

UNIVERSIDADE DE SÃO PAULO  
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO  
BACHAREL EM CIÊNCIAS DE COMPUTAÇÃO

**PROGRAMAÇÃO ORIENTADA A OBJETOS: Sistema de denúncia de casos de  
feminicídio, assédio e violência doméstica**

ALBERTO CAMPOS NEVES  
GABRIEL VAN LOON BODÊ DA COSTA DOURADO FUENTES ROJAS  
GIOVANI DECICO LUCAFÓ  
TAMIRIS FERNANDES TINELLI

SÃO CARLOS - SP

## 1. INTRODUÇÃO

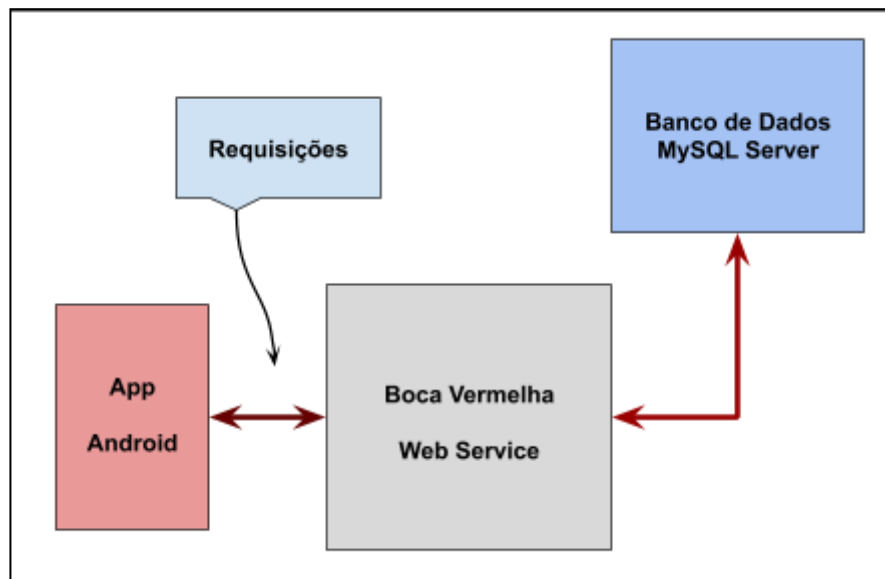
Neste projeto foi implementado um sistema android de denúncia de casos de violência contra a mulher denominado pelo grupo de **Boca Vermelha**. Seu funcionamento está dividido entre dois projetos distintos com diferentes funcionalidades (Android e Java Server) que serão explicados detalhadamente nos tópicos futuros.

## 2. ESTRUTURA DO PROJETO

A totalidade do projeto está distribuída entre as seguintes estruturas:

