

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"

FACULDADE DE CIÊNCIAS - CAMPUS BAURU

DEPARTAMENTO DE COMPUTAÇÃO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ANDRÉ FONSECA LEOPOLDINO ALVES

**BOTÃO DO PÂNICO UNESP BAURU - UM PROTÓTIPO PARA
AS UNIVERSIDADES PÚBLICAS BRASILEIRAS**

BAURU

2017

ANDRÉ FONSECA LEOPOLDINO ALVES

**BOTÃO DO PÂNICO UNESP BAURU - UM PROTÓTIPO PARA
AS UNIVERSIDADES PÚBLICAS BRASILEIRAS**

Trabalho de Conclusão de Curso do Curso
de Ciência da Computação da Universidade
Estadual Paulista “Júlio de Mesquita Filho”,
Faculdade de Ciências, Campus Bauru.
Orientador: Profa. Dra. Simone das Graças
Domingues Prado

André Fonseca Leopoldino Alves

Botão do Pânico UNESP Bauru - Um protótipo para as universidades públicas brasileiras/
André Fonseca Leopoldino Alves. – Bauru, 2017-

38 p. : il. (algumas color.) ; 30 cm.

Orientador: Profa. Dra. Simone das Graças Domingues Prado

Trabalho de Conclusão de Curso – Universidade Estadual Paulista “Júlio de Mesquita Filho”
Ciência da Computação, 2017.

1. Aplicativo Web 2. Aplicativo Mobile 3. Google Firebase 4. GPS 5. TypeScript I. Universidade
Estadual Paulista "Júlio de Mesquita Filho". II. Faculdade de Ciências.

André Fonseca Leopoldino Alves

Botão do Pânico UNESP Bauru - Um protótipo para as universidades públicas brasileiras

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Profa. Dra. Simone das Graças Domingues Prado
Orientador

Profª Adjª Dagmar Ap. Cynthia França Hunger
Convidado

Profa Dra Andréa Carla Gonçalves Vianna
Convidado

Bauru, _____ de _____ de _____.

Dedico este trabalho à meus pais, que
não mediram esforços para que eu
chegasse até aqui.

Agradecimentos

Agradeço aos meus pais e irmão, que sempre estiveram próximo a mim, me deram apoio em todos esses anos e nunca deixaram de acreditar na minha capacidade.

Agradeço aos amigos que fiz durante a graduação, pelos tempos vividos em república, grupos de estudo, etc.

Agradeço a minha orientadora, professora Simone das Graças Domingues Prado, que aceitou me orientar e deu todo o suporte necessário para realização deste trabalho. Agradeço também aos professores que contribuíram com o meu aprendizado e desenvolvimento.

Aos colegas de trabalho por onde passei, do Laboratório de Tecnologia da Informação Aplicada, da Unesp de Bauru, Mstech, GD e GFT, com os quais pude aprender ainda mais sobre tecnologia e como aplicá-la.

*"Eduquem as crianças, para que não seja
necessário punir os adultos".*

(Pitágoras)

Resumo

A violência no Brasil é evidente. As universidades públicas possuem livre acesso e é comum enfrentarem dificuldades para garantir a segurança de quem a frequenta. O processo para implantar medidas para sanear a situação é lento e burocrático. O avanço da tecnologia da informação mudou a forma de interação entre as pessoas e transformou o computador de mesa em um aparelho de mão com alta conectividade e poder de processamento, capaz de obter sua geolocalização através do GPS. Este projeto teve como objetivo desenvolver um aplicativo móvel capaz de acionar a segurança do campus de uma universidade através de um “botão do pânico”. Foram utilizadas tecnologias de última geração como Angular e Google Firebase para implementar o sistema, a fim de buscar máxima performance e eficiência neste protótipo.

Palavras-chave: Violência pública, tecnologia, aplicativo web, aplicativo mobile, Angular, Firebase

Abstract

The violence in Brazil is evident. The public schools have free access and it is common to struggle with security of the people there. The process to implement new measures to sanitize the situation is slow and has a lot of bureaucracies. The advancement of technology changed the way people interact, and transformed the desktop into a hand device high connectivity and processing power, capable of obtaining its geolocation through the GPS. The objective of this project was to develop a mobile application capable to trigger the campus security through a “panic button”. It was used last generations technologies like Angular and Google Firebase to implement the system, in order to obtain maximum performance and efficiency in this prototype.

Keywords: Violence, technology, web application, mobile application, Angular, Firebase

Lista de ilustrações

Figura 1 – Localização da Praça de Esportes.	14
Figura 2 – Popularização dos Smartphones no Brasil	16
Figura 3 – Satélites do GPS em órbita na Terra	18
Figura 4 – Arquitetura do Angular.	21
Figura 5 – Árvore de <i>Componentes</i>	22
Figura 6 – View.	23
Figura 7 – Databinding	23
Figura 8 – Injeção de Dependência	24
Figura 9 – Arquitetura de uma aplicação Cordova	25
Figura 10 – Página de Login do aplicativo Web.	29
Figura 11 – Página principal do aplicativo Web.	30
Figura 12 – Ocorrências em andamento	30
Figura 13 – Histórico de ocorrências - Estado de São Paulo	31
Figura 14 – Histórico de ocorrências de um determinado local	32
Figura 15 – Acesso ao aplicativo Mobile	33
Figura 16 – Página principal do aplicativo Mobile	34
Figura 17 – Botão do Pânico desabilitado	35
Figura 18 – QR Code para download do Aplicativo no Google Play	35

Lista de abreviaturas e siglas

API	Application Programming Interface
DOM	Document Object Model
GPS	Global Positioning System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IBGE	Instituto Brasileiro de Geografia e Estatística
JSON	JavaScript Object Notation
NPM	Node Package Manager
PaaS	Platform as a Service
SDK	Software Development Kit

Sumário

1	INTRODUÇÃO	13
1.1	Problema	13
1.2	Justificativa	14
1.3	Objetivos	15
1.3.1	Objetivo Geral	15
1.3.2	Objetivos Específicos	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Dispositivos móveis	16
2.1.1	Aplicativos Móveis	17
2.1.2	Sistema de Posicionamento Global (GPS)	17
2.2	Aplicações Web	18
2.2.1	Plataforma como serviços	19
3	TECNOLOGIAS	20
3.1	TypeScript	20
3.2	Frameworks	20
3.2.1	Angular	20
3.2.1.1	Módulos	21
3.2.1.2	Componentes e Templates	21
3.2.1.3	Metadata	22
3.2.1.4	Databinding	22
3.2.1.5	Serviços	23
3.2.1.6	Injeção de Dependência	24
3.2.1.7	Angular Google Maps API	24
3.2.2	Ionic	24
3.2.2.1	Apache Cordova	25
3.3	Google Firebase	26
3.3.1	Gerenciamento de usuários	26
3.3.2	Realtime Database	26
3.3.3	Hospedagem	26
3.4	Git	27
4	DESENVOLVIMENTO	28
4.1	Angularfire	28
4.2	Implementação do Aplicativo Web	28

4.2.1	Autenticação	29
4.2.2	Página Principal	29
4.2.3	Ocorrência	30
4.2.4	Histórico de ocorrências	31
4.3	Implementação do Aplicativo Mobile	32
4.3.1	Autenticação	32
4.3.2	Pedido de Socorro	33
4.4	Publicação na Play Store	34
4.5	Publicação na Apple Store	35
5	CONCLUSÃO	36
5.1	Trabalhos Futuros	36
	REFERÊNCIAS	37

1 Introdução

A preocupação cotidiana em relação à questão da violência e da segurança pública no Brasil tem assumido papel de destaque na atualidade. A proximidade da violência e da criminalidade é sentida por todos, fenômeno este que vem atingindo, principalmente, os grandes centros urbanos da nação (FADUL, 2007).

As universidades, como instituições públicas, são ambientes de transmissão e troca de conhecimentos, lugar no qual todos os indivíduos deveriam se sentir seguros. Entretanto, as universidades refletem, reproduzem e não estão imunes aos acontecimentos da sociedade. Têm sido frequentes as ocorrências de furtos, assaltos, homicídios, estupro e tráfico de drogas no ambiente das universidades públicas e privadas de todo o país (PIROLO; MORESCO, 2012).

O uso da tecnologia da informação para aumentar a segurança nas grandes cidades é uma prática comum e feito principalmente por intermédio de câmeras de monitoramento, policiais devidamente treinados equipados com smartphones, rádios e notebooks com acesso à Internet. Infelizmente, essa não é a realidade nas universidades públicas brasileiras, ainda que por motivos de gestão, recursos e prioridade.

