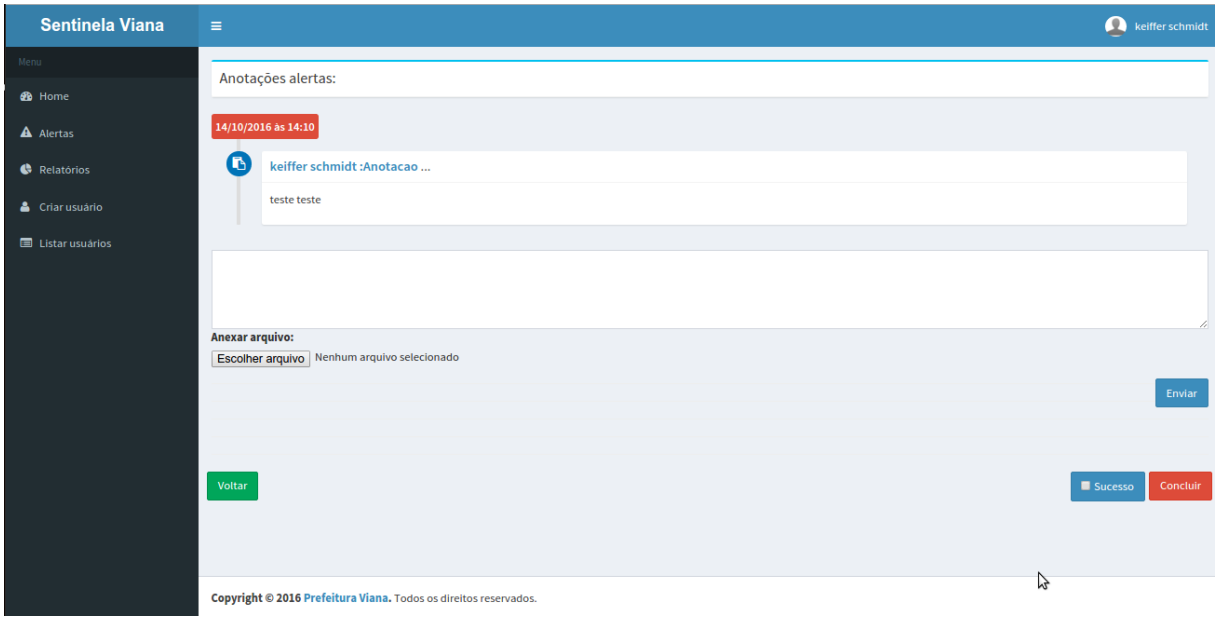
Fonte: O autor.

Figura 41 - Lista ocorrências.



Fonte: O autor.

Figura 42 - Anotação sobre a ocorrência.



Fonte: O autor.

5.2 Breve descrição de utilização do sistema

O usuário irá se autenticar no sistema, assim que houver um alerta será exibido no mapa, juntamente, com a localização, sendo esse alerta detalhado na listagem de “alertas”, possuindo atributos como” tipo de alerta”, “descrição” e endereço através da latitude e longitude, o usuário poderá criar anotações do que foi feito com aquela ocorrência, se identificaram nas câmeras ou não e o que foi feito, podemos também criar novos usuários, caso a autenticação tenha sido feita por um administrador e listar todas as notificações já fechadas.

5.3 Avaliação pelos Usuários

5.3.1 Aplicação *Mobile*

Houve um período de testes para verificar o quão simples seria para os usuários e sua usabilidade, durante o projeto houve melhorias em relação a interface, e por se tratar de uma aplicação simples e com poucas telas não houve grandes dificuldades tendo o retorno positivo por grande parte dos usuários.

5.3.2 Sistema interno de controle Vídeomonitoramento

Foram feitos inúmeros ajustes para melhoria do sistema pois era mais complexo tanto seu planejamento quanto seu uso, pois possui mais telas e interação com o usuário, dialogado muitas vezes com os próprios usuários para adaptar o sistema a eles e deixar da maneira mais clara possível, depois de pronto foram realizados vários treinamentos com os usuários. O retorno dos treinamentos foi positivo e acelerou o trabalho junto as câmeras da cidade e deixou o serviço mais confiável para criação de relatório e anexos de imagens.

6. CONCLUSÕES

Para criar o aplicativo *mobile* proposto nesse trabalho, o uso do *Cordova* , foi de fácil compilação para as plataformas *Android e IOS*. A utilização da *Linguagem Web* foi produtivo, pois há inúmeros módulos na internet para uso aberto, tendo uma comunidade participativa e com muitos colaboradores.

O *NodeJS* se mostrou muito útil, embora o ambiente seja assíncrono e orientando a eventos, ele torna o *back-end* mais rápido e de fácil manipulação, depois de um certo tempo de experiência.

Os usuários do Aplicativo *Mobile* relataram ser fácil e intuitivo, o uso do mesmo, sendo um dos pontos principais dos requisitos. Os usuários do sistema interno de vídeomonitoramento, após um treinamento relataram ser muito eficiente e prático. Ambos *APP* e sistema interno tiveram algumas críticas construtivas, e em breve terão melhoramento, partindo para uma versão 2.0.

O *APP* está reforçando a ações para aumentar a segurança no município de Viana, em conjunto com a central de vídeomonitoramento, auxiliando no reporte rápido das ocorrências de violência e no registro de um maior número de denúncias.

Esperamos que a sociedade contribua com o uso do APP e que outras prefeituras possam utilizar desse recurso, surgindo assim mais investimento e estudos na área de tecnologia.

