

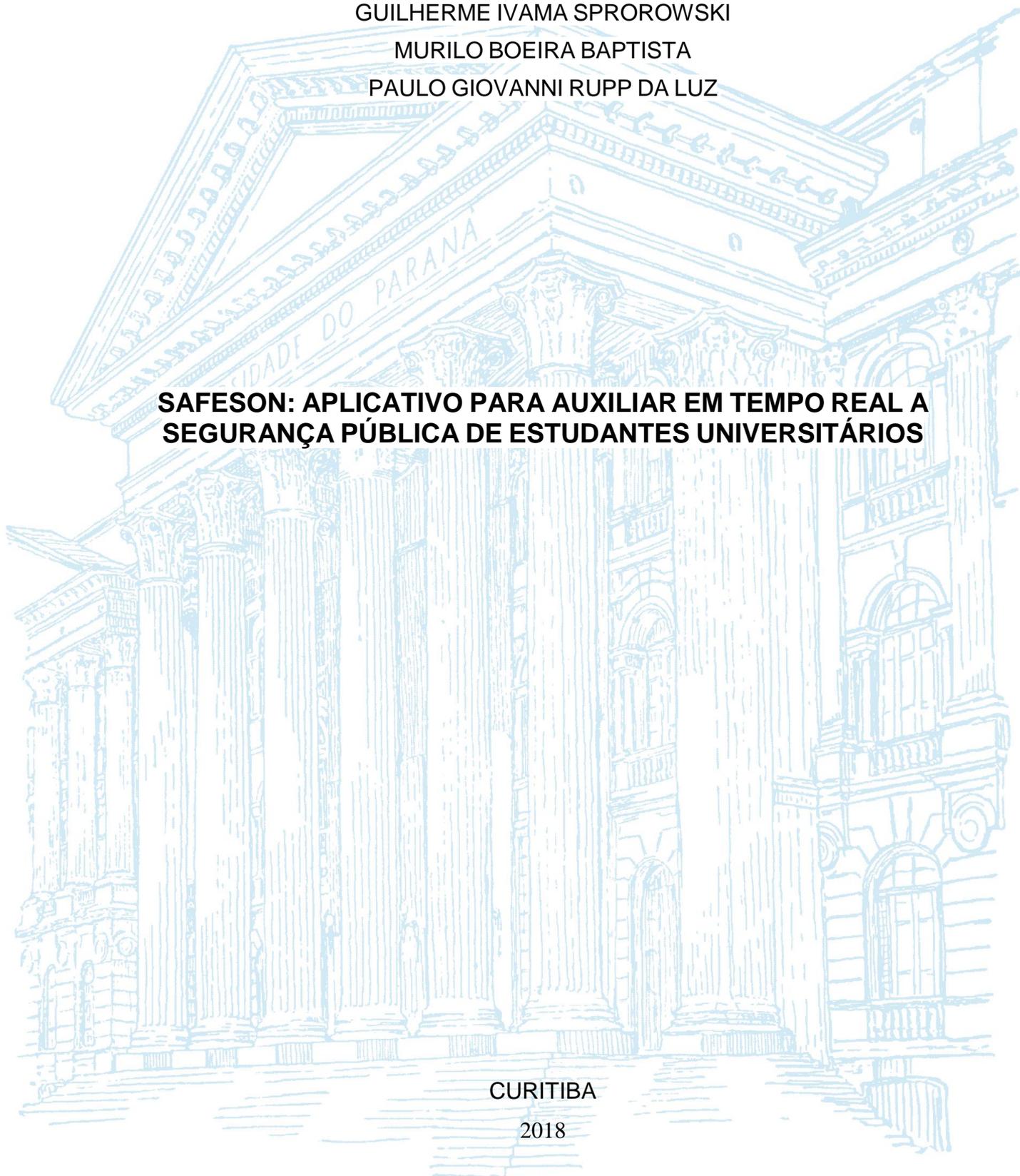
UNIVERSIDADE FEDERAL DO PARANÁ

BRUNO ALVES DE SOUZA  
ELIAS DAL BEN ARRUDA  
GUILHERME IVAMA SPROROWSKI  
MURILO BOEIRA BAPTISTA  
PAULO GIOVANNI RUPP DA LUZ

**SAFESON: APLICATIVO PARA AUXILIAR EM TEMPO REAL A  
SEGURANÇA PÚBLICA DE ESTUDANTES UNIVERSITÁRIOS**

CURITIBA

2018



BRUNO ALVES DE SOUZA  
ELIAS DAL BEN ARRUDA  
GUILHERME IVAMA SPROROWSKI  
MURILO BOEIRA BAPTISTA  
PAULO GIOVANNI RUPP DA LUZ

**SAFESON: APLICATIVO PARA AUXILIAR EM TEMPO REAL A  
SEGURANÇA PÚBLICA DE ESTUDANTES UNIVERSITÁRIOS**

Trabalho De Conclusão de Curso apresentado ao Curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Federal do Paraná como requisito parcial à obtenção de título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientadora: Professora Mestre Andreia de Jesus

CURITIBA  
2018

## TERMO DE APROVAÇÃO

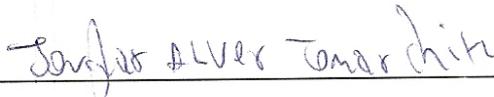
Bruno Alves de Souza  
Elias Dal Ben Arruda  
Guilherme Ivama Sprorowski  
Murilo Boeira Baptista  
Paulo Giovanni Rupp da Luz

SAFESON: Aplicativo para auxiliar em tempo real a segurança pública de  
estudantes universitários

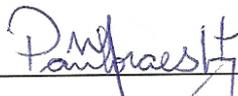
Monografia aprovada como requisito parcial à obtenção do título de  
Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação  
Profissional e Tecnológica da Universidade Federal do Paraná.



Profª. Andreia de Jesus  
Orientadora – SEPT/UFPR



Profº. Jouglas Tomaschitz  
SEPT/UFPR



Profº. Paulo Eduardo Sobreira Moraes  
SEPT/UFPR

Curitiba, 07 de Dezembro de 2018.

“If you say "I told you so", you are the one who has failed. Because you knew, but did not manage to stop the train wreck..” (Robert C. Martin)

## RESUMO

Este projeto tem como objetivo apresentar, documentar e especificar o aplicativo *SafesON*, desenvolvido como trabalho de conclusão de curso do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Federal do Paraná. O sistema proposto neste projeto é um aplicativo para dispositivos móveis *Android*, que tem como objetivo a segurança de seus usuários por meio do compartilhamento de posicionamento em rotas, botão de pânico, alertas de ocorrências e outras funcionalidades. Para o desenvolvimento do *SafesON* foi utilizado *Android Studio*, *MySQL*, *NodeJS*, *Google Maps API* e outras ferramentas. Enfim, o aplicativo *SafesOn* visa: (1) possibilitar mais segurança aos usuários do aplicativo, compartilhando suas rotas em tempo real e informações sobre locais de risco; (2) promover mais agilidade para agir em situações de risco com o acompanhamento em tempo real por terceiros da rota a ser feita pelo usuário.

**Palavras-chave:** Aplicativo, Dispositivo Móvel, Android, Segurança Pública.

## **ABSTRACT**

This project essay seeks to present, document and specify the SafesON application, developed as a course conclusion paper for the graduation of Analysis and Systems Development at Federal University of Paraná. The system proposed in this project is an application for Android mobile devices, aimed at the security of its users through the sharing of route positioning, panic button, event alerts and other functionalities. For the development of this system was used Android Studio, MySQL, NodeJS, Google Maps API and other tools. The application is mainly used to make the user's home path a bit more secure by sharing their paths with the people they want. Ultimately, the SafesON application aims to: (1) provide users with more security of the application, save their routes in real time and information on risk locations; (2) promote more agility to act at risky situations with real-time monitoring by third parties of the route to be made by the user.

**Keywords:** Application, Mobile Device, Android, Public Safety

## LISTA DE FIGURAS

Figura 1 - QUESTÃO SOBRE O MEIO DE LOCOMOÇÃO (POSSÍVEL ESCOLHER MAIS DE UMA RESPOSTA).....	14
Figura 2 - QUESTÃO SOBRE A NECESSIDADE DO APP .....	14
Figura 3 - QUESTÃO SOBRE SEGURANÇA E PERIGO .....	15
Figura 4 - QUESTÃO SOBRE A ADESÃO AO MONITORAMENTO DE TRAJETO .	16
Figura 5 - QUESTÃO SOBRE O VALOR DA SEGURAÇA. ....	17
Figura 6 - EXEMPLO DE MENU LATERAL .....	19
Figura 7 - APLICATIVO UBER .....	19
Figura 8 - OPERATING SYSTEM MARKET SHARE .....	20
Figura 9 - SMARTPHONE OS MARKET SHARE .....	21
Figura 10 - DIAGRAMA SOBRE O SISTEMA ANDROID .....	22
Figura 11 - CICLO DE VIDA DA ACTIVITY.....	26
Figura 12 - CICLO DE VIDA DO SERVIÇO .....	27
Figura 13 - APLICATIVO LIFE 360 .....	28
Figura 14 - APLICATIVO FUI ASSALTADO AQUI .....	29
Figura 15 - APLICATIVO POLICIA POPULAR.....	30
Figura 16 - CICLO DE VIDA DO PROJETO.....	38
Figura 17 - KANBAN NO TRELLO .....	40
Figura 18 - TELA DE LOGIN.....	45
Figura 19 - TELA PRINCIPAL .....	46
Figura 20 - GERAR ROTA .....	47
Figura 21 - SELECIONAR MEIO DE TRANSPORTE .....	47
Figura 22 - INICIAR PERCURSO.....	48
Figura 23 - SELECIONAR OBSERVADORES .....	49
Figura 24 - ADICIONAR ACONTECIMENTO.....	50
Figura 25 - ENVIAR PEDIDO DE AJUDA .....	50
Figura 26 - GERENCIAR AMIGOS .....	51
Figura 27 - GERENCIAR GRUPOS .....	52
Figura 28 - GERENCIAR CONVITES .....	52

## LISTA DE TABELAS

Tabela 1 - TABELA COMPARATIVO DE APLICATIVOS.....	30
Tabela 2 - LEVANTAMENTO DE REQUISITOS FUNCIONAIS.....	32
Tabela 3 - REQUISITOS NÃO FUNCIONAIS .....	33
Tabela 4 TABELA DE BUGS.....	93

## LISTA DE TABELAS

APF - Adaptive Project Framework

APM - Adaptive Project Management

API -Interface de Programação de Aplicativos

ART - Android Runtime

CRUD - Create, Read, Update e Delete

DEX - Dalvik Executable

HAL - Camada de abstração de hardware

SGBD – Sistema Gerenciador de Bancos de Dados

SQL - Structured Query Language

TADS – Tecnologia em Análise e Desenvolvimento de Sistemas

TDD - *Test Driven Development*

UFPR – Universidade Federal do Paraná

UI - Interface do Usuário

Widgets - Componente de uma interface gráfica

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>12</b>
1.1 PROBLEMA .....	13
1.2 JUSTIFICATIVA .....	15
1.3 OBJETIVOS .....	17
1.3.1 Objetivo geral .....	17
1.3.2 Objetivos específicos.....	17
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>18</b>
2.1 CONCEITO DE APLICATIVO MÓVEL.....	18
2.2 A PLATAFORMA <i>ANDROID</i> .....	19
2.2.1 BREVE HISTÓRIA E EXPANSÃO DO <i>ANDROID</i> .....	20
2.2.1 A ARQUITETURA <i>ANDROID</i> .....	21
2.2.2 DESENVOLVIMENTO COM <i>ANDROID</i> .....	24
2.3 TRABALHOS CORRELATOS .....	28
<b>3 MATERIAIS E MÉTODOS</b> .....	<b>31</b>
3.1 LEVANTAMENTO DE REQUISITOS .....	31
3.1.1 ESPECIFICAÇÃO DE REQUISITOS .....	32
A) REQUISITOS FUNCIONAIS .....	32
B) REQUISITOS NÃO FUNCIONAIS.....	33
3.2 ANÁLISE DOS DADOS.....	35
3.3 ANÁLISE DO APLICATIVO.....	36
3.3.1 DIAGRAMA E ESPECIFICAÇÃO DE CASOS DE USO.....	36
3.3.2 DIAGRAMA DE CLASSE .....	37
3.3.3 DIAGRAMA DE SEQUÊNCIA .....	38
3.4 GERÊNCIA DO PROJETO .....	38
3.5 TECNOLOGIAS E LINGUAGENS APLICADAS.....	41
3.6 CASOS SOBRE A IMPLEMENTAÇÃO DO <i>SAFESON</i> .....	43
<b>4 APRESENTAÇÃO DO APLICATIVO</b> .....	<b>44</b>
4.1 O APLICATIVO <i>SAFESON</i> .....	44
4.2 TELAS DO APLICATIVO MÓVEL.....	45
<b>5 CONSIDERAÇÕES FINAIS</b> .....	<b>53</b>
5.1 CONCLUSÃO DO PROJETO .....	53
5.2 TRABALHOS FUTUROS .....	54