O avanço da tecnologia nas últimas décadas permitiu a popularização dos smartphones, dispositivo de mão capaz de conectar à Internet via redes de telecomunicações sem fio apto a processar e trocar informações em tempo real com outros dispositivos com acesso à Internet.

Os aplicativos móveis disponíveis neste mercado infinito de ideias e soluções fazem parte do dia-a-dia do cidadão do século XXI, criando tendências e alimentando a cultura da velocidade, rapidez e instantaneidade, permitindo que uma mensagem seja enviada na escala de milissegundos para o outro lado do planeta. A era da conexão é a era da mobilidade. A internet sem fio, a telefonia celular de última geração traz novas questões em relação ao espaço público e espaço privado (LEMOS, 2004), como a privacidade e segurança.

1.1 Problema

A discussão sobre modelos de segurança adotados em campi universitários ganhou destaque após o assassinato, em maio de 2011, de um aluno no campus da capital da Universidade de São Paulo.

Problemas de segurança são recorrentes em campi universitários e estes necessitam ter um modelo de segurança eficiente. Como implantar um modelo de segurança no ambiente universitário que seja capaz de dar segurança à comunidade universitária e que, ao mesmo tempo, leve em consideração a especificidade de um ambiente acadêmico? (CUBAS; ALVES; CARVALHO, 2013).

A Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP) campus de Bauru ocupa a área de 4.800.000 metros quadrados (200 alqueires), sendo parte deste território de pouca ou zero luminosidade no período noturno, especialmente o trajeto para a Praça de Esportes, que se encontra “fora” do campus e tem acesso através de um trecho que percorre sob a rodovia SP-255 Bauru – Jaú (Figura 1), local perigoso, que possui relatos de assaltos, não recomendado de ser percorrido a noite, dada a ausência de agentes de segurança no local, baixa iluminação e principalmente por ser uma área externa ao campus.

Figura 1 – Localização da Praça de Esportes.



Fonte: Google Maps.

Apesar de tamanha extensão do campus, diversas medidas já foram tomadas para aumentar o grau de segurança na universidade, incluindo a instalação de câmeras de segurança nas portarias 1 e 2, que registram as placas dos veículos que entram e saem das dependências da universidade e câmeras nos estacionamentos. Nos últimos anos, também foi feita a instalação de novos postes de iluminação para ampliar a área iluminada. O campus ainda conta com uma equipe de agentes de segurança, equipados com telecomunicação via rádio e motocicletas.

A UNESP de Bauru possui um menor índice de violência quando comparada com outras universidades como USP (Cidade Universitária) e UNESP de Botucatu, tendo em vista as notícias encontradas numa pesquisa realizada através do motor de busca Google.

1.2 Justificativa

O processo para implantar medidas de segurança nas universidades públicas é lento, burocrático e caro. Um projeto com novos pontos para a instalação de iluminação por exemplo,

pode levar anos para de fato concretizar-se, criando uma abertura para projetos menores, de baixo custo e curto prazo de efetivação.

Uma opção interessante que surge neste cenário são os aplicativos mobile, produzidos principalmente por jovens universitários que buscam conhecimento em tecnologias emergentes, criando soluções inteligentes e inovadoras.

Os aplicativos mobile possuem uma infinidade de aplicações, dada a quantidade de recursos disponíveis neste dispositivo, como *Global Positioning System (GPS)*, leitor de impressão digital, sensor infravermelho, sensor de movimento, monitor cardíaco, câmera, sensor de proximidade, sensor de luz, acelerômetro, entre outros.

A eficiência e a precisão com que os smartphones trabalham com o GPS é satisfatória, e é suficiente para o funcionamento de diversas aplicações usadas no dia-a-dia e de alto desempenho, como Google Maps, Waze e Uber.

1.3 Objetivos

1.3.1 Objetivo Geral

Este trabalho teve por objetivo desenvolver, para os frequentadores da universidade UNESP Bauru, uma aplicação capaz de acionar a equipe de segurança do campus rapidamente em situações emergenciais, visando aumentar a eficiência do atendimento e a segurança como um todo, bem como registrar as ocorrências e gerar relatórios.

1.3.2 Objetivos Específicos

Para concretizar este trabalho, foi necessário o estudo de diversas ferramentas e a codificação de duas aplicações distintas que deverão funcionar em conjunto. Para tanto, foi preciso:

- a) Desenvolver o aplicativo Mobile, ferramenta que irá enviar o pedido de socorro;
- b) Desenvolver o aplicativo Web, ferramenta que irá receber o pedido de socorro;
- c) Elaborar um banco de dados que atenda as aplicações utilizando a plataforma Google Firebase;

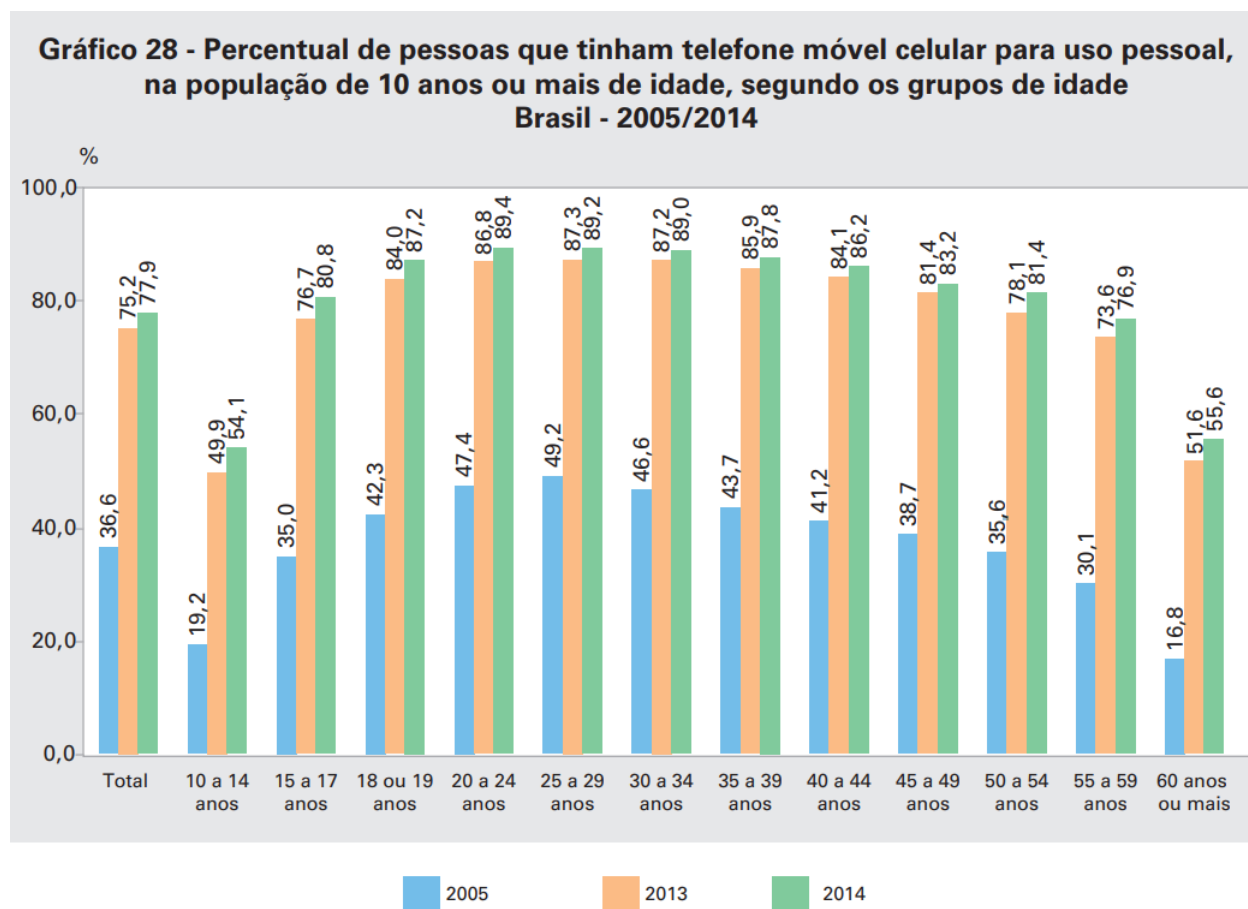
2 Fundamentação Teórica

Para projetar e desenvolver as aplicações proposta neste trabalho, foi necessário compreender como a tecnologia é utilizada atualmente pelas pessoas, bem como as formas mais simples e convenientes de criar soluções práticas, eficientes e impactantes para o público alvo.