REFERÊNCIAS

ANUÁRIO BRASILEIRO DE SEGURANÇA PÚBLICA. **Fórum Brasileiro de Segurança Pública**. Pinheiros–SãoPaulo - SP - 2015. Disponível em: < http://www.agenciapatriciagalvao.org.br/dossie/wp-content/uploads/2015/10/9-Anuario-Brasileiro-de-Seguranca-Publica-FSB_2015.pdf >. Acesso em: 15 de setembro 2017.

BAUER, C; KING, G. **Hibernate in Action**, Greenwich: Manning Publications, 2005. 408 p.

BITOVI. **An open source JavaScript Template library**, 2017. Disponível em: <http://www.embeddedjs.com/>. Acesso em 19 de setembro de 2017.

CORDOVA. **Overview**, 2017. Disponível em: <http://cordova.apache.org/docs/en/latest/guide/overview/index.html>. Acesso em: 14 de setembro de 2017.

DETRAN-ES. **Detran|ES divulga anuário com redução de acidentes**, 2017. Disponível em: < https://detran.es.gov.br/3108---detranes-divulga-anuario-com-reducao-d>.Acesso em: 14 de setembro de 2017.

ESTEVARENGO, L. F. **Desenvolvendo Jogos Mobile com HTML5**. São Paulo: Novatec Editora, 2016.

EXPRESSJS. **Express Framework web rápido, flexível e minimalista para Node.js**, 2017. Disponível em: <http://expressjs.com/pt-br/>. Acesso em: 16 de setembro de 2017.

EXPRESSSESSION. **Express session**, 2017. Disponível em: <https://github.com/expressjs/session>. Acesso em: 19 de setembro de 2017.

FRAMEWORK7. **Framework7**, 2017. Disponível em: <https://framework7.io/>. Acesso em: 16 de setembro de 2017.

GETMDL. **Material Design Lite**, 2017. Disponível em: <https://getmdl.io/>. Acesso em: 16 de setembro de 2017.

GIT. **Getting Started - A Short History of Git**, 2017. Disponível em: <https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git>. Acesso em: 14 de setembro de 2017.

GOOGLE. **Google Maps para todas as plataformas**, 2017. Disponível em: <https://developers.google.com/maps/>. Acesso em: 17 de setembro de 2017.

GITHUB. **The world's leading software development platform**, 2017. Disponível em: <https://github.com/>. Acessa em: 14 de setembro de 2017.

ICTEA. **O que é a linguagem de programação JavaScript?**, 2017. Disponível em: <http://www.ictea.com/cs/knowledgebase.php?action=displayarticle&id=8801&language=portuguese-pt>. Acesso em: 14 de setembro de 2017.

JQUERY. **What is jQuery?**, 2017. Disponível em: <http://jquery.com>. Acesso em: 20 de setembro de 2017.

MARQUES, L. H. C; JÚNIOR, D. J. L. R. **Aplicação do ORM Doctrine para abstração de banco de dados no desenvolvimento de software em PHP na UNASUS/UFMA**. Jornada de Informática do Maranhão, [S.L], 2014.

MDN WEB DOCS. **CSS**, 2017. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acesso em: 14 de setembro de 2017.

MDN WEB DOCS. **HTML**, 2017. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em: 14 de setembro de 2017.

MDN WEB DOCS. **Primeiros passos**, 2017. Disponível em: < https://developer.mozilla.org/pt-BR/docs/AJAX/Getting_Started>. Acesso em: 21 de setembro de 2017.
MOMENTJS. **Momento.js**, 2017. Disponivel em: <https://momentjs.com/>. Acesso em: 20 de setembro de 2017.

NODEJS. **About Node.js**, 2017. Disponível em: < https://nodejs.org/en/about/>. Acesso em: 9 de setembro de 2017.

NPMJS. **Express fileupload**, 2017. Disponível em: <https://www.npmjs.com/package/express-fileupload>. Acesso em: 19 de setembro de 2017.

ORACLE. **What is MySQL?**, 2017. Disponível em: <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>. Acesso em: 23 de setembro de 2017.

PEREIRA, A., L., G.; GRASSI, R., A. Políticas públicas de combate à criminalidade no Espírito Santo, situação atual e perspectivas de mudanças. **Nexos Econômicos**. Salvador – BA. v 6, n. 2, p.77. 2012

PEREIRA, C. P. **Aplicações web real-time com Node.js**. São Paulo: Casa do Código, 2013. 176 p.

PM2. **Advanced, production process manager for Node.js**, 2017. Disponível em: <http://pm2.keymetrics.io/>. Acesso em: 20 de setembro de 2017.

PREZOTTO, E. D; BONIATI, B. B. **Estudo de Frameworks Multiplataforma Para Desenvolvimento de Aplicações Mobile Híbrida**. Anais do EATI - Encontro Anual de Tecnologia da Informação e Semana Acadêmica de Tecnologia da Informação, Rio grande do sul, n. 1, p. 72-79, nov. 2014.

SECRETARIA DE DESENVOLVIMENTO SOCIAL, G. Segurança. **Prefeitura Municipal de Viana**, 2017. Disponível em: < http://www.viana.es.gov.br/site/publicacao/viana-completa-70-dias-sem-homicidios>. Acesso em 03 dez, 2017.

SEQUELIZE. **Sequelize**, 2017. Disponível em: <https://github.com/sequelize/sequelize>. Acesso em: 20 de setembro de 2017.

TONDIM. **TÔ AQUI: APLICATIVO PARA GEORREFERENCIAMENTO EM AMBIENTES RESTRITOS na FURB**. Trabalho de conclusão de curso, [S.L], 2015.

W3SCHOOLS. **jQuery - AJAX get() and post() Methods**, 2017. Disponível em: < https://www.w3schools.com/jquery/jquery_ajax_get_post.asp>. Acesso em: 20 de setembro de 2017.