<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>56</b>
<b>APÊNDICE A – QUESTIONÁRIO APLICADO NA PESQUISA COM OS UNIVERSITÁRIOS</b> .....	<b>58</b>
<b>APÊNDICE B – DOCUMENTAÇÃO TÉCNICA</b> .....	<b>59</b>
<b>1 DIAGRAMA DE CASOS DE USO</b> .....	<b>59</b>
<b>2 ESPECIFICAÇÃO DE CASO DE USO</b> .....	<b>59</b>
2.1.1 CASO DE USO: UC001 - LOGAR NO APLICATIVO .....	59
2.1.2 CASO DE USO: UC002 - ALERTAR:.....	60
2.1.3 CASO DE USO: UC003 - REPORTAR ACONTECIMENTO: .....	62
2.1.4 CASO DE USO UC004 - ELABORAR ROTA:.....	63
2.1.5 CASO DE USO UC005 - SELECIONAR OBSERVADORES: .....	64
2.1.6 CASO DE USO UC006 - GERENCIAR AMIGOS:.....	66
2.1.7 CASO DE USO UC007 - GERENCIAR GRUPOS: .....	67
2.1.8 CASO DE USO UC008 - VISUALIZAR CONVITES: .....	69
2.1.9 CASO DE USO UC009 - VISUALIZAR ROTAS .....	71
2.1.10 CASO DE USO 010 - CANCELAR ROTA: .....	72
<b>3 DICIONÁRIO DE DADOS</b> .....	<b>73</b>
<b>4 DIAGRAMA DE ENTIDADE E RELACIONAMENTO</b> .....	<b>77</b>
<b>5 DIAGRAMA DE GANTT</b> .....	<b>78</b>
<b>6 DIAGRAMA DE CLASSE(APLICATIVO)</b> .....	<b>80</b>
<b>7 DIAGRAMA DE CLASSE(SERVIDOR)</b> .....	<b>81</b>
<b>8 DIAGRAMA DE CLASSE (CONEXÃO AO SERVIDOR)</b> .....	<b>82</b>
<b>9 DIAGRAMA DE SEQUÊNCIA</b> .....	<b>83</b>
9.1 ALERTA .....	83
9.2 CANCELAR ROTA.....	84
9.3 ELABORAR ROTA.....	85
9.4 GERENCIAR AMIGOS.....	86
9.5 GERENCIAR GRUPOS.....	87
9.6 LOGAR NO APLICATIVO .....	88
9.7 REPORTAR ACONTECIMENTO .....	89
9.8 SELECIONAR OBSERVADORES .....	90
9.9 VISUALIZAR CONVITES .....	91
9.10 VISUALIZAR ROTAS .....	92
<b>10 TABELA DE BUGS CONHECIDOS</b> .....	<b>93</b>

## 1 INTRODUÇÃO

A falta de segurança pública tem sido uma questão recorrente no país tendo impacto, principalmente, em pessoas que estão em traslado no dia a dia, atingindo todas as camadas sociais e trazendo risco ao cidadão independentemente do meio de locomoção, sendo este um problema que também afeta a vida de universitários.

Os dados de violência no país (G1, 2018) são preocupantes e a cada ano os números nos mostram que viver no Brasil tem sido um desafio para muitos. Além dos problemas de desemprego, saúde e educação que as pessoas vivenciam por aqui, a falta de segurança agrega nesse conjunto de malefícios que interferem diretamente na vida das pessoas, as quais não possuem muitas alternativas para garantir ou mesmo aumentar a sua segurança pessoal.

Com o constante desenvolvimento da tecnologia e o constante aumento de sua acessibilidade, cada vez mais surgem aplicativos que agregam e facilitam a vida das pessoas em todos os âmbitos, e com a segurança não é diferente, existindo desde sistemas de rastreamento a automóveis e alarmes, até funcionalidades simples presentes em alguns aparelhos *mobile* e algumas aplicações como mensagens de emergência, porém ainda assim, todos agem de maneira limitada.

Então, considerando as questões citadas, propõem-se o desenvolvimento de uma aplicação *android*, com objetivo de promover mais segurança aos seus usuários. O foco é auxiliar pessoas em traslado, mais especificamente estudantes universitários, os quais representam um grande fluxo de locomoção para diferentes locais; além de que, em muitos casos, surge uma preocupação dos pais quanto à segurança dos filhos; tem-se também uma variedade significativa de meios de locomoção, como será evidenciado nas próximas seções deste documento.

Na sequência são abordados os temas referentes ao desenvolvimento do projeto, segmentados em capítulos. Neste primeiro capítulo é abordada a motivação para o projeto em si, descrevendo o problema encontrado, justificativa e o objetivo a ser alcançado, bem como os objetivos específicos.

No segundo capítulo é descrita a fundamentação teórica do projeto, levando em conta conceitos de aplicativos móveis, estruturação da plataforma *Android* e trabalhos relacionados com o proposto neste projeto.

Já no terceiro capítulo são abordados os materiais utilizados no desenvolvimento do aplicativo, bem como a metodologia aplicada, sendo descritos

todos os requisitos funcionais e não funcionais; além da demonstração dos diagramas elaborados, que foram utilizados como base no desenvolvimento do aplicativo.

O quarto capítulo é voltado para a apresentação do aplicativo, demonstrando suas funcionalidades e a interface desenvolvida para o uso do programa.

Por fim, o último capítulo aborda as considerações finais do projeto e as possibilidades de trabalho futuros.

## 1.1 PROBLEMA

Buscando compreender quais seriam as principais necessidades dos potenciais usuários, além dos principais requisitos do sistema, foi realizada uma pesquisa com 86 universitários da Universidade Federal do Paraná em Curitiba, em que eles responderam um questionário contendo 7 (sete) questões (Apêndice A). Tal pesquisa foi elaborada utilizando a ferramenta Google Forms, por trazer facilidade na hora de elaborar as questões, além de permitir a elaboração de gráficos para exemplificar melhor os resultados obtidos.

Em pesquisa realizada com 86 universitários da Universidade Federal do Paraná em Curitiba, em que eles responderam um questionário contendo 7 (sete) questões (Apêndice A),

Foi constatado, entre as respostas obtidas, que mais de 70% dos respondentes, entre os meios de locomoção até a universidade, não utilizam transporte particular (Figura 1), estando mais vulneráveis ao se locomoverem. Com isso, fica evidente a necessidade e a abrangência que alternativas dentro deste panorama comportam, e a utilização de uma aplicação móvel pode fornecer possibilidades para aumentar a segurança de universitários.

Outro ponto a ser observado, com base no questionário aplicado, é o fato de que 20% dos universitários declararam se sentirem seguros no seu percurso de ida e de volta da universidade (Figura 2), reiterando ainda mais a necessidade de investimentos e apropriação do avanço da tecnologia para oferecer mais segurança às pessoas.

Foi identificado ainda que 14,1% dos questionados já sofreram algum tipo de abordagem intimidadora em seu percurso (Figura 3), sintetizando a insegurança existente entre muitos universitários.

Além destes fatos, é evidente a incapacidade existente de ação por parte da segurança pública no momento de abordagens e assaltos, sendo esta uma

ineficiência que pode ser minimizada ou até mesmo combatida com o uso da tecnologia e com o avanço da comunicação e compartilhamento de informação. Fatores estes pouco explorados ainda nos dias atuais com este objetivo, tornando a segurança pública ainda mais limitada ao tentar combater crimes e evitar sua ocorrência.

### Qual o meio de locomoção que você mais utiliza?

85 respostas

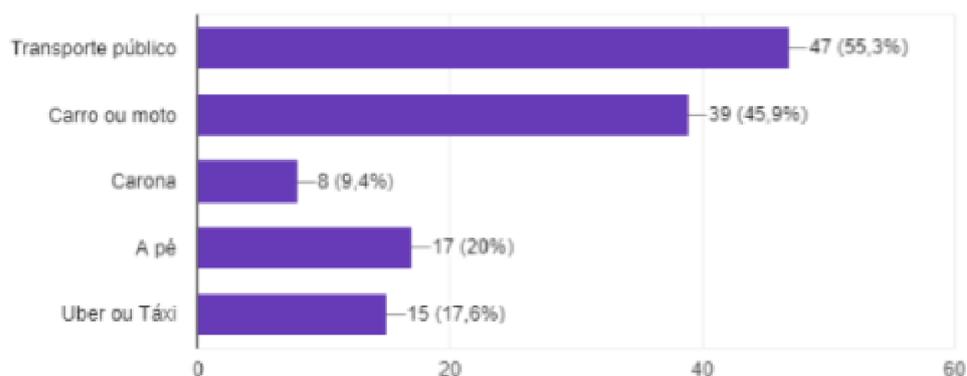


Figura 1 - QUESTÃO SOBRE O MEIO DE LOCOMOÇÃO (POSSÍVEL ESCOLHER MAIS DE UMA RESPOSTA)

FONTE: Os autores (2018).

### Você se sente seguro(a) em seu percurso de ida e volta da universidade?

85 respostas

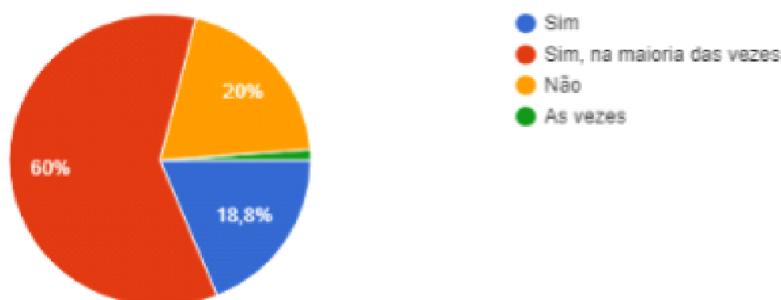


Figura 2 - QUESTÃO SOBRE A NECESSIDADE DO APP

FONTE: Os autores (2018).

No percurso de ida ou volta da universidade você já foi assaltado(a) ou sofreu alguma abordagem intimidadora?

85 respostas

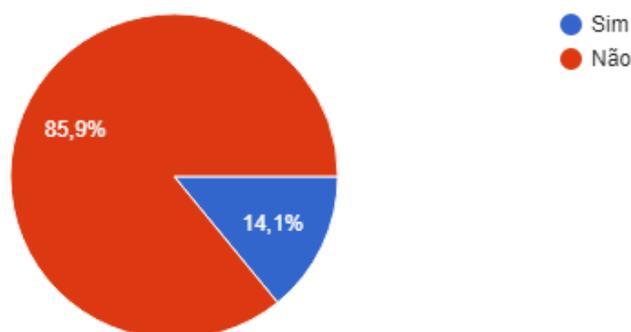


Figura 3 - QUESTÃO SOBRE SEGURANÇA E PERIGO

FONTE: Os autores (2018).

## 1.2 JUSTIFICATIVA

Com o avanço tecnológico existente nos dias atuais, a possibilidade por mais segurança e melhoria na qualidade de vida das pessoas torna-se cada vez mais alcançável, e um grande fator que viabiliza e impulsiona tudo isso é a possibilidade de se utilizar a comunicação como um fator a favor da segurança.

Com os dados obtidos na pesquisa mencionada (Figura 4), fica evidente como o compartilhamento de informação entre os usuários torna-se uma prática fundamental na busca por mais segurança, e uma forma de atingir esse objetivo passa pela utilização de aplicativos móveis. E isso é possível devido o aumento do uso desses dispositivos na sociedade, pois são ferramentas importantes em prol da comunicação entre os usuários, fator este a ser explorado, visando a segurança pública.

A facilidade no compartilhamento de informações dentro deste panorama evidencia sua eficiência, sendo a maneira mais adequada na busca por agregar e promover a interação entre os usuários, a fim de promover mais segurança.