2.1 Dispositivos móveis

O uso de *smartphones* está cada vez mais presente no dia a dia do cidadão brasileiro (Figura 2). A popularização desses aparelhos se deve a gama de modernidade presente nos aparelhos, como câmeras, *bluetooth*, *GPS*, etc., e a praticidade do uso, principalmente, com o aumento de opções de pacotes de redes móveis que as operadoras oferecem.

Figura 2 – Popularização dos Smartphones no Brasil



Fonte: : IBGE, Diretoria de Pesquisas, Coordenação de Trabalho e Rendimento, Pesquisa Nacional por Amostra de Domicílios 2005/2014

Em 4 anos, o número de pessoas que usam esse tipo de celular no Brasil, subiu 3,5 vezes, passando de 14% em 2002 para 62% em 2016. O aumento mundial foi de 33,3% em 2012 para 70% no ano de 2016. Em 4 anos o uso de smartphones mais que dobrou em todo o mundo, segundo a pesquisa Google Consumer Barometer ([GOOGLE, 2016b](#)).

Segundo dados divulgados pelo IBGE ([IBGE, 2016](#)), mostrado pela Pesquisa Nacional por Amostra de Domicílios, apontam que o uso dos *smartphones* ultrapassou o uso de computadores, sendo a principal escolha dos brasileiros para se conectar à Internet em 2014.

Com a chegada da terceira geração (3G) os celulares ganharam novas tecnologias e novos conceitos, que visam unir o conforto e a praticidade para satisfazer o usuário ([HERNANDEZ; ALVES; SALVADOR, 2012](#)), bem como ampliar as possibilidades em virtude do avanço da conectividade dos aparelhos.

O uso de *smartphones* tem aumentado a cada dia, são usados como ferramentas de trabalho, lazer e comunicação. O *smartphone* é um telefone celular com funcionalidades avançadas que podem ser estendidas através de programas e aplicações executadas no seu sistema operacional. As pessoas podem desenvolver programas e aplicações que podem funcionar no aparelho, mesmo sem ter vínculo com o fabricante do celular.

2.1.1 Aplicativos Móveis

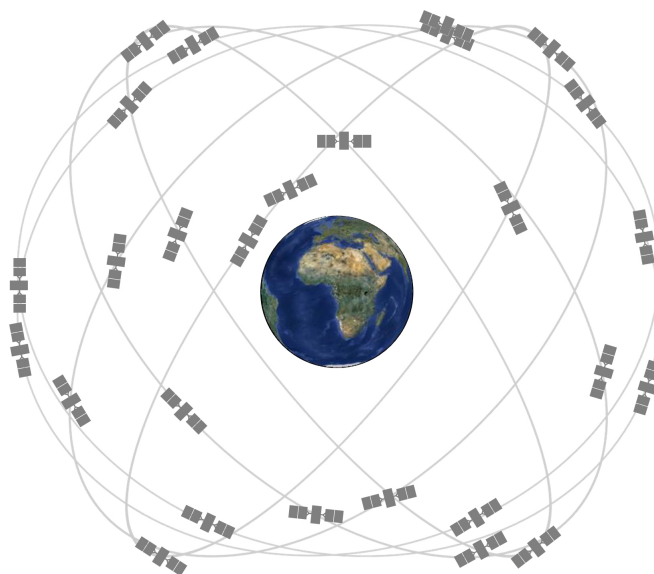
Atualmente o uso de dispositivos móveis está em alta em todo o mundo. Os aplicativos móveis são produtos projetados e desenvolvidos para serem executados especificamente em dispositivos eletrônicos móveis, sendo os mais comuns os *PDA's*, conhecidos também como *Palmtops*, *Tablets*, leitores de *mp3*, telefones celulares e *smartphones* mais modernos e com larga capacidade de armazenamento e processamento ([SILVA; PIRES; NETO, 2015](#)).

O uso de celulares hoje em dia vai além do simples contato com familiares e amigos. Através de um *smartphone* podemos executar diversas tarefas diárias, além também de ter opções de diversão, como aplicativos de jogos, redes sociais ou até mesmo mecanismos de buscas e localização através do *GPS*, que permitiu a popularização de serviços de transporte como *Uber* e aplicativos para obter informações do trânsito, como o *Waze*, que é capaz de calcular rotas otimizadas em tempo real.

2.1.2 Sistema de Posicionamento Global (GPS)

O *GPS* foi projetado e desenvolvido pelo Departamento de Defesa dos Estados Unidos. A intenção inicial era utilizá-lo, principalmente, para a navegação com propósitos militares. Somente com a descoberta da grande precisão do sistema e com o aumento da eficiência dos receptores é que ele passou a ser também utilizado pela comunidade civil. O sistema *GPS* consiste, atualmente, de 24 satélites (Figura 3) distribuídos em 6 órbitas planas ([ORTH; VIEIRA; DEBERT, 2004](#)).

Figura 3 – Satélites do GPS em órbita na Terra



Fonte: <https://www.nasa.gov/>

A constelação atual de satélites *GPS* garante a todo o instante, em qualquer lugar da superfície terrestre, que pelo menos três satélites possam ser observados (ORTH; VIEIRA; DEBERT, 2004). A comunicação com os satélites independe de conexão com a *Internet*, embora quando disponível, seu uso permite uma precisão ainda maior do posicionamento.

As aplicações na modernidade são ilimitadas, tendo como destaques mais recentes a utilização por *drones* (veículo aéreo não tripulado), carros autônomos, aplicativos móveis, voo e navegação, robôs, esportistas e aventureiros.

Este sistema é relevante para este trabalho pois é utilizado para detectar a geolocalização de uma vítima. Ao acionar o botão do pânico, o aplicativo móvel solicita seu posicionamento e recebe um par de coordenadas geográficas, que é então enviado para a aplicação *web* e projetado no mapa da universidade para visualização em tempo real pelos agentes de segurança da universidade.

2.2 Aplicações Web

Na década de 90 com a popularização da Internet houve uma mudança de paradigma quanto à forma de criar e disponibilizar sistemas computacionais, que até então eram instalados individualmente nas máquinas dos usuários através de CDs ou até disquetes. Esse processo era lento e ineficiente, pois a atualização para novas versões ou até correções de bugs necessitavam geralmente de uma reinstalação manual do software.

O avanço da Internet permitiu que sistemas fossem disponibilizados através de um servidor *web*, em oposição a manter uma cópia do sistema instalada em cada computador,

excluindo a necessidade de mídias externas para instalação e inconveniências para atualizá-lo. Além disso, os sistemas *Web* podem ser acessados de qualquer lugar e computador, desde que haja conexão com a Internet.

A literatura oferece diversas definições para esse tipo de sistema, sendo uma delas a de Filho e Padua (2003), que define aplicações *Web* como "produtos de software ou sistemas de informática que utilizam uma arquitetura distribuída, pelo menos parcialmente sob protocolo *HTTP*. Em consequência, pelo menos parte das interfaces com o usuário é acessível através de um navegador (*browser*)".

2.2.1 Plataforma como serviços

O fator de definição que torna *PaaS* (*Platform as a Service*) exclusiva é que permite que desenvolvedores desenvolvam e implementem aplicativos da *Web* em uma infraestrutura hospedada. Ou seja, *PaaS* permite aproveitar os recursos de computação aparentemente infinitos de uma infraestrutura de nuvem (ORLANDO, 2011).

Quando *PaaS* não está sendo usada, configurar um ambiente de desenvolvimento ou de teste geralmente requer as seguintes tarefas (ORLANDO, 2011):

- a) Adquirir e implementar o servidor;
- b) Instalar o sistema operacional, ambientes de tempo de execução, repositório de controle de origem e qualquer outro programa mediador necessário;
- c) Configurar o sistema operacional, os ambientes de tempo de execução, o repositório e um programa mediador adicional;
- d) Mover ou copiar código existente;
- e) Testar e executar o código para assegurar que tudo funcione.

Em contrapartida, ao utilizar uma Plataforma como Serviço, as etapas de adquirir e configurar um ambiente para hospedar e manter uma aplicação *Web* não são necessárias, já que o provedor deste serviço disponibiliza esse recurso através de máquinas virtuais previamente configuradas e prontas para serem usadas. O código é enviado através da rede, do computador do desenvolvedor para a plataforma, e em poucos minutos a aplicação está disponível e funcional.