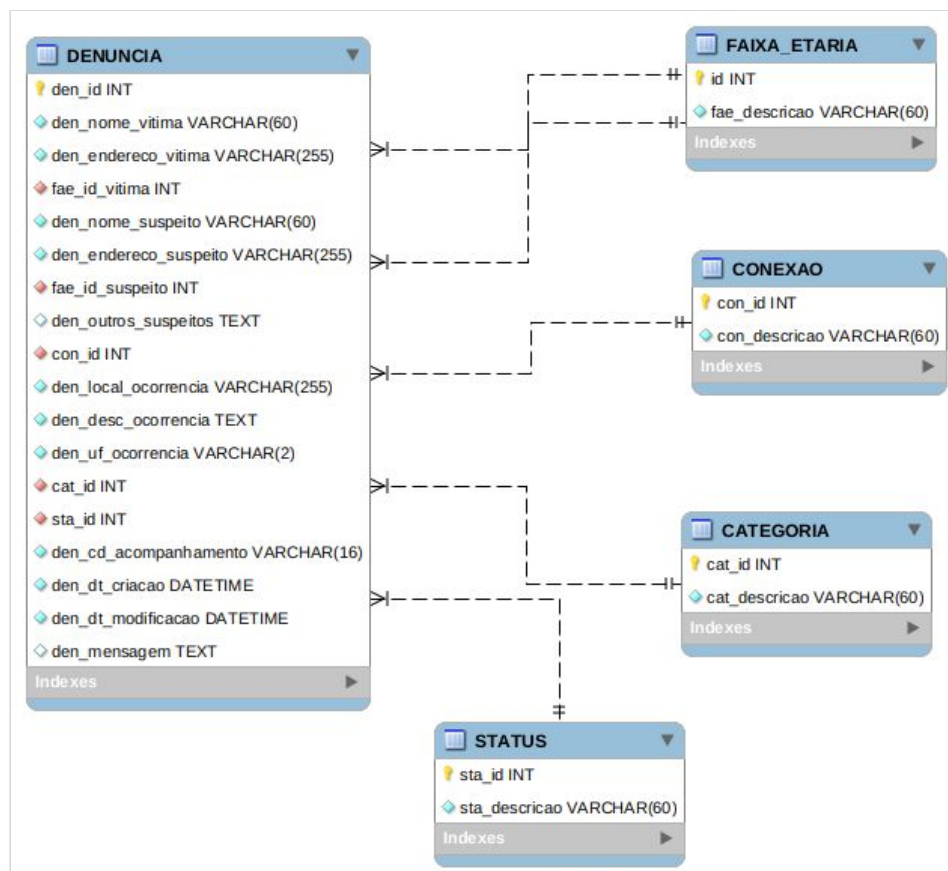
- **Banco de dados:** responsável pelo armazenamento de todos os dados gerados pelas aplicações.
  - **Tecnologias utilizadas:** Oracle MySQL Server para executar o servidor e Oracle MySQL Workbench para gerar os Modelos de Entidades e Relacionamentos (MER).
- **Web Service:** responsável por fazer a ponte entre as aplicações client-side com o banco de dados. É nessa aplicação que está toda a regra de negócio do sistema.
  - **Tecnologia utilizada:** Java EE para aplicações web utilizando Spring Boot apenas para realizar o mapeamento e reflection de requisições. Para realizar as conexões com o banco de dados foi utilizado o driver JDBC.
  - **IDE utilizada:** Eclipse IDE for Enterprise Java Developers versão 4.11.0.
  - **Servidor utilizado:** Tomcat v 9.0 Server no ambiente Ubuntu 18.04 LTS.
- **Android:** aplicação client-side que permite ao usuário enviar denúncias ao sistema, acompanhar as denúncias enviadas e também possui algumas outras funcionalidades que serão abordadas adiante.
  - **Tecnologia utilizada:** linguagem Kotlin para o desenvolvimento da aplicação. Retrofit v2.6.0 para o disparo de requisições ao Webservice e, por fim, Any Chart v.1.1.2 free Trial para a geração de gráficos.
  - **IDE utilizada:** Android Studio v 3.4.1
  - **Dispositivos utilizados:** Pixel 2 com API 24 (Emulado) e também Moto G(6) (Device) com Android 9.0

Veja a esquemática do projeto a seguir:



### 3. BANCO DE DADOS - ESTRUTURA E COMO EXECUTAR

Para o correto funcionamento do aplicativo, foi modelado um banco de dados que permitisse ao sistema salvar e recuperar as denúncias de maneira eficiente. Veja a seguir o Modelo de Entidade e Relacionamento (MER) utilizado:



Além de modelar os dados, também foi necessário gerar um lote de dados de teste pois uma das funcionalidades do aplicativo android necessitava de informações recebidas pelo Webservice para gerar dados acerca das denúncias.

Para resolver esse problema, o banco de dados foi populado com cerca de 1300 registros de teste. Porém, como a ideia era simular a realidade da violência desses dados no contexto nacional atual as categorias de violência foram distribuídas baseadas em dados liberados por órgãos responsáveis como os liberados no **Relatório Geral da Central de Atendimento à Mulher - 2017**, disponível no site do respectivo ministério (<https://www.mdh.gov.br>).

Junto dos arquivos do projeto, portanto, consta o script para criação do banco de dados do projeto assim como o script para a inserção dos dados de teste.