Você se sentiria mais seguro(a) com algum conhecido monitorando seu trajeto de ida e volta da universidade através de um aplicativo?&nbsp;

86 respostas

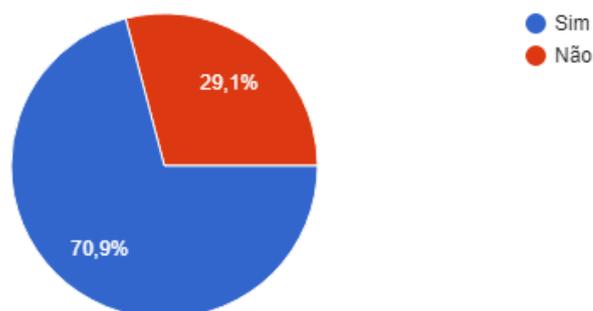


Figura 4 - QUESTÃO SOBRE A ADESÃO AO MONITORAMENTO DE TRAJETO

FONTE: Os autores (2018).

Tendo em vista os problemas apresentados na seção 1.1, fica evidente a possibilidade do uso de uma tecnologia que visa a melhoria da segurança pública de maneira eficiente. Sendo que, em um primeiro momento, o foco é estudantes universitários em seu percurso até a universidade.

Ainda com relação ao questionário aplicado, constatou-se a efetividade de duas funcionalidades principais no projeto proposto: (1) o monitoramento de trajeto e (2) a geração de rotas. Dentro do panorama apresentado na pesquisa, mais de 70% dos questionados declararam que se sentiriam mais seguros caso pudessem contar com o monitoramento de um conhecido durante seu percurso até a universidade (Figura 4), além de 82,6% declarar que utilizaria um percurso mais longo caso este fosse mais seguro (Figura 5).

Logo, existindo um grande número de pessoas com possibilidade de compartilhar informação facilmente, a utilização da coletividade e da interação entre os usuários pode resultar em um conhecimento a ser distribuído em prol de mais segurança pública. Então, visando promover e manipular este conhecimento, o aplicativo *SafesOn* é proposto para: (1) possibilitar mais segurança aos usuários do aplicativo, compartilhando suas rotas em tempo real e informações sobre locais de risco; (2) promover mais agilidade para agir em situações de risco com o acompanhamento em tempo real por terceiros da rota a ser feita pelo usuário.

Você faria um caminho mais longo para o seu destino caso fosse mais seguro?

86 respostas

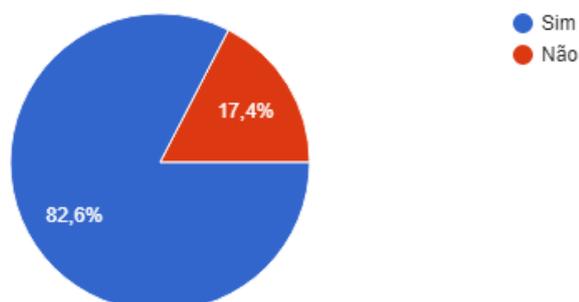


Figura 5 - QUESTÃO SOBRE O VALOR DA SEGURANÇA.

FONTE: Os autores (2018).

### 1.3 OBJETIVOS

#### 1.3.1 Objetivo geral

Desenvolvimento de um aplicativo, denominado *SafesON*, para auxiliar em tempo real a segurança pública de estudantes universitários em seus trajetos entre universidade e seu destino e vice-versa.

#### 1.3.2 Objetivos específicos

- Identificar junto a universitários a necessidade de um aplicativo que auxilie na segurança pública;
- Levantar requisitos junto a universitários para a construção de um aplicativo que auxilie na segurança pública.
- Traçar rotas seguras com base em áreas demarcadas de alto risco no mapa-múndi fornecido pelo *Google Maps*;
- Desenvolver funcionalidades para situações de emergência;
- Desenvolver ferramenta para criação de grupos para que o usuário possa compartilhar em tempo real a sua rota com um conjunto de usuários em específico.
- Desenvolver recursos para a interação dos usuários com o aplicativo em tempo real.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 CONCEITO DE APLICATIVO MÓVEL

Aplicativo móvel, mais conhecido como APP, é um programa designado para funcionar em dispositivos móveis, como celular ou tablet (TECHNOPEDIA, 2018). Essas aplicações são geralmente criadas para servir propósitos similares a programas que podem ser acessados pelo computador. Aplicativos geralmente não são aplicações com múltiplas funções integradas, possuem apenas uma função específica, onde o usuário tem liberdade para escolher o que adicionar ao seu dispositivo dependendo das suas necessidades.

Os dispositivos móveis são práticos e úteis, porém em sua maioria possuem menos espaço para a interface que outras plataformas e não possuem controles precisos para jogos, desenho e outros aplicativos. Para resolver o problema de espaço é muito utilizado menus laterais que se encolhem e expandem em resposta ao usuário (FIGURA 6) e telas específicas para apenas uma funcionalidade. Já para o segundo problema citado é necessário o uso de hardware externo que se conectem com o dispositivo móvel, o que cria um desafio para o desenvolvedor, pois nem todos que utilizarem o aplicativo vão possuir as peças necessárias.

O desenvolvimento de um aplicativo móvel geralmente parte de um conceito de software que já está concluído para outras plataformas, porém conforme são inseridas novas funcionalidades nos celulares as possibilidades de aplicativos aumentam e alguns que não são viáveis em outras mídias surgem como, por exemplo, o Uber (FIGURA 7). Que é um aplicativo de transporte privado urbano, que utiliza o sistema de localização para encontrar os motoristas mais próximos do cliente que realizou o pedido de viagem e atendê-lo o mais rápido possível.

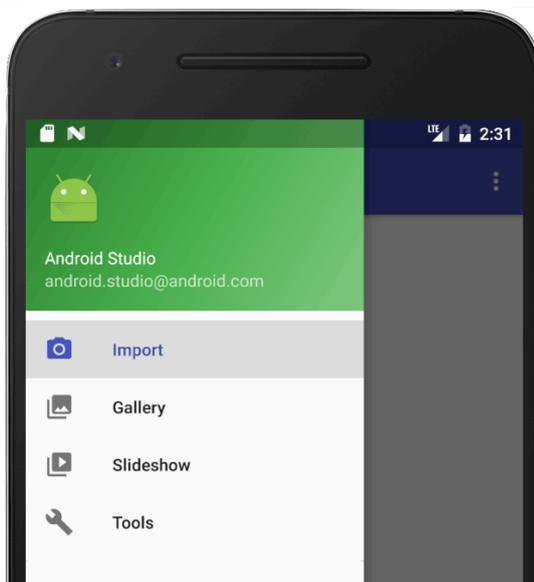


Figura 6 - EXEMPLO DE MENU LATERAL

FONTE: (Criação de uma gaveta de navegação, 2018)

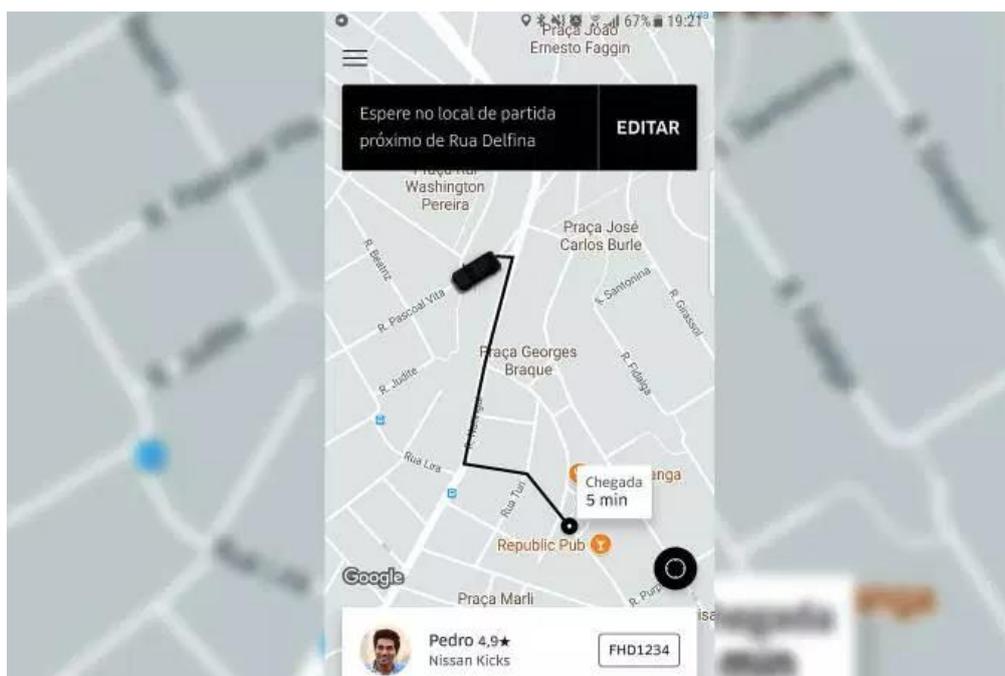


Figura 7 - APLICATIVO UBER

FONTE: (Uber vai mostrar no mapa a cor do carro solicitado, 2017)

## 2.2 A PLATAFORMA ANDROID

*Android* é um *software* de código-fonte aberto desenvolvido pela *Google* para dispositivos que possuem *touchscreen*, como celulares e *tablets* (DEVELOPERS, 2018). Trata-se de um sistema operacional e, atualmente, encontra-se na versão 9.0.

### 2.2.1 BREVE HISTÓRIA E EXPANSÃO DO ANDROID

O *Android* foi inicialmente desenvolvido pela *Android Inc.* Após ser comprada pela *Google*, em 2005, a empresa continuou desenvolvendo o seu sistema operacional baseado no núcleo *Linux* para protótipos parecidos com os telefones *BlackBerry*, sem *touchscreen* e com um teclado físico. Contudo, com o lançamento do *iPhone, smartphone* da *Apple*, o projeto inicial foi refeito para possuir suporte a *touchscreen*. Então, em 2008, foi lançado a primeira versão do *Android*, *Android 1.0 - Alpha*. Foi definido que as versões teriam nomes em ordem alfabética e a partir da versão 1.5 - *Cupcake* cada lançamento possuiria um nome de um doce diferente.

Para competir com o seu principal rival, *iOS* da *Apple*, o *Android* já passou por grandes atualizações, para melhorar sua performance e adicionar novas *features*. E um fator que impulsiona o objetivo de conquistar o mercado passa pelo apoio à *Google* por parte da *Open Handset Alliance - OHA*. O OHA é um grupo que ajuda no desenvolvimento da plataforma e conta com empresas como *Intel, Samsung, LG, HTC, Sony Ericsson, Dell, Acer, Motorola*, etc. Esse grupo possui o objetivo de desenvolver um sistema móvel de código-aberto que seja poderoso e flexível, atendendo a necessidade de todos os envolvidos. Graças a colaboração deste grupo o *Android* é o sistema operacional móvel mais utilizado hoje em dia (LECHETA, 2015).

No mercado de sistemas operacionais, o *Android* possui 39,9% da fatia em 2018 e vem conquistando o primeiro lugar entre os sistemas operacionais, passando até na frente do *Windows* da *Microsoft*, com 36,45% em 2018, que esteve décadas com a maior fatia do mercado (Figura 8).