Dada a facilidade e a conveniência de trabalhar desta forma, foi escolhido utilizar este paradigma para implementar este trabalho. Dentre as alternativas disponíveis, o *Google Firebase* (Seção 3.3) se destacou por oferecer planos gratuitos para aplicações de pequeno porte e por possuir recursos interessantes, como gerenciamento de usuários (Seção 3.3.1) e banco de dados em tempo real (Seção 3.3.2).

3 Tecnologias

Neste capítulo são apresentadas as principais tecnologias utilizadas para o desenvolvimento deste projeto.

3.1 TypeScript

A linguagem *TypeScript* surgiu nos meados de 2012 para auxiliar no desenvolvimento de aplicações web de larga escala, dado que seu “antecessor”, o *JavaScript*, foi criado inicialmente para ser uma linguagem de script para acessar e manipular páginas *HTML*, através do *DOM* (*Document Object Model*) ([SOMASEGAR, 2012](#)).

O fato do *JavaScript* não ter sido designado para ser usado como linguagem de programação não impediu que fosse usada para tal, porém deixa a desejar por não conter diversas características para o desenvolvimento e manutenção de aplicações robustas, como tipagem estática, estruturas de classes, módulos e interfaces. Dada a importância destes fatores, a Microsoft criou o *TypeScript*, que é um *superset* do *JavaScript*, isto é, o próprio *JavaScript*, mas com novos recursos para enriquecê-lo.

O código escrito em *TypeScript* é compilado para código *JavaScript*, garantindo a compatibilidade entre eles e também a interpretação pelos navegadores já existentes, em qualquer plataforma ou sistema operacional.

3.2 Frameworks

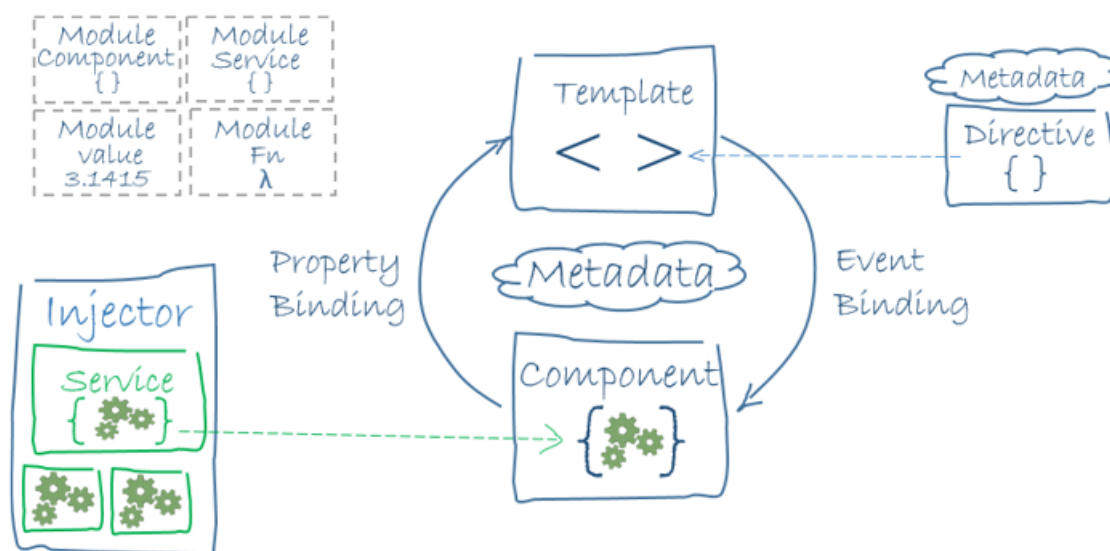
Sabemos que o *HTML* (Linguagem de Marcação de Hipertexto) por si só é uma ótima linguagem para trabalhar com documentos e informações estáticas, porém quando se trata de aplicações dinâmicas, torna-se necessário o uso de tecnologias adicionais para atender as suas necessidades, como bibliotecas e frameworks, capazes de alterar as informações mostradas nos documentos *HTML* ([GOOGLE, 2012](#)). Para o desenvolvimento deste projeto foram utilizados dois *frameworks*.

3.2.1 Angular

Angular é um *framework* de código aberto criado pelo Google para desenvolvimento de aplicações *Web*, que trabalha com *HTML* e *JavaScript*. É possível também usar *TypeScript*, já que este é compilado para código *JavaScript*. O *framework* consiste em criar páginas *HTML* com marcações específicas (*Templates*) e *Componentes* para controlá-las (Seção [3.2.1.2](#)), além de *Serviços* (Seção [3.2.1.5](#)) para cuidar de lógicas e regras de negócios que a aplicação pode

necessitar. O *framework* também oferece diversos *Módulos* (3.2.1.1), *Injeção de Dependência* (Seção 3.2.1.6), ferramentas para *unit test* e outras boas práticas para o desenvolvimento de software (Figura 4) (GOOGLE, 2012).

Figura 4 – Arquitetura do Angular.



Fonte: <https://angular.io/guide/architecture>

Esta tecnologia foi utilizada para desenvolver ambas as aplicações proposta por este trabalho. O objetivo da seguinte subseção é detalhar superficialmente o funcionamento desse *framework*.

3.2.1.1 Módulos

As aplicações *Angular* são modulares, e devem conter pelo menos um módulo, o *root module* (módulo raiz), que é responsável por inicializar a aplicação (GOOGLE, 2016a). Um projeto simples poderia conter apenas este módulo, porém, na prática são criados diversos módulos para separar e organizar o código em “pedaços” menores e específicos, para maior reusabilidade e manutenibilidade. Esta modularidade do *framework* permite também utilizar módulos desenvolvidos por terceiros, o que contribui com a produtividade. Neste projeto os principais módulos utilizados foram o *Angular Google Maps API* (Seção 3.2.1.7) e *Angularfire* (Seção 4.1).

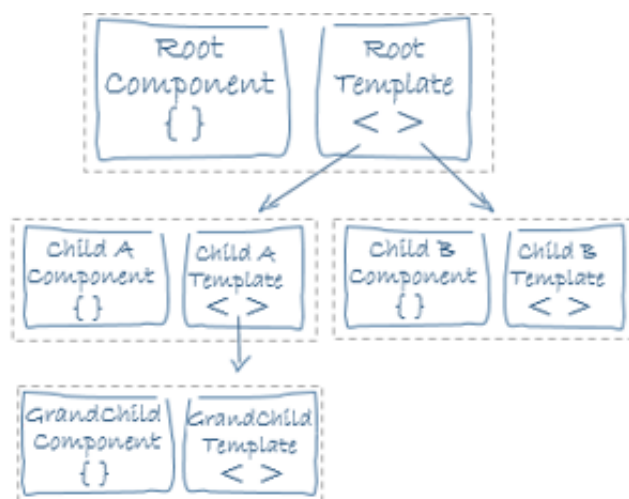
3.2.1.2 Componentes e Templates

Os Componentes são responsáveis por controlar as *Views*, isto é, determinar o comportamento e controlar os dados que são mostrados na tela para o usuário. Esta lógica deve ser definida numa Classe, que irá auxiliar na formação de uma *View*. Esta Classe interage com a *View* através de uma *API* que contém propriedades e métodos. O *framework* cria,

atualiza e destrói *Componentes* conforme o usuário navega pela aplicação (GOOGLE, 2016a). Os *Componentes* são reutilizáveis e podem importar outros *Componentes*, gerando uma árvore de *Componentes* (Figura 5).

Um *Template* é um documento *HTML* que contém informações para dizer ao *framework* como fazer a renderização do seu respectivo *Componente*. Esta relação é estabelecida através do uso de *Metadata* (GOOGLE, 2016a).

Figura 5 – Árvore de *Componentes*.



Fonte: <https://angular.io/guide/architecture>

3.2.1.3 Metadata

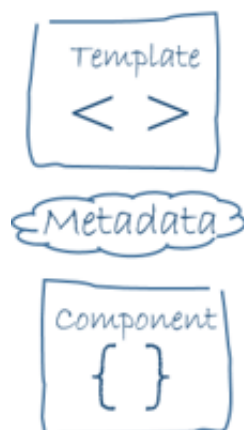
Metadata é um trecho de código que deve ser adicionado no *Componente* para mostrar ao *Angular* como processar esta Classe. É neste momento que o *framework* irá relacionar o *Componente* com seu respectivo *Template*, através da propriedade *templateUrl*, que deve receber o endereço do *Template* relativo ao módulo (GOOGLE, 2016a). Estas três entidades juntas formam uma *View* (Figura 6).