#### 4. WEB SERVICE - ESTRUTURA E COMO EXECUTAR

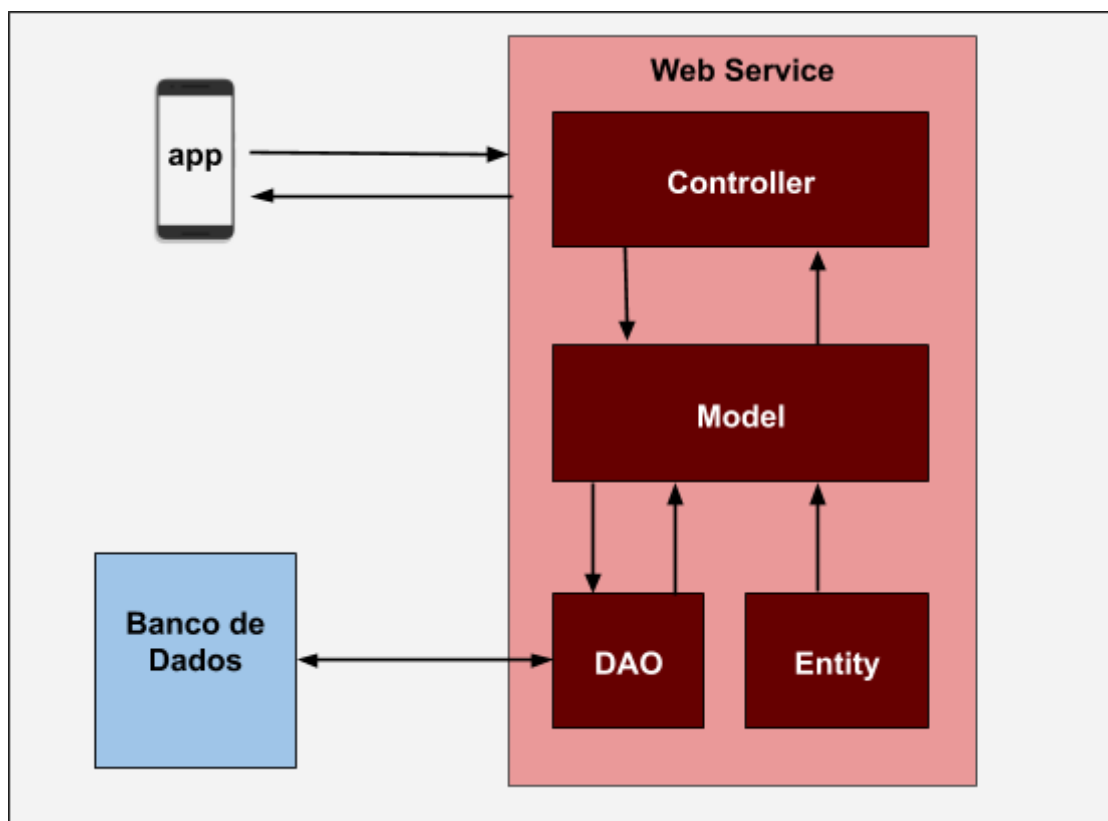
Conforme foi especificado no documento de proposta de projeto, o Web Service do Boca Vermelha atua como executor das regras de negócio do projeto como um todo, fazendo as transações entre as aplicações client-side com as server-side.

Como foi desenvolvido apenas a aplicação mobile, apenas as requisições que ela necessita foram implementadas, sendo elas:

- **url/bocaVermelha/denuncia/cadastrar:** a requisição recebe via POST os dados de uma nova denúncia para ser cadastrada no sistema. Caso a inserção seja feita com sucesso o web service retorna o **Código de Acompanhamento** e uma mensagem de sucesso. Caso contrário retorna um BAD REQUEST junto com a causa do erro.
- **url/bocaVermelha/denuncia/acompanhamento:** a requisição recebe via POST um código de acompanhamento e retorna, caso encontre, os dados não sensíveis referentes ao status atual da denúncia (nomes e endereços não são retornados). Caso contrário retorna um BAD REQUEST junto com a causa do erro.
- **url/bocaVermelha/denuncia/estatisticas:** a requisição retorna ao usuário uma série de dados que podem ser utilizados para gerar diferentes gráficos. Caso haja algum erro na requisição retorna um BAD REQUEST junto com a causa do erro.

A estrutura do Web Service é segmentada de maneira hierárquica, sendo cada uma das camadas responsável pela abstração de alguma parte do funcionamento do projeto.

- **Controller:** camada responsável pelo mapeamento das url's. Toda vez que uma requisição é recebida pelo servidor o Spring Boot decodifica a url e executa a classe e o método responsável pela requisição. O Controller também é responsável por definir quais dados serão recebidos e enviados por determinada url.
- **Model:** camada responsável pelas regras de negócio do sistema. Ela verifica se os dados estão consistentes por meio de validações e realiza as chamadas de camadas inferiores. Além disso, o model é responsável pela manipulação das Entities.
- **Entities:** conjunto de classes que implementam os atributos ou objetos que serão manipulados pelo sistema. Essas classes não atuam na regra do negócio, mas são importante para modelar os dados manipulados pelo sistema.
- **Data Access Objects (DAO):** camada responsável pelo acesso ao banco de dados. nela está implementada as queries que se comunicam com o banco e retornam, quando necessário, os dados para uma camada superior.