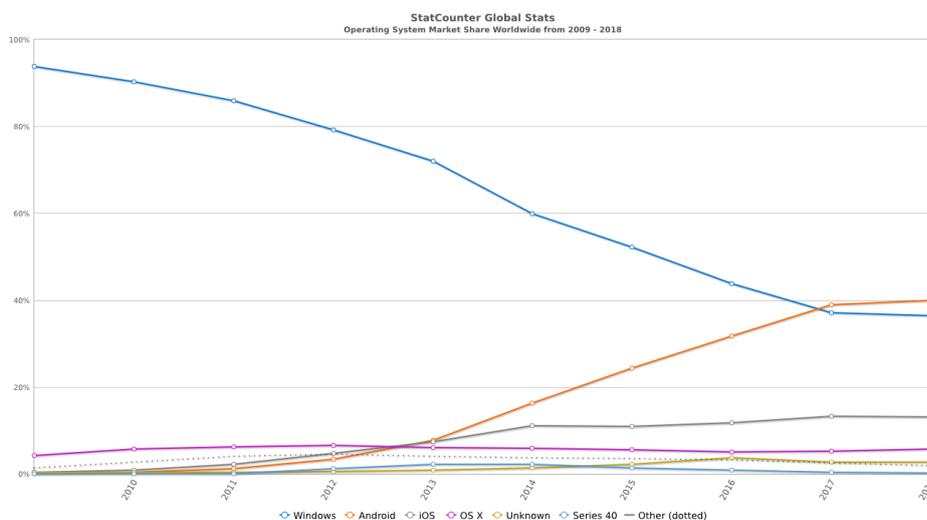
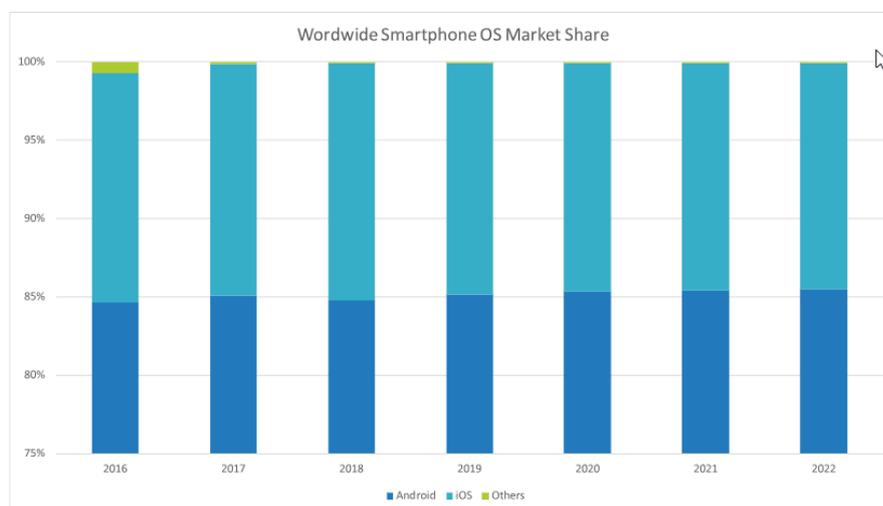


Figura 8 - OPERATING SYSTEM MARKET SHARE

FONTE: (Statcounter, 2009 - 2018)

De acordo com a pesquisa da IDC (2018), o *Android* atingiu 84,8% do mercado de *smartphones* (FIGURA 9), com apenas um competidor relevante, o *iOS*, com 15,1% no mesmo ano. É esperado o crescimento neste mercado, porém nada extremamente significativo, pois já possui grande parte do mesmo.



Year	2016	2017	2018	2019	2020	2021	2022
Android	84,6%	85,1%	84,8%	85,2%	85,3%	85,4%	85,5%
iOS	14,7%	14,7%	15,1%	14,8%	14,6%	14,6%	14,5%
Others	0,7%	0,2%	0,1%	0,1%	0,1%	0,1%	0,1%
TOTAL	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%

Figura 9 - SMARTPHONE OS MARKET SHARE

FONTE: (Statcounter, 2009 - 2018)

### 2.2.1 A ARQUITETURA ANDROID

O sistema operacional *Android* possui muitas ferramentas, essa seção fala sobre esse aspecto da plataforma. O diagrama da FIGURA 10 apresenta os principais componentes da arquitetura *Android*.

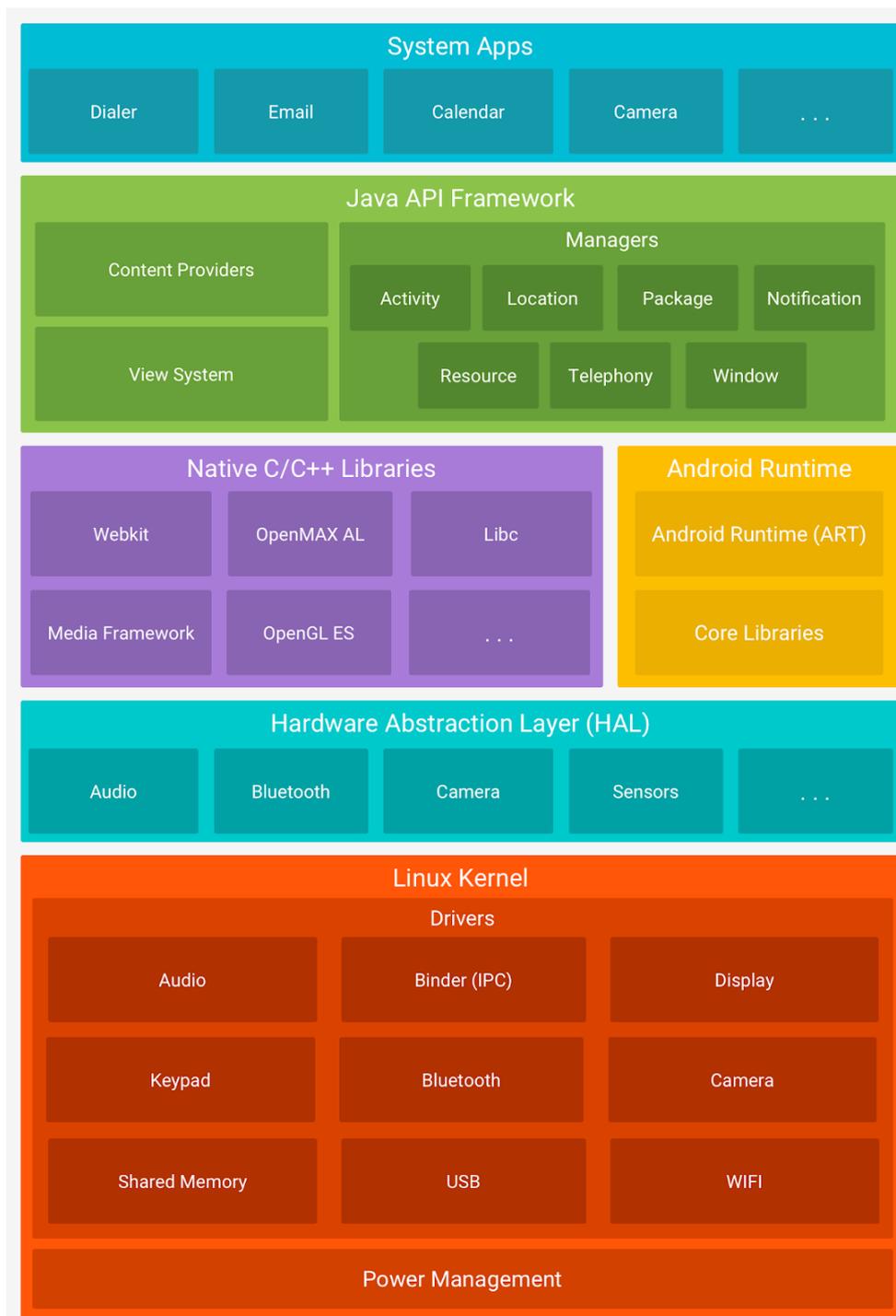


Figura 10 - DIAGRAMA SOBRE O SISTEMA ANDROID

FONTE: (Platform Architecture, 2018)

A seguir uma breve explicação do que cada componente da FIGURA 10 significa e representa na arquitetura Android (DEVELOPERS, 2018):

- Aplicações do sistema (*System Apps*) – cor azul no topo: essa é a camada mais superior na pilha da plataforma e é composta de aplicações que foram desenvolvidas como, por exemplo, Alarme, Relógio, Calculadora, Calendário,

entre outras. Nessa camada também ficam as aplicações desenvolvidas por terceiros e, também, é fornecida a possibilidade de substituição dos aplicativos principais.

- Estrutura de Aplicação *Java (Java API Framework)* - cor verde: constituída por serviços de alto nível e APIs. Essa camada é utilizada pelas aplicações. Segundo KANJILAL (2016), os principais serviços da plataforma *Android* são: (1) Gerenciador de Atividades (*Activity Manager*) - gerencia o ciclo de vida dos aplicativos fornecendo uma pilha de navegação inversa, para quando o usuário clicar no botão “Voltar” ele possa navegar para atividades anteriores na ordem que foram abertas (DEVELOPERS, 2018); (2) Provedores de Conteúdo (*Content Providers*) - Esse serviço permite a comunicação de dados entre aplicativos; (3) Gerenciador de recursos (*Resource Manager*) - fornece acesso a recursos sem código, como imagens, strings, gráficos, arquivos e outros; (4) Gerenciador de Localização (*Location Manager*) - contém classes que definem serviços baseados na localização do *Android*; (5) Gerenciador de Notificações (*Notification Manager*) - esse gerenciador garante permissão para aplicativos utilizarem notificações personalizadas na barra de status; (6) Sistema de Visualização (*View System*) - composto por vários elementos de UI, esse componente ajuda o desenvolvedor a criar interfaces fáceis e intuitivas; (7) Gerenciador de Chamadas (*Telephony Manager*) - fornece informações sobre os serviços de telefonia do celular. Aplicações podem utilizar qualquer método desta classe para determinar os serviços e estados do telefone, assim como algumas informações de inscritos (DEVELOPERS, 2018).
- Bibliotecas C/C++ Nativas (*Native C/C++ Libraries*) - cor roxa: essas bibliotecas dão suporte a serviços, componentes e aplicações que foram escritas em C ou C++, como os componentes ART e HAL (DEVELOPERS, 2018). A arquitetura de APIs *Java* procura fornecer as funcionalidades de algumas dessas bibliotecas nativas para o desenvolvedor, por exemplo: o banco de dados *SQLite (SQLite Library)*; o mecanismo para navegadores; a representação de imagens 2D e 3D; a *media codecs* para áudio e vídeo (STUDYTONIGHT, 2018).
- *Android Runtime (ART)* - cor laranja: esse componente consiste em uma máquina virtual que compila os aplicativos em *bytecode* e, diferentemente de

seu predecessor *Dalvik*, que utiliza just-in-time (JIT), técnica de compilação que ocorre durante a utilização do aplicativo. O ART também possui *ahead-of-time* (AOT), que compila aplicativos antes deles serem utilizados, consumindo menos memória no uso e, conseqüentemente, agilizando o aplicativo e utilizando mais espaço no disco do dispositivo (SOURCE, 2018). Além disso, o ART obriga o sistema operacional a possuir um ótimo gerenciamento de processos, pois desde a versão 5.0 do *Android* cada aplicativo possui uma instância própria dessa máquina virtual, fazendo com que o tempo de *boot* seja cada vez maior (STUDYTONIGHT, 2018).

- Camada de Abstração de *Hardware* (HAL) - cor azul-esverdeado: essa camada é uma interface padrão para a implementação dos fabricantes de *hardware*, assim mudanças no *hardware* não afetam o *Android*. Utilizar o HAL permite a implementação de funcionalidades sem afetar ou modificar as camadas superiores (SOURCE, 2018).
- *Linux Kernel* - cor vermelha: é um núcleo monolítico que providencia serviços para que as funcionalidades do sistema operacional *Android* sejam implementadas. O desenvolvimento da plataforma foi baseado nesse componente.

### 2.2.2 DESENVOLVIMENTO COM ANDROID

Para o desenvolvimento deste projeto foi utilizado o *Android Studio*, uma IDE para desenvolver para o sistema operacional *Android*. As principais características dessa ferramenta são (DEVELOPER, 2018):

- *Gradle*: sistema de automação de *builds*, assim como Maven e Ant;
- Compatibilidade com NDK e C++;
- Emulador de dispositivos móveis;
- Ambiente compatível de desenvolvimento para qualquer dispositivo *Android*;
- Ferramentas de verificação para encontrar códigos que criam gargalos.

A seguir é apresentado alguns conceitos fundamentais para o desenvolvimento Android:

#### a) *Activity*

Toda tela com funcionalidades no *Android* possui uma *activity* para realizar a lógica programada pelo desenvolvedor, geralmente existindo alguma interação com o

usuário. Aplicativos possuem várias atividades e, geralmente, possuem uma atividade principal que é apresentada ao usuário ao iniciar o aplicativo (DEVELOPERS, 2018).

A *activity* é uma classe *Java* que possui um ciclo de vida (FIGURA 11) e funcionalidades pré-desenvolvidas (SOBRINHO, 2018):

- *onCreate()*: chamado após a atividade ser criada pela primeira vez, esse método configura inicialmente o aplicativo, definindo a UI e buscando os *widgets* necessários;
- *onStart()*: chamado quando a aplicação se torna visível para o usuário;
- *onRestart()*: chamado após *onStop()*, quando a atividade é resumida pelo usuário, esse método vai ser seguido pelo *onStart()*;
- *onResume()*: chamado para a atividade começar a interagir com o usuário;
- *onPause()*: esse método é chamado quando a atividade foi minimizada ou outra atividade foi aberta e ainda não foi finalizada com o método *onDestroy()*;
- *onStop()*: chamada quando a atividade está no *background* sem estar visível ao usuário;
- *onDestroy()*: chamado quando a atividade vai ser finalizada por algum motivo com o objetivo de realizar as tarefas finais da aplicação.