3.2.1.4 Databinding

Sem um *framework*, caberia ao programador a responsabilidade de ler, enviar e atualizar os valores entre o código *JavaScript* e o *HTML*, bem como detectar as ações feitas pelo usuário. O *Angular* possui um mecanismo chamado *Databinding*, que consiste em coordenar os valores que estão no *HTML* (*Template*) e no *JavaScript* (*Componente*), através de marcações específicas no *Template*. Essa conexão entre eles pode ser feita de diversas formas conforme a necessidade (Figura 7).

O *property binding* é utilizado para determinar valores de propriedades de elementos *HTML*, como por exemplo o endereço relativo de uma figura, isto é, elemento `` e a propriedade `src` ligados à uma variável do *Componente* (GOOGLE, 2016a).

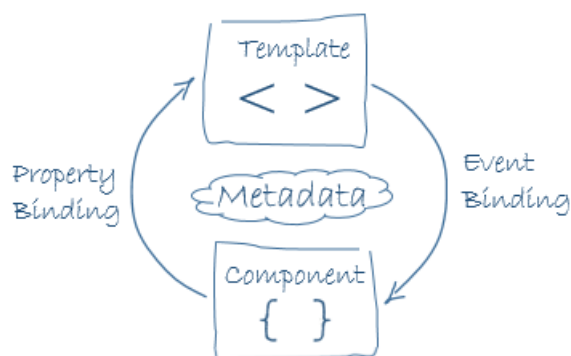
Figura 6 – View.



Fonte: <https://angular.io/guide/architecture>

O *event binding* é aplicado para responder à eventos do *DOM*, que são desencadeados por ações do usuário. Para utilizar este recurso, basta usar uma expressão contendo o nome da função no *Componente* que deve ser chamada após a interação do usuário com este elemento (GOOGLE, 2016a). Exemplo: o botão de pedido de socorro quando acionado pelo usuário através do toque deverá chamar uma função para acionar o socorro (Seção 4.3.2).

Figura 7 – Databinding



Fonte: <https://angular.io/guide/architecture>

3.2.1.5 Serviços

Os *Serviços* englobam uma enorme gama de possibilidades, podendo ser desde um valor, uma função ou até uma funcionalidade que a aplicação necessite. Um *Serviço* é uma *Classe* com propósito bem definido e que deve realizar algo específico. Exemplos: enviar e receber a geolocalização, cálculo de uma taxa, configurações da aplicação, buscar e prover dados (GOOGLE, 2016a).

Na prática, os *Serviços* são utilizados pelos *Componentes* e são fundamentais para qualquer aplicação *Angular*. O papel dos *Componentes* está limitado à controlar o que é

mostrado na tela, portanto, lógica e regras de negócio ou qualquer tarefa não trivial deve ser delegada à um *Serviço*.

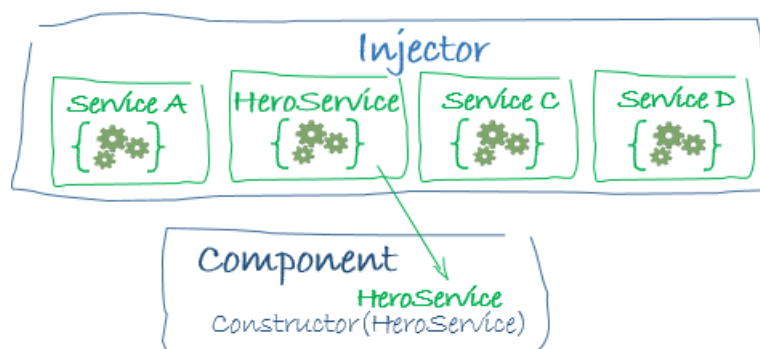
3.2.1.6 Injeção de Dependência

Injeção de Dependência é a maneira de fornecer para uma nova instância de uma *Classe* suas dependências. No *Angular*, este *Design Pattern* é utilizado para injetar *Serviços* nos *Componentes*, os quais são passados como parâmetro no *Construtor da Classe*.

No momento em que o *framework* cria um *Componente*, é feito um pedido para o *Injetor* de quais *Serviços* este *Componente* necessita. O *Injetor* mantém um *Container* com as instâncias dos *Serviços* que foram registrados (Figura 8). É importante lembrar que este registro pode ser feito a nível de *Módulo* (Seção 3.2.1.1) ou *Componente*.

Se durante a execução da aplicação for necessário apenas uma instância de um *Serviço*, este deve ser registrado no *módulo raiz*, enquanto que se registrado em nível de *Componente*, uma nova instância do *Serviço* será criada para cada nova instância do *Componente* (GOOGLE, 2016a).

Figura 8 – Injeção de Dependência



Fonte: <https://angular.io/guide/architecture>

3.2.1.7 Angular Google Maps API

É uma biblioteca *open source* desenvolvida em *JavaScript* e *CoffeScript* para integrar o *Google Maps* com o framework *Angular*. Oferece um conjunto de recursos para trabalhar com mapas em aplicações *Web*, como marcação de pontos dado a latitude e longitude, calcular rotas, distâncias, tempo de percurso, etc.

3.2.2 Ionic

O *Ionic framework* é um *SDK (Software Development Kit)* *open source* que permite desenvolver aplicativos para dispositivos móveis. O foco do *framework* é a aparência e a interação do usuário com a aplicação. O *Ionic* disponibiliza para o desenvolvedor componentes

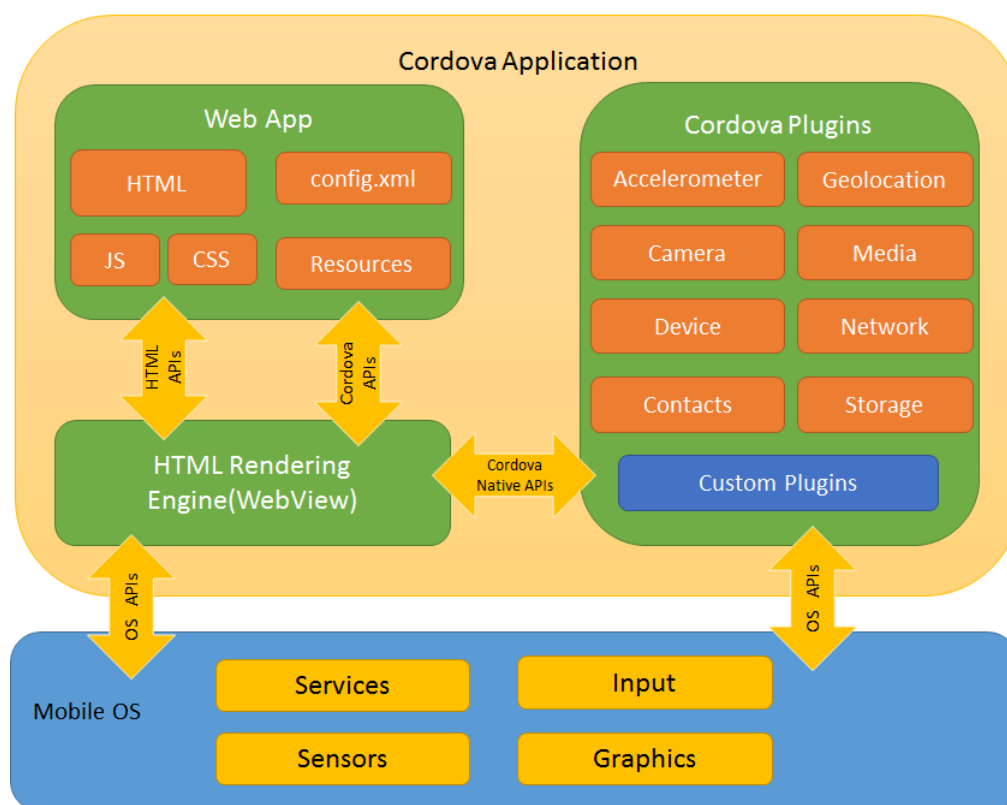
visuais já funcionais, como botões para navegação, menus, alertas, formulários, entre outros. O *Ionic* foi desenvolvido a partir do *Angular* e do *Apache Cordova* (IONIC, 2016).

Esta tecnologia foi utilizada neste trabalho para desenvolver o aplicativo móvel, através do qual os frequentadores da universidade poderão enviar o pedido de socorro.