Por fim, para a execução deste projeto é necessário fazer algumas configurações:

1. **Configuração do Tomcat:** é necessário ter alguma versão do Tomcat instalada na máquina.
2. **Importar o projeto ao workspace do Eclipse.**
3. **Adicionar o Tomcat à IDE:** a IDE Eclipse especificada anteriormente é ideal para executar este projeto, pois permite que o usuário selecione o servidor Tomcat disponível na máquina para rodar a aplicação. Para isso basta ir em Project > Properties > Server e configurar o servidor local.
4. **Configurar o banco de dados:** para configurar os dados de acesso ao banco de dados é necessário configurar os dados da classe do package *br.com.bocaVermelha.dao.Conexao* com os nome do banco de dados e os dados de acesso do usuário.
5. **Executando o servidor:** Após todas as configurações, basta executar ao adicionar o Web Service ao servidor. Para isso basta ir em Window > Show View > Server, na aba de servidores selecionar o Tomcat instalado e, com o lado direito selecionar Add and Remove. Por fim, após adicionar basta executar o projeto em Run > Run as > Run on server.

É importante verificar no Console se o servidor executa o Spring Boot na hora da inicialização, pois caso contrário o Web Service não executará corretamente.

## 5. ANDROID - ESTRUTURA E COMO EXECUTAR

O aplicativo Android atua como a parte client-side voltada para o usuário final de todo o sistema, pois é por meio dela que as denúncias serão de fato disparadas para serem cadastradas no sistema.

Seguindo o que foi especificado no documento de proposta, o aplicativo foi pensado para possuir uma interface simples e intuitiva para que qualquer pessoa possa conseguir cadastrar uma denúncia. O aplicativo, por prezar pelo anonimato, não pede que o usuário se cadastre e em nenhum momento salva os dados pessoais do mesmo.

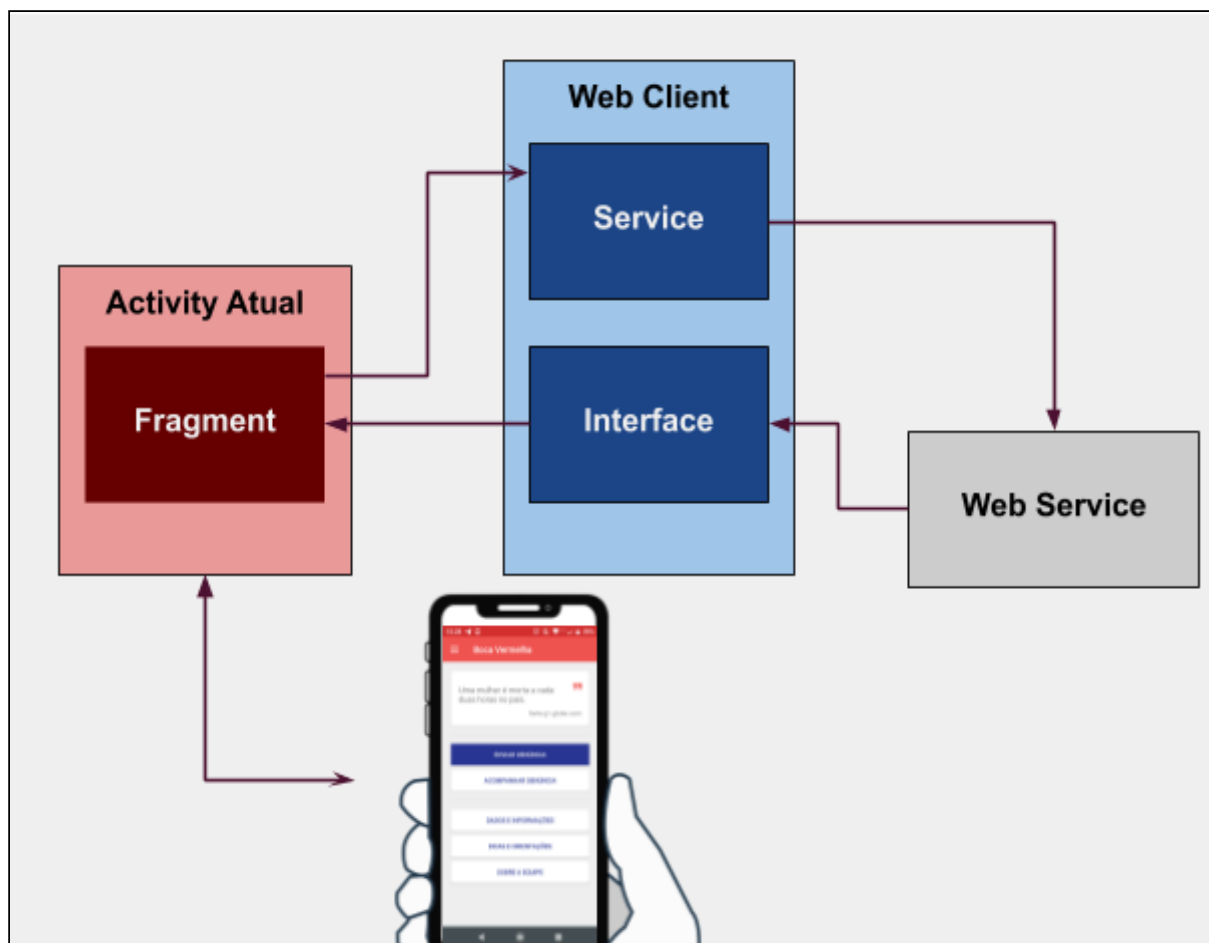
Além da função de realizar e acompanhar as denúncias realizadas, o grupo decidiu que seria interessante que o aplicativo também tivesse um papel informativo ao usuário e,

