



---

**Faculdade Nova Roma**  
**Curso de Bacharelado em Ciências da Computação**

---

## **Security Location**

# **DOCUMENTO DE ARQUITETURA**

Igor Leonardo Nogueira de Lima

Recife, 13 de Dezembro de 2019

## Sumário

1.	Introdução	3
2.	Tecnologias Utilizadas	4
2.1	Front-end	4
2.1.1	Angular	4
2.2	Back-end	4
2.2.1	Google Maps API	4
2.2.2	Firebase	4
2.2.3	NodeJs	5
2.3	IDE- Integral Development Environment	5
2.3.1	Visual Studio Code	5
2.4	Sistema de Controle de Versão	5
2.4.1	GIT	5
2.5	Servidor	6
2.5.1	Heroku	6
2.6	Banco de Dados	6
2.6.1	Firebase Database	6
2.7	Ferramentas Case	7
2.7.1	UML – Unified Modeling Language	7
3.	Distribuição do Sistema	7
3.1	Descrição	7
3.1.1	Cliente	7
3.1.2	Servidor de Aplicação	7
3.1.3	Servidor de Banco de Dados	8

# Documento de Arquitetura

## 1. Introdução

Esta aplicação foi desenvolvida após perceber a necessidade de verificar de acordo com região o nível de perigo, assim evitando possíveis ocorrências de virem a acontecer. Muito útil também para turistas que não tem conhecimento algum sobre a região e para transportadoras e motoristas de aplicativo(Uber, 99 POP).

Este documento tem como objetivo informar as decisões arquiteturais significativas nas tomadas para o desenvolvimento da Aplicação Security Location, que possui como funcionalidades o cadastro de ocorrências, a verificação por bairro dos locais onde houveram ocorrências cadastradas e um gráfico que verifica a quantidade de ocorrências por bairro.

De modo geral, serão apresentadas as ferramentas utilizadas passando do Front-end até o Back-end, as IDEs, o banco de dados e ferramentas de CASE que forma utilizadas para o desenvolvimento, análise, modelagem da aplicação e criação de artefatos e diagramas.

A distribuição do sistema também será apresentada para deixar claro como as partes do sistema se distribuem e comunicam.

## **2. Tecnologias Utilizadas**

### **2.1 Front-end**

#### **2.1.1 Angular**

Para o desenvolvimento da aplicação foi escolhido o framework Angular, que utiliza tecnologias primárias para desenvolvimento web (HTML, CSS e JavaScript) de fácil aprendizado, possibilitando uma fácil implementação da construção do Front-end. Framework bastante utilizado por ter fácil interação com o desenvolvedor, e ter fácil integração com outras ferramentas de desenvolvimento. O Angular conta com o typescript, que basicamente adiciona uma tipagem estática ao javascript, que por padrão é uma linguagem de tipo dinâmico, ou seja, as variáveis e funções podem assumir tipos distintos durante o tempo de execução.

### **2.2 Back-end**

#### **2.2.1 Google Maps API**

O google maps API foi utilizado para fazer toda parte de geolocalização no que se refere identificar a localização do usuário, e informar onde houveram ocorrências registradas por outros usuários e informar quais áreas tem mais risco, de acordo com o número de ocorrências registradas.

#### **2.2.2 Firebase**

A utilização do firebase se deu, pelo fato dele ser um banco de dados alojado na nuvem, dando mais segurança para a transferência dos dados e armazenamento de informações dos usuários, para garantir isso o firebase só permite inserção ou leitura após os usuários estarem autenticados no aplicativo.

O firebase é uma plataforma de desenvolvimento web e mobile que armazena dados em tempo real, por isso é conhecido por ser um banco realtime, ou seja, sincroniza de forma automática todos os dados fornecidos a ele e por ele em tempo real, o usuário só precisa estar conectado à internet para utilizar o mesmo.

### **2.2.3 NodeJs**

Foi utilizado para obter o conteúdo em nível de Front-end (cliente) e enviar para o servidor fazendo uma comunicação entre cliente e servidor.