3.2.2.1 Apache Cordova

O *Apache Cordova* é um *framework open source* para desenvolvimento de aplicativos móveis que permite utilizar tecnologias *web* (*HTML*, *CSS* e *JavaScript*) para criar aplicações multiplataformas, isto é, capazes de funcionar em diferentes sistemas operacionais, como *Android*, *iOS* e *Windows Phone*. O *framework* oferece uma *API* padronizada para acessar os diferentes recursos do dispositivo móvel, como sensores, GPS, contatos, rede, mídias, câmera e armazenamento interno (Figura 9) (APACHE, 2013).

Figura 9 – Arquitetura de uma aplicação Cordova



Fonte: <https://cordova.apache.org/static/img/guide/cordovaapparchitecture.png>

O grande benefício de utilizar essa tecnologia é a conveniência de não precisar desenvolver uma aplicação para cada plataforma, já que cada uma delas possui sua própria tecnologia e linguagem de programação para criar seus respectivos aplicativos.

3.3 Google Firebase

O *Firebase* é uma plataforma para criar aplicativos *iOS*, *Android* e *Web* que fornecem sincronização automática de dados, serviços de autenticação, mensagens, armazenamento de arquivos, análises, entre outros recursos. O *Firebase* é utilizado para criar e desenvolver serviços de *back-end* para aplicativos móveis ([GOOGLE, 2017](#)).

3.3.1 Gerenciamento de usuários

É possível cadastrar e autenticar usuários usando a plataforma *Firebase*, que armazena na nuvem os dados dos usuários cadastrados e oferece diversas formas para realizar a autenticação.

O *Firebase Authentication* fornece serviços de *back-end* e bibliotecas prontas para autenticar usuários em aplicativos *web*. Ele oferece suporte à autenticação por meio de senhas, números de telefone e provedores de identidade terceirizados, como *Facebook* ([FIREBASE, 2017](#)).

3.3.2 Realtime Database

O *Firebase Realtime Database* é um banco de dados hospedado na nuvem. Os dados são armazenados como *JSON* e sincronizados em tempo real com todos os dispositivos conectados. É possível que aplicativos em diferentes plataformas, como *iOS*, *Android* e *JavaScript* compartilhem uma instância do *Realtime Database* e recebam automaticamente atualizações com os dados mais recentes ([FIREBASE, 2017](#)).

Em vez de solicitações *HTTP* típicas, essa tecnologia usa a sincronização de dados, isto é, sempre que os dados são alterados no banco de dados, todos os dispositivos conectados recebem essa atualização em milissegundos ([FIREBASE, 2017](#)). Este recurso foi fundamental para a escolha desta ferramenta, já que é fundamental enviar o pedido de socorro e a localização da vítima para a central de segurança o mais rápido possível.

3.3.3 Hospedagem

O *Firebase hosting* oferece hospedagem rápida e segura para aplicativos *web*. Com este recurso, é possível hospedar aplicações através da criação de um subdomínio no domínio *firebaseapp.com*. Com a *CLI* (Interface de Linha de Comando) do *Firebase*, é possível implantar arquivos de diretórios locais para o servidor que hospeda a aplicação ([FIREBASE, 2017](#)).

Apesar da hospedagem ser gratuita e portanto limitar o espaço disponível para os arquivos da aplicação e o número de usuários simultâneos, a hospedagem fornecida pelo ([FIREBASE, 2017](#)) será suficiente para esta etapa do projeto.

3.4 Git

Este projeto foi desenvolvido utilizando o *GIT* como sistema de controle de versão e o *GitHub* como serviço de hospedagem do código fonte. Tal repositório pode ser acessado livremente através do endereço <https://github.com/aflalves/botao-do-panico-unesp-bauru>.

4 Desenvolvimento

Neste capítulo são apresentados os detalhes de implementação das aplicações, através de explicações de trechos de código fonte, figuras e exemplos.

4.1 Angularfire

O *Angularfire* é a biblioteca oficial do *Angular* para utilizar o *Firebase*. Este projeto de código aberto permite facilmente realizar a conexão entre aplicativos e o banco de dados do *Firebase*, bem como usufruir de suas funcionalidades, como o real-time database, autenticação, etc.

Este pacote é instalado via *NPM*, e a configuração deve ser feita no módulo raiz da aplicação *Angular*, para que fique disponível para todos os outros módulos que forem utilizar este recurso. O algoritmo 1 demonstra como essa configuração foi implementada.

Algoritmo 1 – CONFIGURAÇÃO DO ANGULARFIRE

```
1. export const firebaseConfig = {  
2.     apiKey: "AlzaSyBa9SaixW_S7iccKilq_QNeWkPuF1MI-yYY",  
3.     authDomain: "botao-do-panico-unesp-bauru.firebaseio.com",  
4.     databaseURL: "https://botao-do-panico-unesp-bauru.firebaseio.com",  
5.     projectId: "botao-do-panico-unesp-bauru",  
6.     storageBucket: "botao-do-panico-unesp-bauru.appspot.com",  
7.     messagingSenderId: "618488360157"  
8. }
```

Esta configuração foi realizada em ambas as aplicações, pois é através deste banco de dados que elas se comunicam e compartilham informações. Esta configuração pode ser encontrada no arquivo *app.module.ts* de cada projeto.

4.2 Implementação do Aplicativo Web

Esta seção irá explorar os detalhes da implementação do aplicativo Web, que é responsável por receber o pedido de socorro e ter acesso à localização da vítima em tempo real durante alguma ocorrência.

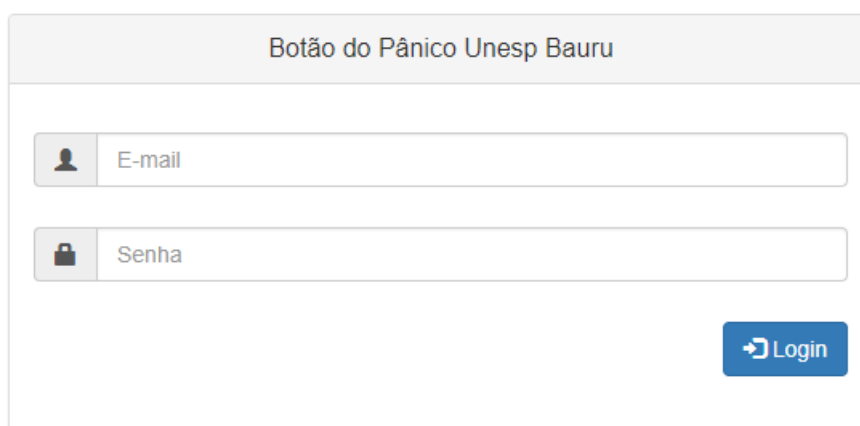
O sistema encontra-se hospedado gratuitamente no serviço de hospedagem fornecido pelo *Google Firebase*, e pode ser acessado em: <https://botao-do-panico-unesp-bauru.firebaseio.com>

4.2.1 Autenticação

Dado o gênero desta aplicação, é esperado que o acesso à esta parte do sistema seja exclusivo dos agentes de vigilância do campus, pois contém dados críticos dos usuários e outras informações que não devem ser acessadas por terceiros.

Para esta fase inicial do projeto, foi implementado uma autenticação simples (Figura 10), na qual existe apenas um usuário disponível para acesso que foi cadastrado manualmente pelo autor no banco de dados *Firebase*.

Figura 10 – Página de Login do aplicativo Web.



Fonte: <https://botao-do-panico-unesp-bauru.firebaseio.com/>

Considerando o momento atual da aplicação, é razoável permitir o livre acesso ao sistema, que pode ser feito através do usuário *seguranca@unesp.com.br* e senha *123456*. O acesso ao site pode ser monitorado e estas credenciais podem ser desabilitadas a qualquer momento pelo autor.

4.2.2 Página Principal

Após a autenticação, o usuário é levado para a tela principal do aplicativo Web (Figura 11). Esta página contém todas as ocorrências já existentes no banco de dados e é capaz de receber em tempo real as novas ocorrências. Para melhor visualização, as mais recentes são as primeiras a serem listadas.

O usuário deve clicar na lupa contida na última coluna da tabela para visualizar a respectiva ocorrência, o que o leva para a tela que contém no mapa a localização da vítima em tempo real (Seção 4.2.3).

Além dessa ação, o usuário pode navegar para a página de histórico (Seção 4.2.4). através do *link* no canto superior direito, bem como sair da aplicação clicando em *Sair*.