por isso, foi inserido uma funcionalidade de **dados e informações** que dispõe de gráficos e informações sobre as denúncias já cadastradas e outra de **dicas e orientações** (o funcionamento dessas telas será mais aprofundado a seguir).

Quanto à estrutura do projeto, podemos segmentar os pacotes da seguinte forma

- **Activities:** estrutura de tela básica do android e controle de conteúdo.
  - **MainActivity:** activity principal do projeto e controla qual conteúdo será exibido para o usuário. Sua função é manipular o Navigation Drawer e definir qual fragment será disposto na tela.
  - **Dicas Activity:** como a página de **Dicas e Orientações** possui um layout que necessita controlar muitos textos dinamicamente optou-se por criar uma activity dedicada para essa tela. Além disso, outro motivador de separar essa activity era permitir que o usuário pudesse abrir essa página para leitura sem perder os dados cadastrados no formulário.
- **Fragments:** responsável por definir qual conteúdo será carregado na tela, além de funcionar como ponte entre **Layout - Funcionalidade** fazendo chamadas à camada de Web Service
- **Models:** similar às Entities no Web Service, pois implementa as abstrações de objetos que serão manipulados pelo aplicativo.
- **Service:** pode ser dividido em outras três subcamadas
  - **Service:** realiza o mapeamento ao definir para quais url's serão disparadas as requisições (as mesmas definidas pelo Webservice).
  - **WebClient:** função que será manipulada pelas outras classes do projeto. Ela inicia o Service e define qual requisição será utilizada.
  - **Response:** interface que permite à classe que chamou o WebClient manipular o conteúdo retornado pelo servidor.

Além disso há também os arquivos da pasta **resources** que possuem todas as configurações de cores, textos, temas, ícones e muitas outras configurações importantes para o funcionamento adequado da aplicação

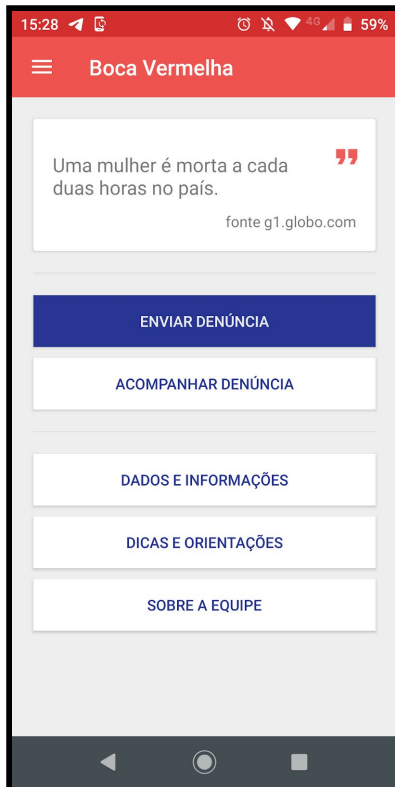


Para a execução do projeto é importante verificar as seguintes configurações:

1. **Importar os arquivos do projeto na IDE Android Studio**
2. **Iniciar o Web Service** no Tomcat e também o **MySQL Server** na máquina.
3. **Encontrar o IP da máquina com o Tomcat:** é preciso verificar o IP da máquina que está servindo como servidor na rede local. É possível testar se o ip encontrado é de fato o que possui o servidor utilizando a seguinte url em um navegador padrão: <http://seu.ip.aqui:8080/bocaVermelha/teste/> e verificando se aparece uma mensagem do Web Service.
4. **Configurando o ip no projeto Android:** o ip do servidor deve ser definido no arquivo *com.example.projetojava.retrofit.RetrofitConfig* e também deve ser configurado no arquivo */res/xml/network\_security\_config.xml*.