O Node.js é uma plataforma construída para desenvolvimento de aplicações web ou mobile escaláveis de alta performance usando JavaScript e o V8 JavaScript Engine, sendo assim podem ser criadas variedades de aplicações web utilizando apenas JavaScript. O V8 JavaScript Engine é um interpretador de JavaScript open source implementado pelo Google, usando C++ utilizado pelo navegador Google Chrome.

## **2.3 IDE- Integral Development Environment**

### **2.3.1 Visual Studio Code**

IDE para desenvolvimento criada pela Microsoft destinada ao desenvolvimento de aplicações web, possui suporte à várias linguagens, como JavaScript, TypeScript, CSS, HTML e etc... A utilização desta ferramenta se dá por ser de fácil aprendizagem e possuir vários plug-ins, isso auxiliou o desenvolvimento da aplicação.

## **2.4 Sistema de Controle de Versão**

### **2.4.1 GIT**

É um sistema de controle de versão distribuído e um sistema de gerenciamento de código fonte. Para armazenamento do versionamento foi utilizado o GitHub. Também foi feita a integração com o Heroku para o Deploy automático após alterações.

## **2.5 Servidor**

### **2.5.1 Heroku**

Servidor que recebe a requisição e hospeda a aplicação em uma url, que pode ser aberta através de navegadores, como IE(Internet Explorer), Google Chrome... etc. Para fazer a hospedagem é necessário a instalação das dependências do heroku e integração do projeto feito de acordo com a tecnologia usada, nesse caso com Angular e Node.JS. Para facilitar o processo de integração e deploy, o heroku possui uma ferramenta que faz integração com o GitHub, facilitando a configuração de deploy inicial e automático. Basta enviar as alterações para o repositório do GitHub que, caso a opção de deploy automático esteja ativa, a aplicação irá receber as alterações automaticamente.

## **2.6 Banco de Dados**

### **2.6.1 Firebase Realtime Database**

O Firebase Realtime Database é um banco em nuvem noSQL que sincroniza os dados em tempo real(Realtime Database), os dados são armazenados como JSON, o serviço fornece aos desenvolvedores uma API permitindo que os dados do aplicativo sejam sincronizados entre os clientes, também funciona off-line, pois o SDK do Firebase Realtime Database mantém os dados em disco. Quando a conectividade é reestabelecida, o dispositivo cliente recebe as alterações perdidas e faz a sincronização com o estado atual do servidor. Sendo esses os principais motivos pela utilização do Firebase.

## **2.7 Ferramentas Case**

### **2.7.1 UML – Unified Modeling Language**

Uml é uma linguagem usada para modelar e planejar um sistema através de desenhos e de uma série de artefatos que auxilia na tarefa de análise, modelagem e documentação de projetos, nela se encontra o comportamento do sistema que é apresentado ao cliente, evitando um mal entendido tanto do cliente quanto do analista ou programador.

- Astah Community

Ferramenta usada inicialmente para modelar diagramas de classe em UML. Optou-se por essa ferramenta por ser profissional e de fácil usabilidade, também por ter sido utilizada em trabalhos acadêmicos na disciplina de Análise e Projeto Orientado à Objetos.

## **3. Distribuição do Sistema**

### **3.1 Descrição**

#### **3.1.1 Cliente**

O cliente utiliza a aplicação através de uma estação desktop ou de um aparelho mobile (através do navegador) onde se comunica com o servidor na nuvem e sincroniza os dados.

#### **3.1.2 Servidor de Aplicação**

É onde a aplicação é hospedada e iniciada, utilizando a url para se comunicar através do navegador web, utilizando os componentes e serviços da aplicação.

### 3.1.3 Servidor de Banco de Dados

Para esta aplicação, foi utilizado o Firebase Database, para persistir os dados no sistema. Tem comunicação com a aplicação utilizando algumas dependências do Firebase e Node Js.