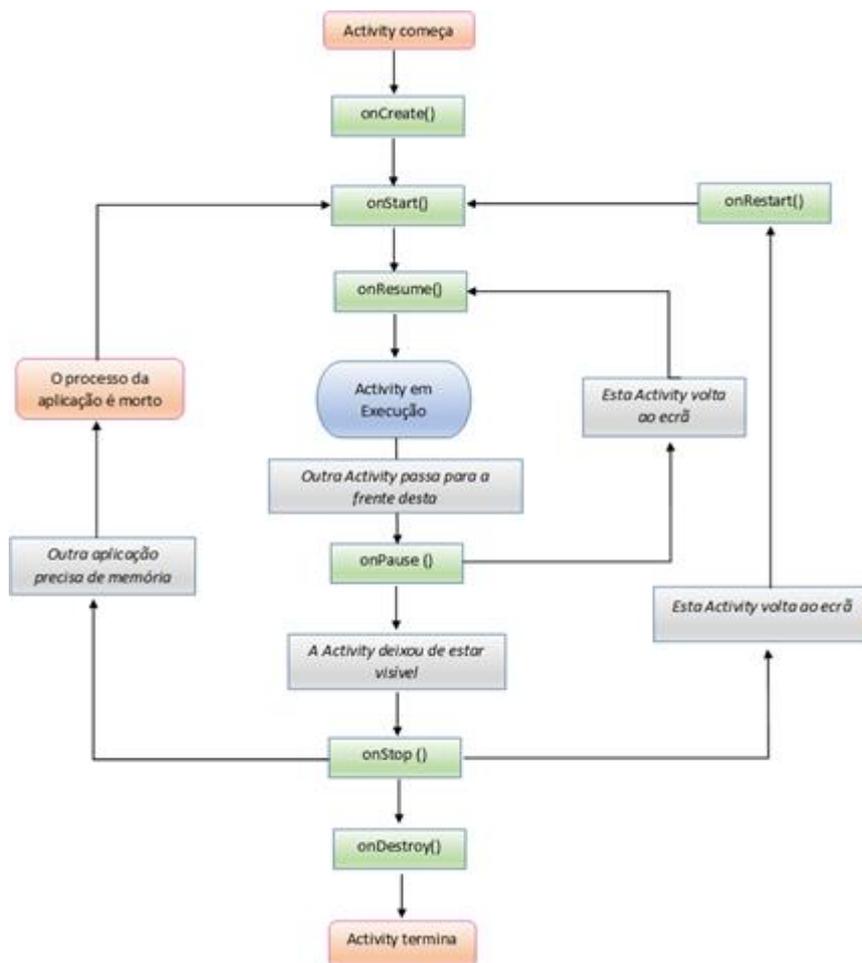


Figura 11 - CICLO DE VIDA DA ACTIVITY

FONTE: SOBRINHO (2018).

## b) Serviços

São componentes de um aplicativo que realizam operações longas e não possuem interface. Após ser iniciado, o serviço fica no *background* até que o aplicativo seja finalizado, segundo Developers (2018), existem duas formas de serviço: (1) Iniciado - um serviço iniciado geralmente é um *download* ou *upload* de arquivos e pode ficar em segundo plano indefinidamente, mesmo que o componente que iniciou o serviço seja destruído. Utiliza-se o método *startService()* para iniciar essa forma de serviço; (2) Vinculado - um serviço vinculado é uma conexão servidor-cliente permitindo os componentes fazerem requisições, solicitações e obterem resultados, sendo destruído quando o componente desfaz o vínculo com o serviço. Utiliza-se o método *bindService()* para vinculá-lo.

Assim como a *activity*, um serviço possui um ciclo de vida (FIGURA 12) e alguns métodos únicos pré-desenvolvidos:

- *onStartCommand()*: chamado toda vez que um serviço é iniciado pelo método *startService()*;
- *onBind()*: retorna um canal de comunicação para um serviço vinculado;
- *onUnbind()*: chamado quando todos componentes se desconectaram do serviço vinculado;

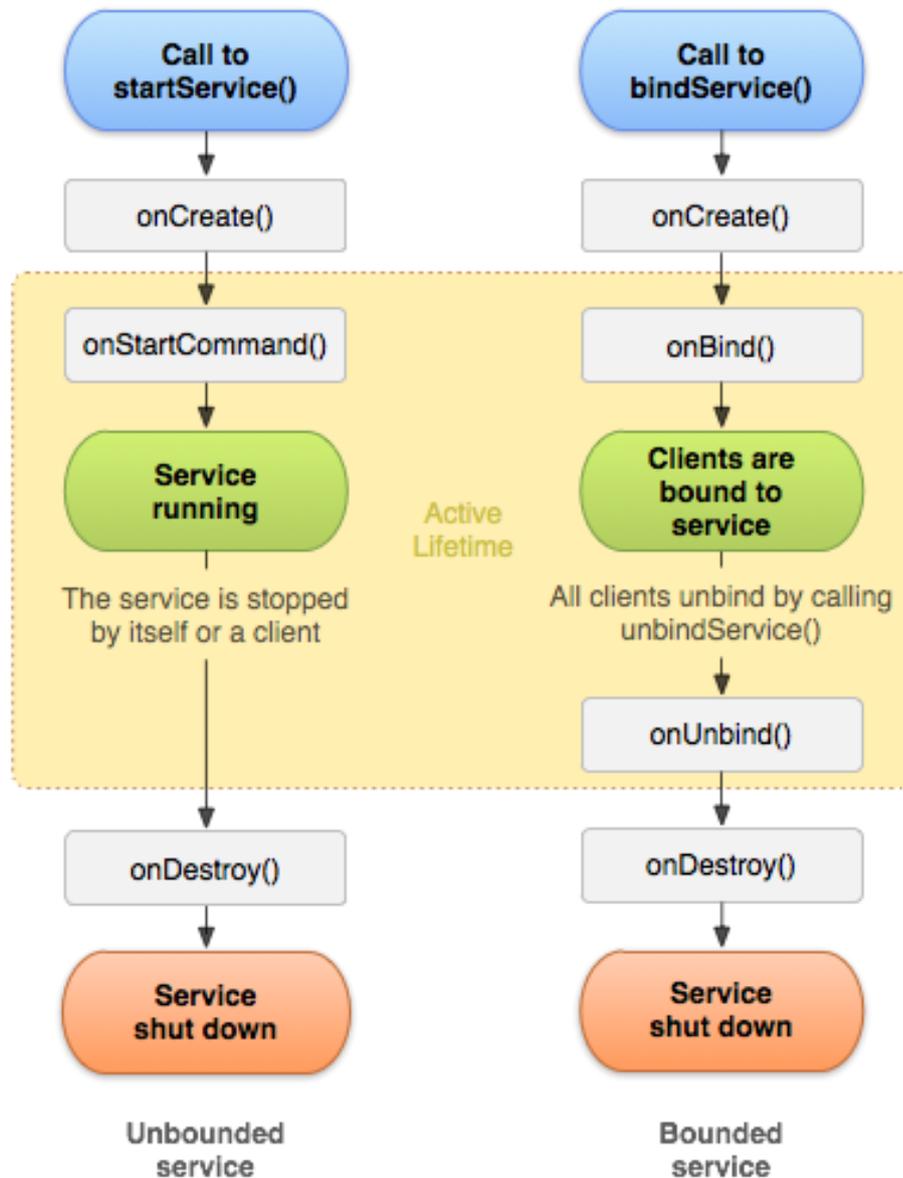


Figura 12 - CICLO DE VIDA DO SERVIÇO

FONTE: Android Developers (2018).

O que foi mostrado nesta seção são algumas ferramentas que o sistema Android fornece para o desenvolvimento de um aplicativo. Para muitos aplicativos terem um bom funcionamento é necessário que ambas existam, pois a ferramenta de serviço é essencial para algumas atividades que acontecem enquanto o usuário está com o aplicativo no background ou com o dispositivo móvel desligado, já a segunda

ferramenta que foi mencionada, a classe Activity, tem como funcionalidade principal criar a interatividade da tela que está sendo apresentada ao cliente. Apesar da plataforma possuir muitos outros elementos que ajudam no desenvolvimento esses são os principais utilizados para a construção de aplicativos.

## 2.3 TRABALHOS CORRELATOS

Os subitens a seguir são os aplicativos correlatos ao aplicativo *SafesON*.

### a) *LIFE 360*

Esse aplicativo (FIGURA 13) é um localizador com foco em família, seus pontos fortes são:

- Compartilhamento da localização para usuários convidados ou grupos selecionados;
- Grupos com as funcionalidades de localização e conversa
- Alertas por grupo sobre a posição de um usuário;
- Rastreo de telefone perdido;
- Modo pânico que o usuário pode ativar.

Seus pontos fracos são:

- Pode ser utilizado de forma invasiva;
- Não possui áreas de perigo ou ocorrências próximas.

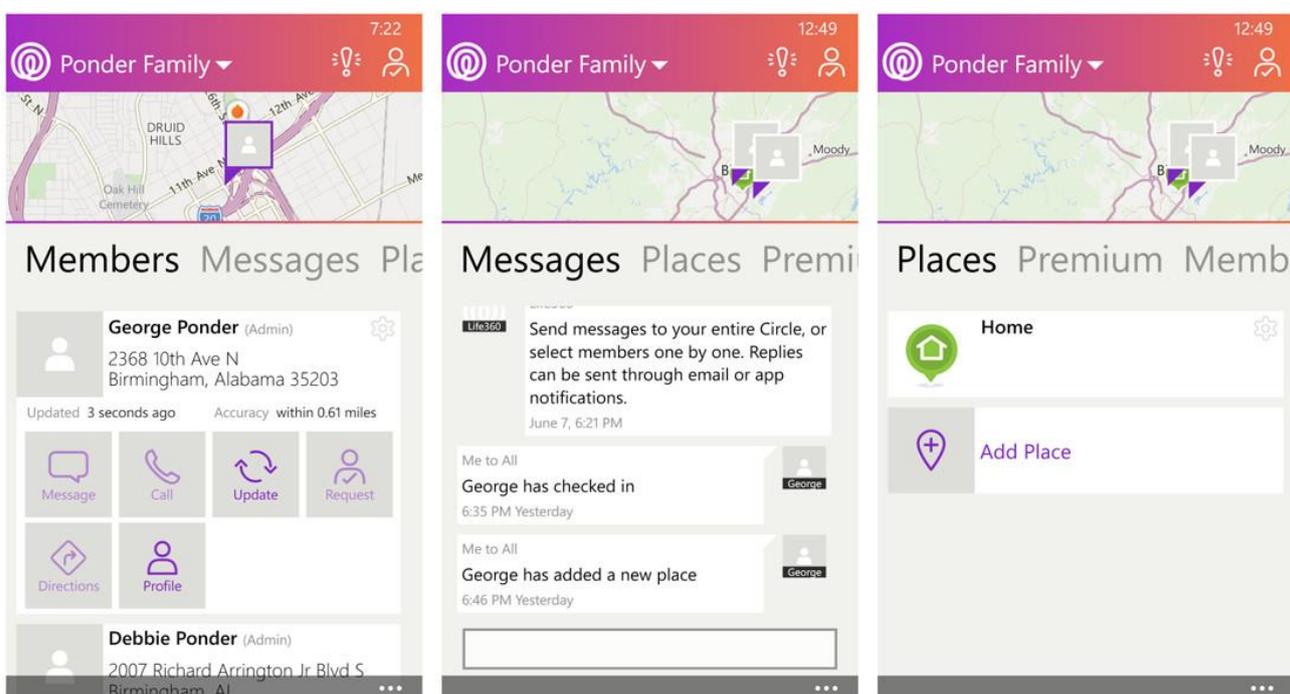


Figura 13 - APLICATIVO LIFE 360

FONTE: (Life360 for Windows Phone, an easy way to keep track of your Family, 2014).

## b) FUI ASSALTADO AQUI

O aplicativo “Fui Assaltado Aqui” (FIGURA 14) tem o foco na funcionalidade de pontos em que usuários foram assaltados. Seus pontos fortes são:

- Mostra ocorrências no mapa;
- Possui uma interface simples e de fácil entendimento;
- Navegação do mapa pelo endereço.

Seus pontos fracos são:

- Não possui locais favoritos, como casa ou trabalho;
- Sem o *feedback* do usuário causa o desuso do aplicativo;
- Não possui outras funcionalidades.