## 6. ANDROID - TELAS DO APLICATIVO E SUAS FUNCIONALIDADES

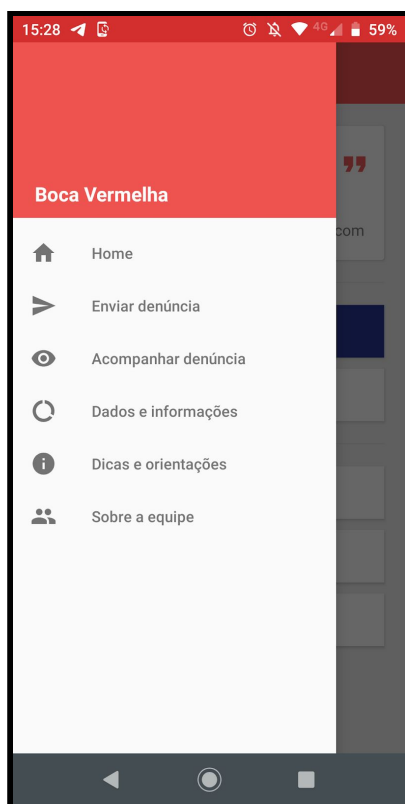


### MAIN ACTIVITY - HOME FRAGMENT

Tela inicial do aplicativo com todas as opções de navegação. O botão de criar denúncia é destacado dos outros conforme o padrão de design do google e indica ao usuário qual é a principal funcionalidade do sistema.

Outra decisão de projeto foi a inserção de um card com a citação de um dado recente acerca da violência contra a mulher para informar ao usuário qual o contexto da violência contra a mulher no Brasil atualmente.

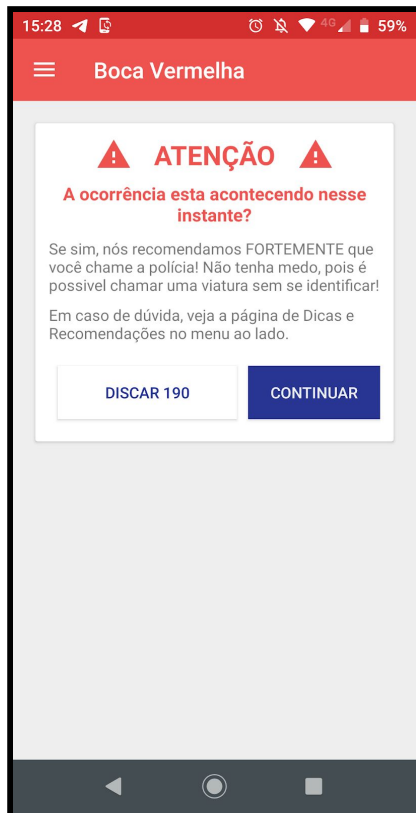
No momento a frase ele é estática mas poderia facilmente ser substituída por uma frase que fosse atualizada periodicamente por meio do Web Service.



### MAIN ACTIVITY - MENU LATERAL

Dentro de todas as telas disponibilizadas pela Main Activity é possível alterar a página atual por meio do menu lateral.

Além de ser um facilitador, o menu lateral também tem um papel importante no cadastro da denúncia pois permite ao usuário abrir a tela de Dicas e Orientações para realizar consultas sobre tipos de violências ou como funciona uma denúncia.



### MAIN ACTIVITY - TELA DENÚNCIA

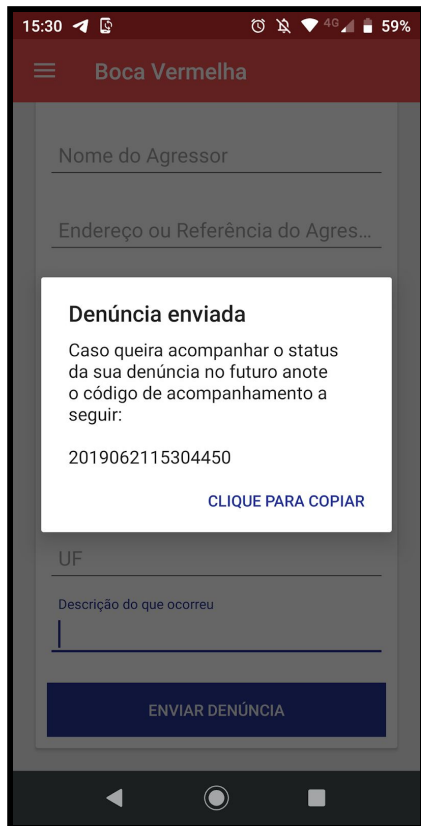
Tela anterior ao formulário de denúncia e que possui instruções para caso a agressão / violação esteja acontecendo nesse instante.