Figura 11 – Página principal do aplicativo Web.

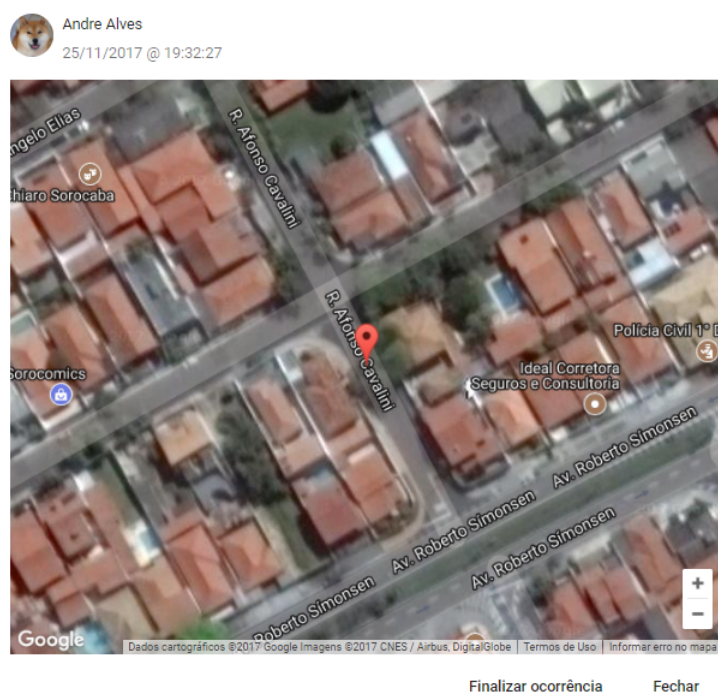
Botão do Pânico - Unesp						Histórico	Sair
Nome	Campus	Data	Hora	Perfil	Status	Detalhes	
Andre Alves	Unesp Bauru	25/11/2017	23:36:13	Visitante	Nova	Q	
Andre Alves	Unesp Bauru	25/11/2017	23:26:6	Visitante	Nova	Q	
Andre Alves	Unesp Bauru	25/11/2017	19:32:27	Visitante	Nova	Q	
Andre Alves	Unesp Bauru	25/11/2017	18:11:34	Visitante	Atendida	Q	
Andre Alves	Unesp Bauru	25/11/2017	18:07:56	Visitante	Atendida	Q	
Andre Alves	Unesp Bauru	25/11/2017	16:07:53	Visitante	Atendida	Q	
Andre Alves	Unesp Bauru	25/11/2017	16:05:17	Visitante	Atendida	Q	
Andre Alves	Unesp Bauru	25/11/2017	15:44:39	Visitante	Atendida	Q	
Andre Alves	Unesp Bauru	19/11/2017	19:44:19	Visitante	Atendida	Q	

Fonte: <https://botao-do-panico-unesp-bauru.firebaseio.com/>

4.2.3 Ocorrência

A página de ocorrência oferece em tempo real a localização da vítima que acionou o socorro (Figura 12). O celular envia a cada dois segundos sua geolocalização, e o ponto projetado no mapa é atualizado automaticamente, permitindo que os agentes de segurança tenham acesso à informação atual e precisa.

Figura 12 – Ocorrências em andamento



Fonte: https://botao-do-panico-unesp-bauru.firebaseio.com

Há duas ações disponíveis nesta tela, que podem ser encontradas no canto inferior direito da janela. Ao selecionar *Finalizar Ocorrência*, o pedido de socorro é dado como encerrado

e é interrompido o compartilhamento de localização entre o celular da vítima e o aplicativo Web.

Inicialmente, a intenção era utilizar a foto vinculada ao *Facebook* do usuário cadastrado, porém por dificuldades técnicas, isso não foi possível. Ainda que a foto seja de grande valor para os agentes, este recurso não compromete o funcionamento e eficiência da aplicação.

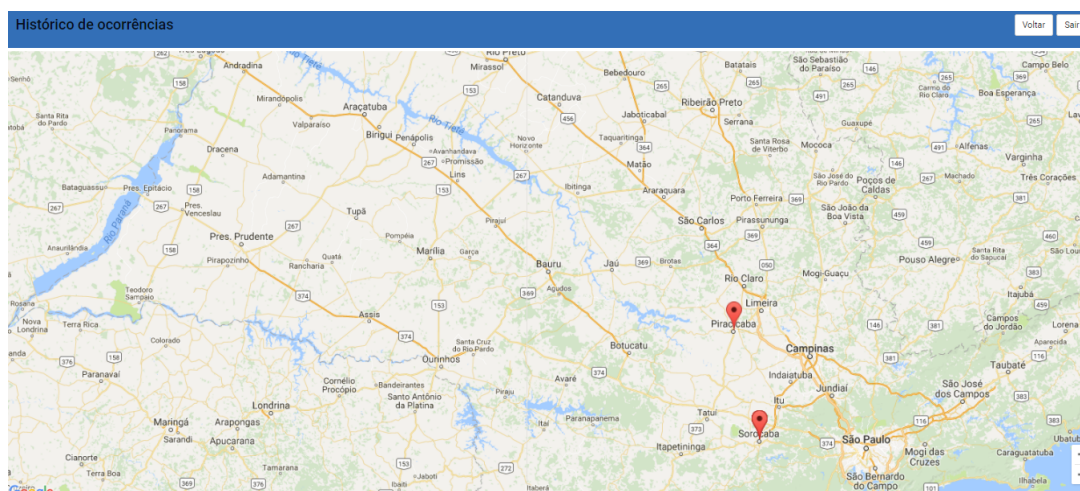
Para inserir o mapa e projetar os pontos, é utilizado a biblioteca *Anguar Google Maps* (Seção 3.2.1.7). O mapa é criado pelo HTML, no arquivo *panic-popup.component.html* através da tag *agm-map*, que recebe como parâmetro o zoom inicial, valores de latitude e longitude para posicionar o mapa e *mapTypeId*, responsável por determinar o estilo do mapa. O tamanho do mapa é configurado através de CSS e é configurado no arquivo *panic-popup.component.css*.

O ponto é marcado através da tag *agm-marker*, que recebe como parâmetro a latitude e longitude do ponto a ser marcado no mapa. Os valores destes atributos são lidos do banco de dados e quando sofrem alteração (vítima se deslocando, por exemplo), o aplicativo automaticamente atualiza as coordenadas com os valores atuais e o posicionamento do ponto é atualizado no mapa.

4.2.4 Histórico de ocorrências

A página de histórico oferece uma forma simples e visual de analisar as ocorrências registradas no aplicativo. Em suma, são projetados no mapa todos os pontos de todas as ocorrências que estão no banco de dados (Figura 13).

Figura 13 – Histórico de ocorrências - Estado de São Paulo



Fonte: <https://botao-do-panico-unesp-bauru.firebaseio.com/relatorios>

O mapa permite aproximar a imagem para obter uma visualização mais acurada de uma determinada região (Figura 14). Para inserir o mapa e projetar os pontos, é utilizado a biblioteca *Anguar Google Maps* (Seção 3.2.1.7). Semelhante como a forma que um ponto é

Esta versão oferece acesso livre ao sistema, podendo ser acessado através do *Facebook* (isto é, não é necessário realizar cadastro, as informações para ingresso no aplicativo são importadas da conta do *Facebook* do usuário) ou caso não queira se identificar, poderá acessar no modo anônimo (Figura 15).

Em ambas as formas de acesso, as informações do usuário não são salvas no banco de dados, e sim gravados nas ocorrências quando acontecem.

Figura 15 – Acesso ao aplicativo Mobile



Fonte: Screenshot do Aplicativo Mobile (produzido pelo autor)

O acesso por *Facebook* é realizado através do plugin *Ionic Cordova* chamado *cordova-plugin-facebook4*, que é injetado no aplicativo Web no arquivo *login.ts*. Com poucas linhas de código, é possível concretizar o acesso com as informações da conta do *Facebook*, caso o aplicativo do *Facebook* esteja instalado no aparelho e funcionando corretamente.

4.3.2 Pedido de Socorro

Tendo finalizado a fase de ingresso, o usuário é redirecionado para a página principal do aplicativo, que contém o botão do pânico e outras opções, como teste de conexão e telefonar para vigiância (Figura 16). Ao chegar nesta tela, o sistema irá automaticamente tentar obter a geolocalização do dispositivo e verificar se há conectividade com o banco de dados, para determinar se o aplicativo está pronto para uso. Alguns segundos depois, um alerta será exibido quando este teste finalizar, informando ao usuário o resultado deste teste.