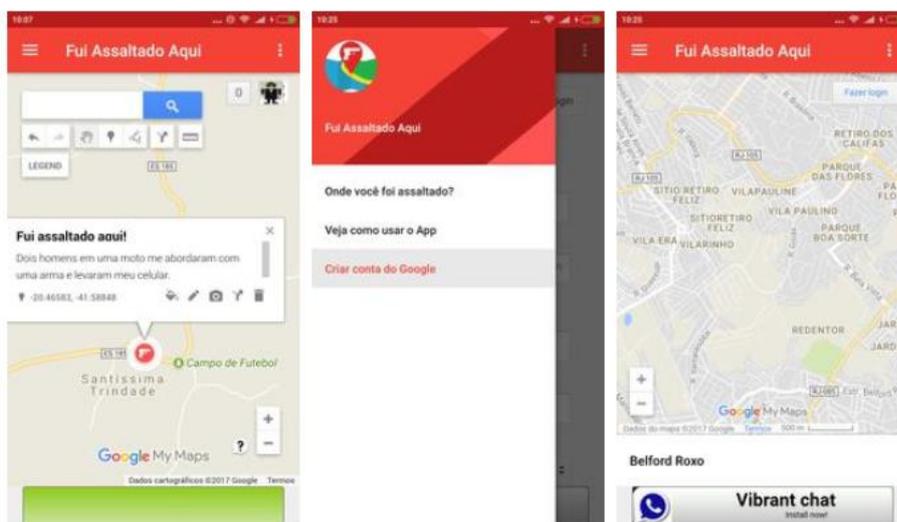


Figura 14 - APLICATIVO FUI ASSALTADO AQUI

FONTE: (Fui Assaltado Aqui, 2018).

## c) Polícia Popular - Segurança Pública

Este aplicativo (FIGURA 15) tem foco na segurança do bairro do usuário, seus pontos fortes são:

- Alertas de perigos próximos de sua localidade;
- É possível criar grupos de conversa;
- Possui mapa com locais favoritos, delegacias e hospitais;
- Possui configuração de preferência de alertas.

Seus pontos fracos são:

- Não permite que o usuário compartilhe sua rota;
- Não possui botão de pânico;
- Se não houver usuários relatando alertas, não existe perigos no mapa.

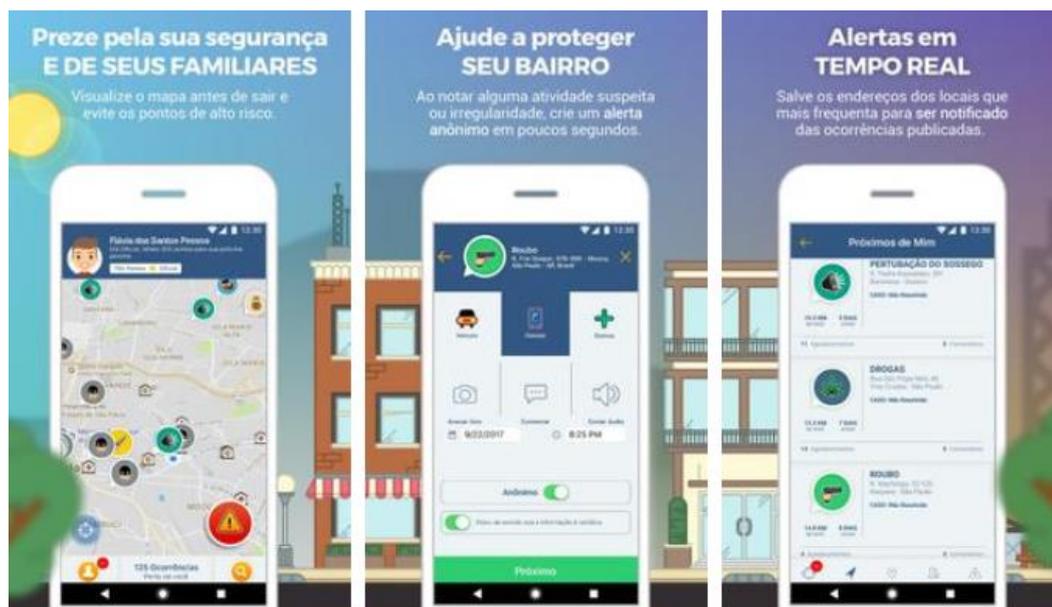


Figura 15 - APLICATIVO POLICIA POPULAR

FONTE: (CITZs Polícia Popular, 2018).

A Tabela 1, faz a comparação entre o *SafesON* e os aplicativos correlatos. Como é possível observar, a partir da Tabela 1, entre os aplicativos apresentados, alguns já possuem funcionalidades de compartilhamento de informações quanto a segurança. Contudo, nenhum deles possui integração entre estas funcionalidades, não existindo uma forma de o usuário gerar rotas entre dois pontos tendo em vista os acontecimentos reportados por outros usuários. Então, visando trazer praticidade aos usuários, o app *SafesON* integra as principais *features* apresentadas.

Tabela 1 - TABELA COMPARATIVO DE APLICATIVOS

FONTE: Os autores (2018).

<i>Features</i>	<i>Life 360</i>	Fui Assaltado Aqui	Polícia Popular	<i>SafesON</i>
Alertas de perigo no mapa		X	X	X
Criar grupos de conversa	X		X	X
Botão de pânico	X			X
Locais favoritos	X		X	
Compartilhar rotas				X
Compartilhar posicionamento fora de rota	X			

Como é possível observar a partir da tabela apresentada, entre os aplicativos apresentados, alguns já possuem funcionalidades de compartilhamento de informações quanto a segurança. Contudo, nenhum deles possui integração entre estas funcionalidades, não existindo uma forma de o usuário gerar rotas entre dois pontos tendo em vista os acontecimentos reportados por outros usuários. Então, visando trazer praticidade aos usuários, o app SafesON integra as principais features apresentadas.

### 3 MATERIAIS E MÉTODOS

#### 3.1 LEVANTAMENTO DE REQUISITOS

Buscando definir os principais requisitos que compõem a aplicação foi realizado um questionário com alunos da Universidade Federal do Paraná, utilizando a ferramenta de Formulários do *Google* para elaborar e disponibilizar as perguntas, além de ser possível obter uma relação em gráficos das respostas obtidas.

As perguntas do Apêndice A que auxiliaram em definir o perfil do entrevistado foram as perguntas 1, 2, 3 e 4. A pergunta nº 5 teve como objetivo perceber se existe um público para o aplicativo a ser desenvolvido, nas perguntas 6 e 7 foi possível definir alguns requisitos do programa, como: pânico, funcionalidade em que avisa os observadores da rota que o usuário está em perigo.

A partir do resultado obtido foi elaborado a lista de requisitos (seção 3.1.1), a qual define os requisitos funcionais e não funcionais do aplicativo, onde cada requisito está segmentado no padrão descrito a seguir:

- Referência: define um valor identificador para o requisito.
- Requisito: define o nome do requisito em si.
- Prioridade: define qual o impacto do requisito em relação a ideia central do projeto.
  - Prioridade 1: Baixo impacto.
  - Prioridade 2: Médio impacto.
  - Prioridade 3: Alto impacto.
- Descrição: definição descritiva por extenso do que se trata o requisito.
- Requisito de Origem: valor identificador do requisito de origem em relação ao requisito em questão.

### 3.1.1 ESPECIFICAÇÃO DE REQUISITOS

#### a) Requisitos Funcionais

Tabela 2 - LEVANTAMENTO DE REQUISITOS FUNCIONAIS

FONTE: Os autores (2018).

Referência	Requisito	Prioridade	Descrição	Requisito de Origem
REQF- 001	Gerenciar login	3	O aplicativo deve gerenciar acesso ao sistema para cada usuário, de forma que o usuário possa personalizar seus caminhos e contatos.	Não tem
REQF- 002	Realizar <i>login</i> por <i>Gmail</i>	2	Utilizar dados do Gmail para facilitar o cadastro de informações e para facilitar a gerenciamento de contas.	REQF- 001
REQF- 003	Visualizar o mapa	3	O usuário deve poder visualizar o mapa.	Não tem
REQF- 004	Visualizar sua posição	2	O cliente deve poder visualizar no mapa sua posição para traçar rotas e se localizar.	REQF- 003
REQF- 005	Reportar perigo	2	O usuário deve poder indicar no mapa crimes ou tipos de perigo.	REQF - 003
REQF- 006	Traçar trajeto	1	O sistema deve traçar três trajetos para o usuário decidir qual rota o atende melhor.	REQF - 003
REQF- 007	Estimar o tempo do trajeto	1	O sistema irá gerar a partir do trajeto traçado uma estimativa do tempo até o destino.	REQF - 006
REQF- 008	Escolher o método de transporte	1	O usuário deve conseguir escolher qual é o método de transporte a ser utilizado para criar a rota.	Não tem
REQF- 009	Botão de pânico	3	O usuário pode ativar um botão de pânico que manda um alerta após um tempo, prevendo possíveis ocorrências.	Não tem

Referência	Requisito	Prioridade	Descrição	Requisito de Origem
REQF- 010	Compartilhar o trajeto	3	O sistema deve permitir que seus usuários compartilhem seu trajeto escolhido com contatos ou grupos de contatos.	Não tem
REQF- 011	Criar grupo de contatos	3	O sistema deve permitir que seus usuários criem grupos específicos com seus contatos.	Não tem
REQF- 012	Adicionar contato por email	3	O sistema deve permitir que seus usuários adicionem seus contatos por meio de email de outros usuários cadastrados no sistema.	Não tem
REQF- 013	Salvar bateria do usuário	1	O sistema deve salvar o nível da bateria do usuário que está realizando o trajeto em tempo real.	Não tem
REQF- 014	Alertar desvio de trajeto para contatos.	2	O sistema deve alertar os contatos que estão acompanhando o trajeto quando o usuário desviar do caminho traçado e não confirmar o desvio.	Não tem
REQF- 015	Alertar usuário.	2	O sistema deve requisitar que o usuário confirme seu status.	REQF - 014

### B) Requisitos Não Funcionais

Tabela 3 - REQUISITOS NÃO FUNCIONAIS

FONTE: Os autores (2018).

Referência	Requisito	Prioridade	Descrição	Requisito de Origem
REQNF-001	Realizar Gerenciamento de eventos	3	Envio e resposta por <i>socket</i> entre o aplicativo e o <i>back-end</i> , utilizando <i>socket.io</i> ou <i>websocket</i> .	Não tem
REQNF-002	Realizar Conexão individual por rota	2	A cada solicitação de rota realizar uma conexão em <i>socket</i> individual para o gerenciamento da rota.	REQNF-001

Referência	Requisito	Prioridade	Descrição	Requisito de Origem
REQNF-003	Escrever os códigos em inglês	2	Fazer funções, variáveis e anotações em inglês.	Não tem
REQNF-004	Escrever os códigos seguindo a metodologia <i>Clean Code</i>	2	Não fugir da metodologia de maneira nenhuma.	REQNF-003
REQNF-005	Escrever os códigos usando o método <i>Reactive</i>	1	Usar reactive para estruturar as chamadas do aplicativo.	Não tem
REQNF-006	Gerenciar funcionalidades do mapa através da API <i>Maps</i>	3	Usar a API do <i>Maps</i> para o gerenciamento do mapa (inserção de pontos, cálculos específicos, etc).	Não tem
REQNF-007	Realizar testes unitários	1	Usar a api " <i>MochaJS</i> " ou outra semelhante para programar o TDD do <i>back-end</i> .	Não tem
REQNF-008	Gerenciar conexão com o banco de dados usando <i>MySQL</i>	3	Usar a API <i>mysql2</i> para se conectar com o banco de dados.	Não tem
REQNF-009	Desenvolver a aplicação servidor com <i>NodeJS</i>	3	Usar a versão 10.x.x do <i>NodeJS</i> para gerenciar as funcionalidades a nível de servidor.	Não tem
REQNF-010	Gerenciar erros	2	Tratar erros com mensagens amigáveis.	Não tem
REQNF-010	Gerenciar configurações	2	Facilitar a configuração do ambiente por meio de arquivos e variáveis generalizadas.	Não tem
REQNF-012	Gerenciar <i>threads</i>	2	Criar <i>threads</i> específicas para os eventos recebidos.	Não tem
REQNF-013	Realizar as <i>Activities</i> e <i>Views</i> do aplicativo	3	Realizar a conexão das <i>Activities</i> com as <i>Views</i> usando a API	Não tem

Referência	Requisito	Prioridade	Descrição	Requisito de Origem
			" <i>ButterKnife</i> " para facilitar o desenvolvimento.	
<b>REQNF-014</b>	Implementar o código do aplicativo com Java8	2	Utilizar expressões lambda para definir as funções.	Não tem
<b>REQNF-015</b>	Realizar a comunicação dos eventos por JSON	2	No aplicativo, utilizar a API <i>Gson</i> para converter os eventos em JSON.	<b>REQNF-001</b>
<b>REQNF-016</b>	Realizar uma camada de segurança com as conexões em <i>socket</i>	1	Fazer um identificador único por mensagem para validar o envio/recebimento de determinada mensagem.	<b>REQNF-001</b>
<b>REQNF-017</b>	Gravar <i>logs</i> de uso	2	Para cada mensagem enviada e para cada interação do usuário com o servidor, gravar em arquivos de <i>log</i> .	Não tem
<b>REQNF-018</b>	Realizar CRUD Orientado a eventos	2	Utilizar o <i>socket</i> para gerenciar o CRUD.	<b>REQNF-001</b>
<b>REQNF-019</b>	Realizar notificações	3	Utilizar <i>push notifications</i> para se comunicar com as pessoas que estão assistindo a rota e com o usuário.	Não tem

### 3.2 ANÁLISE DOS DADOS

Para o armazenamento e gerenciamento dos dados da aplicação foi utilizado o *MySQL Workbench*, por ser a versão aberta para comunidade, além de suprir todas as necessidades para o desenvolvimento do projeto, tanto para a análise quanto para o desenvolvimento do sistema. Para o projeto optou-se por um banco de dados relacional, no qual a equipe tem familiaridade e facilidade para fazer *queries* complexas, cujo modelo de entidade-relacionamento encontra-se no Apêndice B.