É então dado ao usuário a possibilidade de prosseguir para a denúncia ou então discar para a polícia.

### MAIN ACTIVITY - FORMULÁRIO DE DENÚNCIA

Local onde o usuário irá preencher os dados da denúncia. Caso haja algum dado não preenchido o aplicativo irá indicar o erro ao usuário.

É necessário que o formulário seja preenchido com informações úteis e que permitam ao órgão responsável por investigar a denúncia a localizar tanto a vítima quanto o agressor.

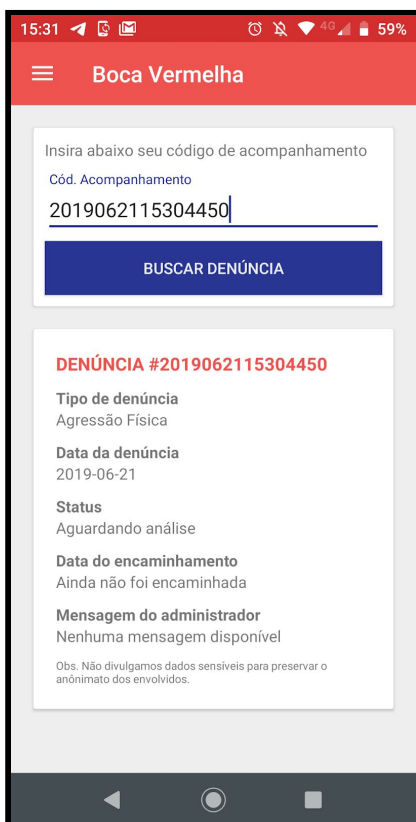


### MAIN ACTIVITY - AO ENVIO DO FORMULÁRIO

Após o envio da denúncia o aplicativo irá enviar os dados para o servidor que irá cadastrar o registro no banco de dados.

O usuário recebe então o código de acompanhamento para verificar o futuro andamento da denúncia.

Uma das desvantagens em não associar a denúncia à alguma conta é o fato de que se o usuário perder o código é impossível recuperá-lo. Porém isso poderia ser contornado ao acrescentar uma opção de configuração que permitisse salvar os códigos gerados no celular, podendo o usuário habilitar ou desabilitar essa opção quando quisesse.



### MAIN ACTIVITY - TELA DE ACOMPANHAMENTO

Permite ao usuário verificar o status atual da sua denúncia ao inserir seu código de acompanhamento.

Veja, no entanto, que a tela não exibe dados sensíveis que possam comprometer a vítima ou identificar o agressor.

Caso a denúncia já tenha sido encaminhada para algum órgão competente ou tenha sido arquivada essa informação também constará na data do encaminhamento.



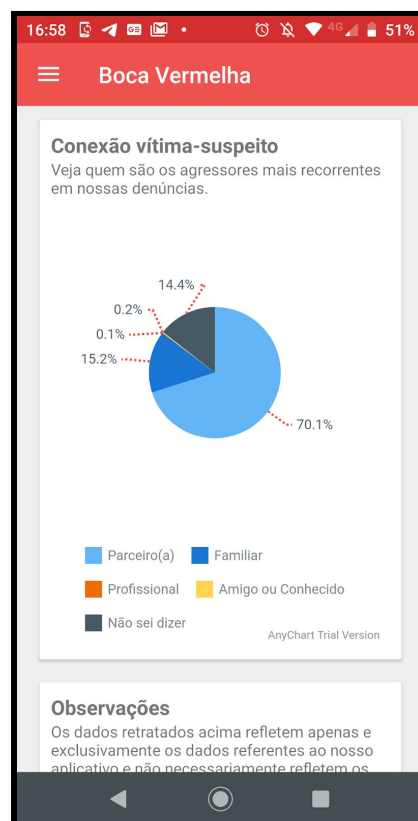
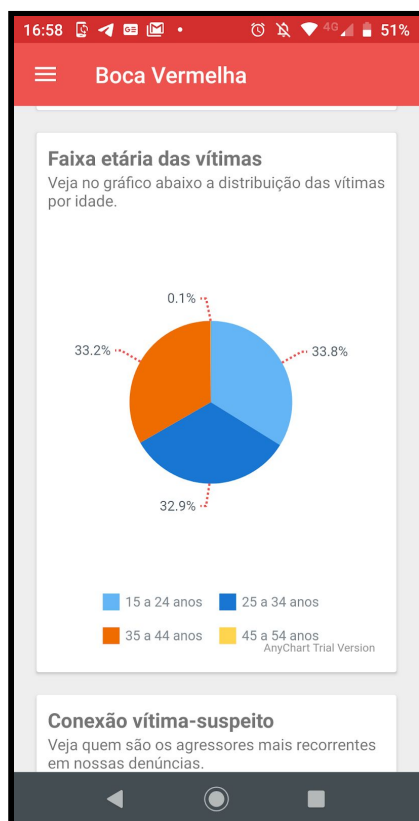
## MAIN ACTIVITY - DADOS E INFORMAÇÕES