Caso o teste tenha falhado, o botão do pânico ficará desabilitado e o campo *Status* exibirá a mensagem de erro (Figura 17). Neste caso, o usuário deve conferir se o GPS do

Figura 16 – Página principal do aplicativo Mobile



Fonte: Screenshot do Aplicativo Mobile (produzido pelo autor)

dispositivo está ativado e se há conexão com a Internet. O teste pode ser novamente através do botão *Testar Socorro*.

A tela principal do aplicativo ainda contém no canto inferior direito o número e o botão para discar automaticamente para a vigilância do campus, facilitando a comunicação em qualquer outra situação que venha acontecer.

4.4 Publicação na Play Store

O aplicativo Mobile foi inserido na loja de aplicativos do *Google (Play Store)* para ser disponibilizado aos frequentadores da Unesp que tiverem interesse em testar o aplicativo em seu estado atual e também para oferecer uma interação mais dinâmica ao público no dia da amostra de TCCs do curso de Ciência da Computação, programada para acontecer em 18 de Dezembro de 2017.

Para instalar o aplicativo, é possível encontrá-lo na *Play Store* buscando por "Botão Unesp" ou diretamente pelo endereço <https://play.google.com/store/apps/details?id=io.ionic.bdpub>.

Uma terceira forma de instalar o aplicativo é através do QR Code gerado pelo autor (Figura 18), que será impresso e disponibilizado ao público no dia da Amostra de TCCS.

Figura 17 – Botão do Pânico desabilitado



Fonte: Screenshot do Aplicativo Mobile (produzido pelo autor)

Figura 18 – QR Code para download do Aplicativo no Google Play



Fonte: Gerado pelo autor através do site: <http://goqr.me/>

4.5 Publicação na Apple Store

Infelizmente não foi possível publicar o aplicativo na loja da *Apple*, pois apesar do aplicativo móvel ter sido desenvolvido usando uma tecnologia multiplataforma (Seção 3.2.2), é necessário um computador com sistema operacional *macOS* para transformar o código fonte do projeto em arquivo binário compatível com o sistema operacional dos telefones celulares da *Apple* para publicação na loja de aplicativos.

5 Conclusão

O desenvolvimento deste trabalho teve como objetivo implementar uma solução de baixo custo para promover maior segurança e eficiência no atendimento de eventuais problemas dentro da universidade, aplicando tecnologias do estado-da-arte da área da computação e desenvolvimento de software.

Apesar da implementação ter sido concretizada conforme planejado e as funcionalidades fundamentais estarem praticáveis, este foi o primeiro passo de uma longa jornada para a universidade adotar de fato a solução proposta, já que o trabalho apresentado possui simplificações na forma de acesso ao sistema e quantidade de dados em sua totalidade.

5.1 Trabalhos Futuros

O aplicativo deve passar por um período de testes nos próximos meses e coleta de feedback dos usuários para que possam ser desenvolvidas novas funcionalidades ou adaptações.

Uma nova fase de desenvolvimento deve surgir para atender estas mudanças e também o envolvimento de novas pessoas com o projeto, como desenvolvedor e designers do Laboratório de Tecnologia Aplica (LTIA) ou a Empresa Júnior de Computação da Unesp de Bauru, juntamente com as entidades cabíveis da universidade para prosseguir com a melhoria e implantação da solução.

O autor fica à disposição para dar continuidade ao projeto e contribuir nas próximas etapas que vierem a surgir.

Referências

- APACHE, S. F. Apache cordova overview. 2013. Disponível em: <<https://cordova.apache.org/docs/en/latest/guide/overview/index.html>>. Acesso em 11 Novembro 2017.
- CUBAS, V. O.; ALVES, R.; CARVALHO, D. Segurança no campus: um breve levantamento sobre as políticas de segurança na usp e em universidades estrangeiras. *Revista Brasileira de Segurança Pública*, v. 7, n. 1, p. 182–198, 2013. Disponível em: <<http://revista.forumseguranca.org.br/index.php/rbsp/article/viewFile/274/140>>. Acesso em 16/04/2017.
- FADUL Élvia. A universidade, a comunidade e a segurança pública: uma abordagem em três dimensões. *Revista Eletrônica Observare*, Universidade Salvador, v. 1, p. 30–88, 2007. Disponível em: <<http://www.observatorioseguranca.org/revista/vol1/pdfrevista/TextoElviaFadul.pdf>>. Acesso em: 15/04/2017.
- FILHO, P.; PADUA, W. de. Engenharia de software. Rio de Janeiro: LTC, 2003.
- FIREBASE. Documentação oficial do firebase. 2017. Disponível em: <<https://firebase.google.com/docs/>>. Acesso em 12 Novembro 2017.
- GOOGLE. What is angularjs? 2012. Disponível em: <<https://docs.angularjs.org/guide/introduction>>. Acesso em 03 Novembro 2017.
- GOOGLE. Angular architecture overview. 2016. Disponível em: <<https://angular.io/guide/architecture>>. Acesso em 05 Novembro 2017.
- GOOGLE. Google consumer barometer: The internet in numbers 2012 - 2016. 2016. Disponível em: <<https://pt.scribd.com/document/340694204/Raport-Consumer-Barometer-2012-2016>>. Acesso em 15 Novembro 2017.
- GOOGLE, C. P. Serviços de back-end de aplicativos para dispositivos móveis. 2017. Disponível em: <<https://cloud.google.com/solutions/mobile/mobile-app-backend-services>>. Acesso em 12 Novembro 2017.
- HERNANDEZ, D. M.; ALVES, F.; SALVADOR, G. S. A popularização dos smartphones e tablets. Centro Estadual de Educação Tecnológica Paula Souza, 2012. Disponível em: <http://www.etechortolandia.com.br/novo/files/ptcc_smartphones.pdf>. Acesso em 15 Novembro 2017.
- IBGE. Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal : 2014. *II Workshop de Iniciação Científica em Sistemas de Informação, Goiânia - GO*, IBGE, 2016. Disponível em: <<https://biblioteca.ibge.gov.br/visualizacao/livros/liv95753.pdf>>. Acesso em 15 Novembro 2017.
- IONIC. Ionic core concepts. 2016. Disponível em: <<https://ionicframework.com/docs/intro/concepts/>>. Acesso em 11 Novembro 2017.

LEMONS, A. Cibercultura e mobilidade: a era da conexão. *Revista Razón y Palabra*, n. 41, 2004. Disponível em: <<http://www.facom.ufba.br/ciberpesquisa/andrelemons/cibermob.pdf>> Acesso em 15/04/2017.

ORLANDO, D. Modelos de serviços de computação em nuvem, parte 2: Plataforma como serviço. 2011. Disponível em: <<https://www.ibm.com/developerworks/br/cloud/library/cl-cloudservices2paas/>>. Acesso em 12 Novembro 2017.

ORTH, D. M.; VIEIRA, S. J.; DEBERT, E. Gps – global positioning system – ferramenta utilizada para mapear unidades de conservação. 2004. Disponível em: <[http://www.fcav.unesp.br/Home/departamentos/engenhariarural/TERESACRISTINA TARLEPISSARRA/Artigo/GPS145.pdf](http://www.fcav.unesp.br/Home/departamentos/engenhariarural/TERESACRISTINA%20TARLEPISSARRA/Artigo/GPS145.pdf)>. Acesso em 13 Novembro 2017.

PIROLO, M. A. M.; MORESCO, M. C. Segurança na universidade: opinião da comunidade acadêmica. *Revista Nupem*, Campo Mourão, v. 4, n. 7, p. 251–259, 2012. Disponível em: <<http://www.fecilcam.br/revista/index.php/nupem/article/view/226>>. Acesso em: 15/04/2017.

SILVA, L. L. B. da; PIRES, D. F.; NETO, S. C. Desenvolvimento de aplicações para dispositivos móveis: Tipos e exemplo de aplicação na plataforma ios. *II Workshop de Iniciação Científica em Sistemas de Informação, Goiânia - GO*, 2015. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wicisi/2015/004.pdf>>. Acesso em 15 Novembro 2017.

SOMASEGAR, S. Typescript: Javascript development at application scale. *Somasegar's blog*, 2012. Disponível em: <<https://blogs.msdn.microsoft.com/somasegar/2012/10/01/typescript-javascript-development-at-application-scale>>. Acesso em 03 Novembro 2017.