Para tanto, foi definido a construção de um único banco de dados, no âmbito do servidor, onde são mantidas as informações referentes ao aplicativo, julgado desnecessário um banco de dados para armazenamento local uma vez que todas as funcionalidades do aplicativo necessitam de conexão com o servidor.

Por meio de reuniões com a equipe foram determinadas as principais entidades e seus atributos para que mudanças necessárias sejam facilmente implementadas. A partir da análise do envio das informações das APIs, pode-se concluir de que maneira tais dados poderiam ser armazenados no banco, tornando possível determinar exatamente como seriam os atributos das entidades.

Já, referente as entidades e seus atributos, foram definidas as relações entre as entidades, dando sequência ao processo de normalização, consistindo em uma metodologia dividida em três etapas, com base no HEUSER (2001). A primeira etapa chamada de “primeira forma normal” que determina que toda relação deve ter uma chave primária e garantir que atributos sejam atômicos, indivisíveis, onde caso esses atributos sejam multivalorados devem ser discriminados e/ou separados em outra relação. A segunda etapa é chamada de “segunda forma normal”, onde são eliminadas informações duplicadas verificando se os atributos não são totalmente dependentes da chave primária. Por fim, a última etapa, chamada de “terceira forma normal”, objetiva eliminar dependências funcionais transitivas, onde os atributos que não são chaves devem ser mutuamente independentes.

Após este processo foi possível gerar o banco de dados em si, contendo todas as entidades e relações, utilizando o *software MySQL Workbench*. Conforme a evolução do desenvolvimento do aplicativo foi ocorrendo, foram feitas pequenas alterações na base.

Para melhor compreensão foi elaborado o dicionário de dados especificando as informações contidas nas tabelas, tais como descrição dos atributos, o que são e para que servem. Tais informações encontram-se no Apêndice B (DICIONÁRIO DE DADOS).

### 3.3 ANÁLISE DO APLICATIVO

#### 3.3.1 DIAGRAMA E ESPECIFICAÇÃO DE CASOS DE USO

O diagrama de caso de uso encontra-se no Apêndice B (DIAGRAMA DE CASOS DE USO), o qual descreve as funcionalidades propostas para ambos atores do sistema. Como pode ser observado no caso de uso, existem dez casos de uso e dois atores.

Os casos de uso são:

- UC001 - Logar no Aplicativo

- UC002 - Alertar
- UC003 - Reportar Acontecimento
- UC004 - Elaborar Rota
- UC005 - Selecionar Observadores
- UC006 - Gerenciar Amigos
- UC007 - Gerenciar Grupos
- UC008 - Visualizar Convites
- UC009 - Visualizar Rotas
- UC010 - Cancelar Rotas

Já os atores que interagem com o sistema são:

a) Usuário: para o cliente ter as funcionalidades de Usuário é necessário que ele realize o primeiro login no aplicativo. Após isso, permite que ele crie rotas com sua posição compartilhada com quem desejar, seja alertado, reporte ocorrências policiais, crie grupos de amigos, visualize rotas em que foi convidado a assistir, gerencie seus amigos.

b) Observador: esse papel assiste rotas que são compartilhadas pelo usuário.

A descrição dos casos de uso também está no Apêndice B (DESCRIÇÃO DOS CASOS DE USO), nesta seção é descrito todos os casos de uso de forma detalhada.

### 3.3.2 DIAGRAMA DE CLASSE

Os diagramas de classe modelados, os quais visam refletir melhor a implementação, seguem o modelo apresentado no livro UML, Guia do Usuário (BOOCH et al. 2005).

No Apêndice B (DIAGRAMA DE CLASSE - APLICATIVO) estão representadas as classes do aplicativo, onde as classes *PlaceInfo* e *ClusterMakers* são modelos auxiliares utilizados dentro da aplicação, que não tem relação com outras classes, somente para armazenamento em memória. Também no Apêndice B (DIAGRAMA DE CLASSE - SERVIDOR) estão representadas as classes do servidor e as relações entre elas. E, por fim, no Apêndice B (DIAGRAMA DE CLASSE - CONEXÃO COM SERVIDOR) também estão representadas as classes de conexão do aplicativo com o servidor via *WebSocket*.

### 3.3.3 DIAGRAMA DE SEQUÊNCIA

Os diagramas de sequência localizados no Apêndice B (DIAGRAMA DE SEQUÊNCIA) representam as interações entre os atores e as classes do sistema, detalhando os métodos utilizados nas funcionalidades desenvolvidas e as conexões entre as classes e as *activitys* do aplicativo.

As seguintes funcionalidades estão representadas nos diagramas de sequência:

- Alerta;
- Elaborar Rota;
- Gerenciar Amigos;
- Gerenciar Grupos;
- Logar no Aplicativo;
- Reportar Acontecimento;
- Selecionar Observadores;
- Visualizar Convites;
- Visualizar Rotas.

### 3.4 GERÊNCIA DO PROJETO

Por se tratar de um projeto complexo e com inovações, necessitando de diversas pesquisas pelos membros da equipe, foi fundamental definir uma metodologia de gerência de projeto que se adequasse a possíveis mudanças e incertezas durante o desenvolvimento do projeto. Sendo assim, a equipe optou por usar o *Adaptive Project Framework* (APF) ou também conhecido por *Adaptive Project Management* (APM).

O ciclo de vida do projeto pode ser visualizado na Figura 16, a qual foi elaborada com base no (LUCIDCHART CONTENT TEAM, 2018).



Figura 16 - CICLO DE VIDA DO PROJETO  
 FONTE: Os autores (2018).

O APF é uma metodologia idealizada por WYSOCKI (2010), que trabalha com fatores desconhecidos, inconstantes, sujeitos a mudanças, tendo como princípio

fundamental “aprender fazendo”, dando assim a possibilidade da equipe aprender e desenvolver ao mesmo tempo, reavaliando os resultados e tomando decisões ao longo do desenvolvimento.

Essa metodologia possui 5 principais fases no projeto, conforme Figura 16 (LUCIDCHART CONTENT TEAM, 2018):

- 1) Escopo do Projeto: fase inicial do projeto, momento em que são definidos os objetivos e requisitos do sistema, ou seja, as condições de satisfação do projeto, para que em cada etapa seja verificado se os resultados são satisfatórios, sendo assim os requisitos recebem valores de prioridade.
- 2) Plano de Ciclo: fase do projeto em que são definidas as tarefas individuais, tendo em vista a ordem que essas tarefas devem ser concluídas, para que outras funcionalidades não sejam prejudicadas.
- 3) Ciclo de Construção: fase de desenvolvimento do sistema rastreando possíveis problemas, agendando funcionalidades incompletas para o próximo ciclo, onde são feitos os trabalhos individuais definidos na etapa anterior.
- 4) *Checkpoint* do Cliente: fase de verificação, buscando efetuar a validação das etapas anteriores, ocorrendo uma reunião da equipe para serem apresentadas as funcionalidades e trabalhos do ciclo, além de definir as estratégias para a próxima fase ou ciclo e, caso seja necessário, agendar refatoração de códigos, melhorias de funcionalidades e mudanças nos documentos. No fluxo dessa metodologia, a próxima etapa do projeto pode ser um novo ciclo ou a própria fase final do desenvolvimento.
- 5) Revisão Final: última fase do projeto, onde a equipe verifica o sucesso do projeto de acordo com o que foi definido na primeira etapa definindo melhorias para possíveis processos futuros.

No início do projeto os cinco integrantes da equipe reuniram-se no dia 31 de julho do presente ano, para definir as principais ideias do projeto, levantar os requisitos de forma geral e priorizá-los, assim definindo os resultados esperados

A segunda fase do projeto foi repetida várias vezes, onde cada vez foi definido tarefas individuais do ciclo. Com essas tarefas definidas foi dado início a terceira fase do projeto, em que cada membro faz seus deveres referentes aos ciclos. Ao término desses deveres foi feita uma verificação com os outros integrantes, onde foi verificado as funcionalidades e trabalhos desenvolvidos no ciclo, e em alguns ciclos foi necessário algumas refatorações no código que entraram para o próximo ciclo como tarefa.

Na última etapa a equipe fez uma reunião para ver os resultados do desenvolvimento do projeto, comparando com os objetivos definidos na fase “Escopo do Projeto” e foram definidos os trabalhos futuros (descrição detalhada no capítulo 5.2).

Para a organização do processo de desenvolvimento foi utilizada a metodologia “Kanban”, com o auxílio da ferramenta “Trello” (Figura 17), em conjunto com os padrões de metodologia *Clean Code*, que consiste em organizar o código de uma forma que seja auto documentado para o entendimento dos desenvolvedores, mantendo um padrão de comunicação entre a equipe em relação às funcionalidades que estão sendo inseridas no projeto de maneira eficiente.

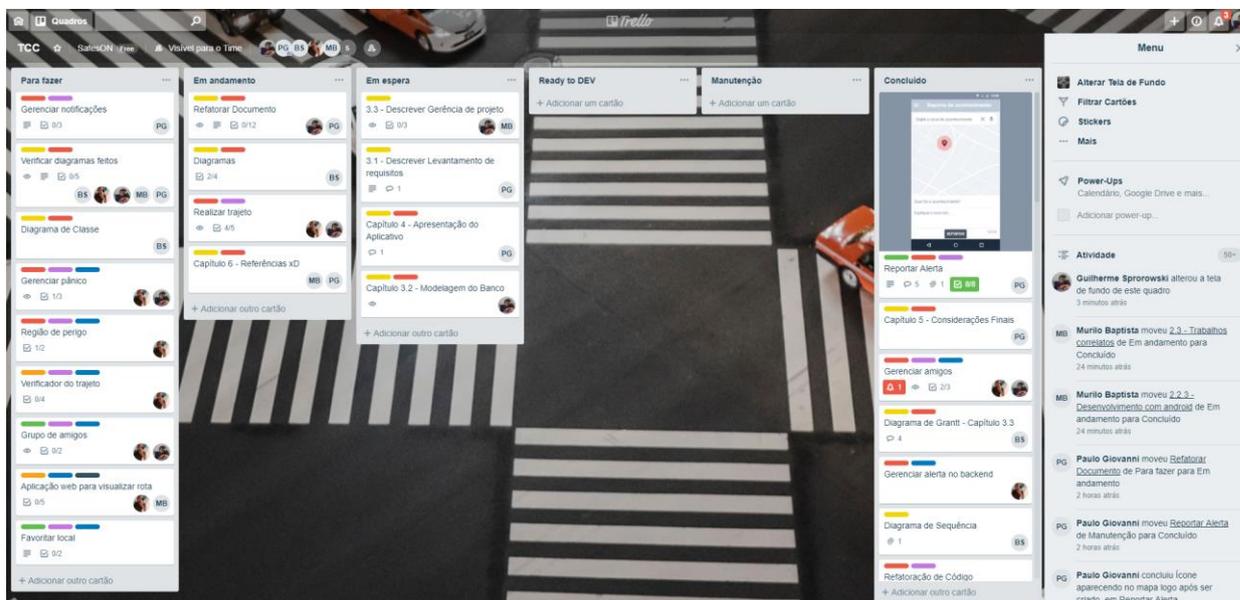


Figura 17 - KANBAN NO TRELLO

FONTE: Os autores (2018).