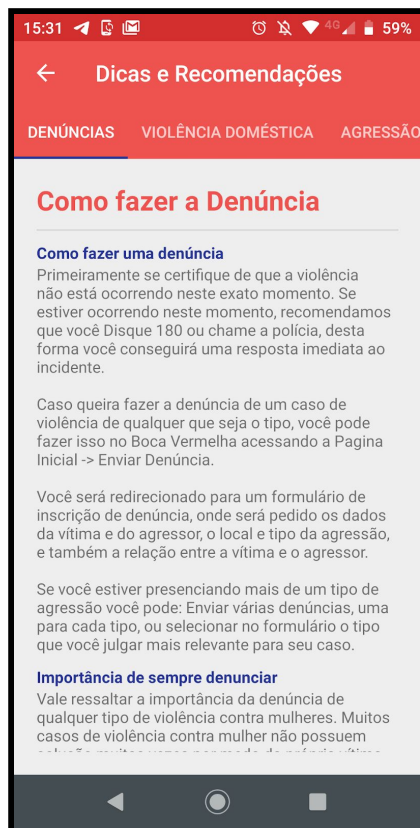
Essa tela tem como objetivo conscientizar os usuários sobre alguns dados obtidos por meio das denúncias cadastradas.

Nessa tela há três gráficos: Denúncias segmentadas por categorias, Vítimas divididas por faixa-etárias e, por fim, os tipos de agressores mais frequentes nas denúncias.

É importante destacar que os dados não são retirados da base de dados do próprio aplicativo e, portanto, reflete apenas à realidade dos usuários do mesmo, não devendo servir generalização para o cenário brasileiro.

Veja abaixo os outros gráficos.

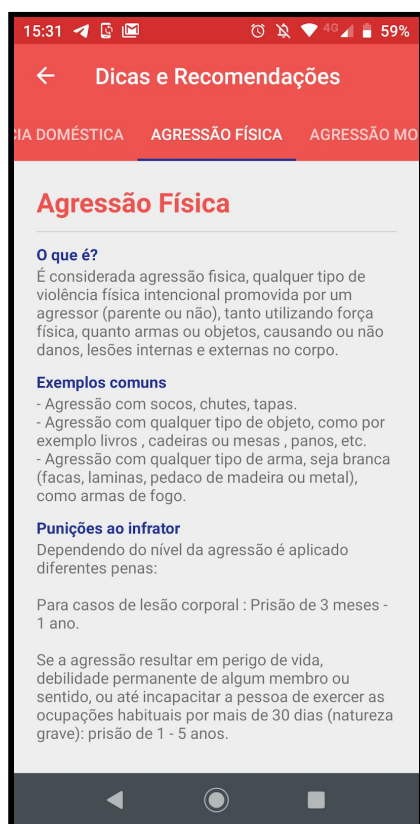




## DICAS ACTIVITY - ABA DENÚNCIA - INICIAL

Essa tela explica para o usuário como funciona o processo da denúncia no aplicativo Boca Vermelha.

O principal objetivo dessa tela é o de explicar ao usuário o que acontece com a denúncia que ele enviou e também a importância da denúncia de nunca ignorar qualquer sinal ou ocorrência que aconteça sob seu conhecimento.



## DICAS ACTIVITY - ABA TIPO DE VIOLÊNCIA

As abas restantes seguem um template padrão, em que cada categoria de violência é explicada sucintamente e em seguida são dados alguns exemplos de ocorrências do tipo.

Por fim, também é exibido ao usuário quais podem ser as punições que um infrator que cometa esse tipo de crime pode receber e quais órgãos geralmente são responsáveis por averiguar tais tipos de casos.

## **7. CONSIDERAÇÕES FINAIS**

Considerando o escopo levantado no documento de proposta, todas as funcionalidade levantadas foram implementadas com sucesso, restando apenas as propostas opcionais (criação de ferramenta desktop para gerência das denúncias) mas que por motivos de complexidade e tempo não puderam ser concluídas.

Por fim, o projeto Boca Vermelha foi realizado com bastante afinco e dedicação por parte do grupo, sempre tentando retratar o problema da violência contra a mulher da maneira mais próxima do cenário brasileiro atual.