As atividades desenvolvidas no projeto foram distribuídas entre os integrantes da equipe da seguinte forma:

- Bruno Alves de Souza: Prototipação de Telas; Diagrama de Casos de Uso; Descrição dos Casos de uso; Diagrama de Classe; Diagrama de Sequência; Diagrama de Gantt; Elaboração do Capítulo 1 – Introdução.
- Elias Dal Ben Arruda: Desenvolvimento do *Server-side*; Integração e Estruturação do Aplicativo com o *Server-side*; Desenvolvimento da Funcionalidade de Visualizar Rotas; Modelagem de Dados; Elaboração da seção 3.5 – Tecnologias e Linguagens Aplicadas deste capítulo.
- Guilherme Ivama Sprorowski: Desenvolvimento da Funcionalidade de Elaborar Rotas; Gerenciar Amigos; Gerenciar Grupos; Visualizar Rotas; Modelagem de Dados; Diagrama de Classe; Interação entre as telas; Elaboração das seções 3.2 - Gerência do Projeto e 3.4 - Análise de Dados deste capítulo.
- Murilo Boeira Baptista: Revisão da Documentação; Elaboração do Capítulo 2 - Fundamentação Teórica.
- Paulo Giovanni Rupp da Luz: Desenvolvimento da Funcionalidade de Reportar Alerta; Refatoração do Código-fonte; Refatoração da Documentação; Elaboração dos Capítulos 1, 4 e 5.

O diagrama de *Gantt* no Apêndice B (DIAGRAMA DE GANTT) foi utilizado para ilustrar o tempo utilizado em cada etapa do desenvolvimento para facilitar a compreensão do andamento do projeto para os envolvidos.

### 3.5 TECNOLOGIAS E LINGUAGENS APLICADAS

Para o desenvolvimento do aplicativo foram utilizadas as seguintes tecnologias:

- Banco de Dados: MySQL. Atende todas as necessidades do projeto, além de ser uma ferramenta licenciada pela GNU *General Public License* (Licença Pública Geral GNU).
- Servidor *NodeJS*: Contém uma documentação *Open Source* muito abrangente em relação aos requisitos deste projeto.

- Conexão com *socket* (que provém uma conexão em tempo real com os usuários do aplicativo): *framework WS* integrado para o *NodeJS*;
- *API Maps* disponibilizada pela *Google* para a integração com o mapa. Por ser um aplicativo nativo em *Android*, tem um suporte facilitado para o uso dos módulos da API.
- *Framework Open Source MochaJS*: para os testes unitários, o qual atende à demanda dos testes de maneira intuitiva e simples de ser configurada, possibilitando, também, a escrita dos testes unitários de forma similar a escrita do aplicativo.
- *Visual Studio Code*: editor de texto principal do projeto. Também é *Open Source* e provém um suporte muito abrangente para *NodeJS*.
- Para o aplicativo, o desenvolvimento é nativo em *Android* e, por isso, será utilizada a própria IDE de desenvolvimento do *Android*, o *Android Studio*, que oferece a integração necessária para o projeto;
- Aplicação da metodologia *Reactive* e sua API para Java, em conjunto com as *features* de código do “Java 8”: para familiarizar a maneira da escrita do aplicativo com a do *back-end*, tornando assim, mais eficiente o desenvolvimento.
- Conexão com o *socket*: aplicação da API *OkHttp* de forma orientada a eventos com o *Reactive*, o qual é referente a uma programação que utiliza fluxos de dados assíncronos que podem ser transformados em um novo fluxo utilizando um conjunto de funções. Este fluxo de dados é emitido por um elemento que é recebido através de elementos de comunicação específicos da metodologia.
- Ambiente de testes: a ferramenta *Smart Websocket Client*, que possibilita testes com o *socket* de uma maneira simplificada; e os *smartphones* pessoais para debugar e rodar o aplicativo, conectados às redes locais de cada desenvolvedor.
- Controle de versão: foi utilizado o *bitbucket* pela condição de criar um projeto compartilhado entre os 5 integrantes da equipe, sendo privado entre os membros.

- Diagramas: foi utilizado a ferramenta *Astah* pela a familiaridade da equipe com a ferramenta e por ela ter uma versão gratuita para estudantes.
- Edição e Criação de ícones logos e *layouts*: utilizado o *Photoshop*, por ser uma ferramenta completa de edição de imagem.
- Prototipagem: uso da ferramenta *MarvelApp* que é uma ferramenta completa, com vários *layouts* e componentes com um *design* elegante e por ser gratuito.
- Linguagem do banco de dados: SQL (*Structured Query Language*)
- Linguagem de programação do aplicativo: Java
- Linguagem de programação da aplicação servidor: Javascript

### 3.6 CASOS SOBRE A IMPLEMENTAÇÃO DO SAFESON

Durante o desenvolvimento, tivemos alguns impasses que se provaram difíceis de resolver, porém, após a dedicação dos integrantes, foram possíveis de resolver e dar continuidade no projeto, entre essas dificuldades, são elas:

- Conexão com a rota: Não queríamos misturar os eventos que são recebidos das demais funcionalidades do aplicativo com os eventos da rota, para isso, tivemos que criar uma conexão dedicada para a rota, contendo nela, a opção de somente ouvir eventos relacionados à rota.
- Conectar e enviar mensagens para o servidor através do aplicativo: Após estudar as melhores formas de como fazer essa interação, e como a mensagem seria enviada, tivemos que criar um módulo de conexão específico com parâmetros específicos em cada mensagem, para poder efetuar uma conexão estruturada de forma que atenda a um CRUD e de forma que atenda a broadcasts dos eventos dos demais usuários.
- Reprodução da rota: Tivemos dificuldades em reproduzir a mesma rota criada por determinado usuário, para seus observadores, para isso, tivemos que salvar um "Polyline" (objeto que representa um "array" de coordenadas) no banco de dados junto com a rota, a partir desse dado, foi possível interpretá-lo ao receber a mensagem de início de rota, desenhando assim, a rota certa no mapa.

- Gerar deploy do servidor: Para realizar o deploy da aplicação servidor em algum endereço da internet, decidimos hospedar no sistema de Cloud da Google que oferece uma maneira simples de subir máquinas virtuais para a nossa demanda, o problema, foi realizar a comunicação com o protocolo do socket na máquina virtual, para isso, foi feito um reverse proxy usando a ferramenta NGINX que permitiu, após as configurações necessárias, o redirecionamento de uma rota servida pelo protocolo HTTP para o protocolo WSS.

Além disso, foi elaborado a tabela de bugs no apêndice B para descrever a solução de alguns bugs encontrados no desenvolvimento.

## **4 APRESENTAÇÃO DO APLICATIVO**

### **4.1 O APLICATIVO SAFESON**

O aplicativo *SafesON* busca fornecer simplicidade e facilidade ao usuário. A primeira tela com que o usuário se depara é a de *login*, e após o mesmo ser efetuado o usuário é levado à tela principal. Buscando facilitar a navegação, a aplicação disponibiliza suas principais funcionalidades já nesta tela, a qual possui um mapa clicável e uma barra de busca para adicionar um marcador em algum local no mapa.

Então, com o marcador adicionado é possível gerar uma rota até este local, selecionando meio de transporte e quais amigos receberão o convite para acompanhar o percurso em tempo real.

Já para adicionar um acontecimento no mapa, selecione um local, clique no ícone de alerta, preencha as informações e salve.

Para adicionar um amigo, abra o menu lateral a partir da tela principal e selecione a opção “Amigos”. Na próxima tela selecione o botão de adicionar, representado pelo ícone “+”, insira o e-mail e clique em “Adicionar”. E, para criar um grupo de amigos, abra o menu lateral a partir da tela principal e selecione a opção “Grupos”. Na próxima tela insira o nome e os contatos que participarão do grupo, então clique em “Criar”.

Para acessar os convites, abra o menu lateral a partir da tela principal e selecione a opção “Convites”. Na próxima tela serão exibidas as notificações referentes a convites de rota.

Para aceitar/recusar um convite, acesse a tela de convites e na notificação desejada pressione o botão “Aceitar” ou “Recusar”.

Para enviar um alerta de pânico, possuindo uma rota atrelada ao seu usuário, pressione 4 vezes o botão de bloquear do smartphone, tal alerta será enviado a todos os observadores.

#### 4.2 TELAS DO APLICATIVO MÓVEL

A Figura 18 apresenta a Tela Inicial, local onde é solicitado que o usuário efetue o login via “Google”.

A descrição do caso de uso desta tela está identificado como UC001 no Apêndice B (ESPECIFICAÇÃO DE CASOS DE USO).

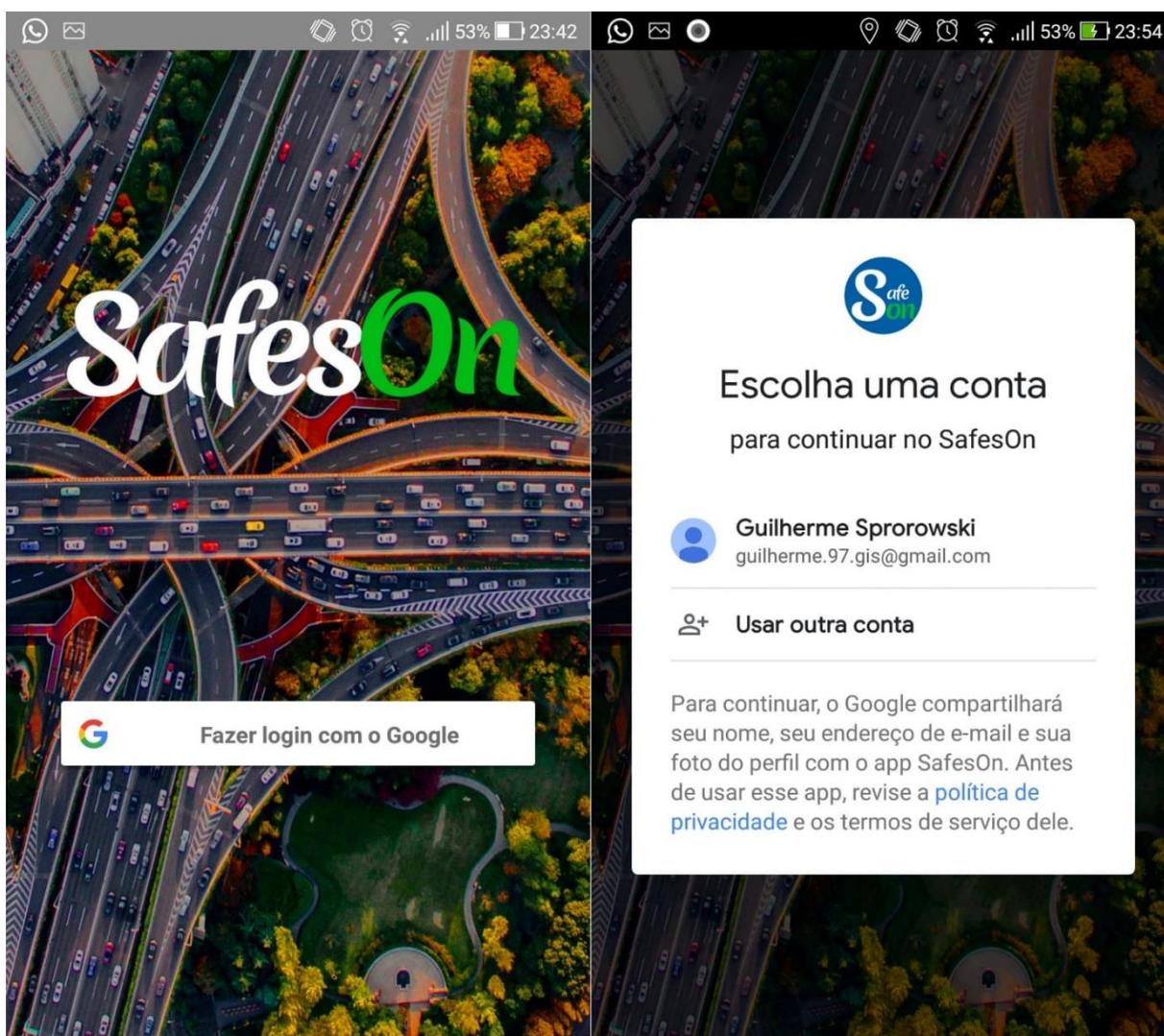


Figura 18 - TELA DE LOGIN

FONTE: Os autores (2018).

A Figura 19 apresenta a Tela Principal, que exibe o mapa com os marcadores de acontecimentos, além da localização do usuário e de seus amigos com rotas compartilhadas.

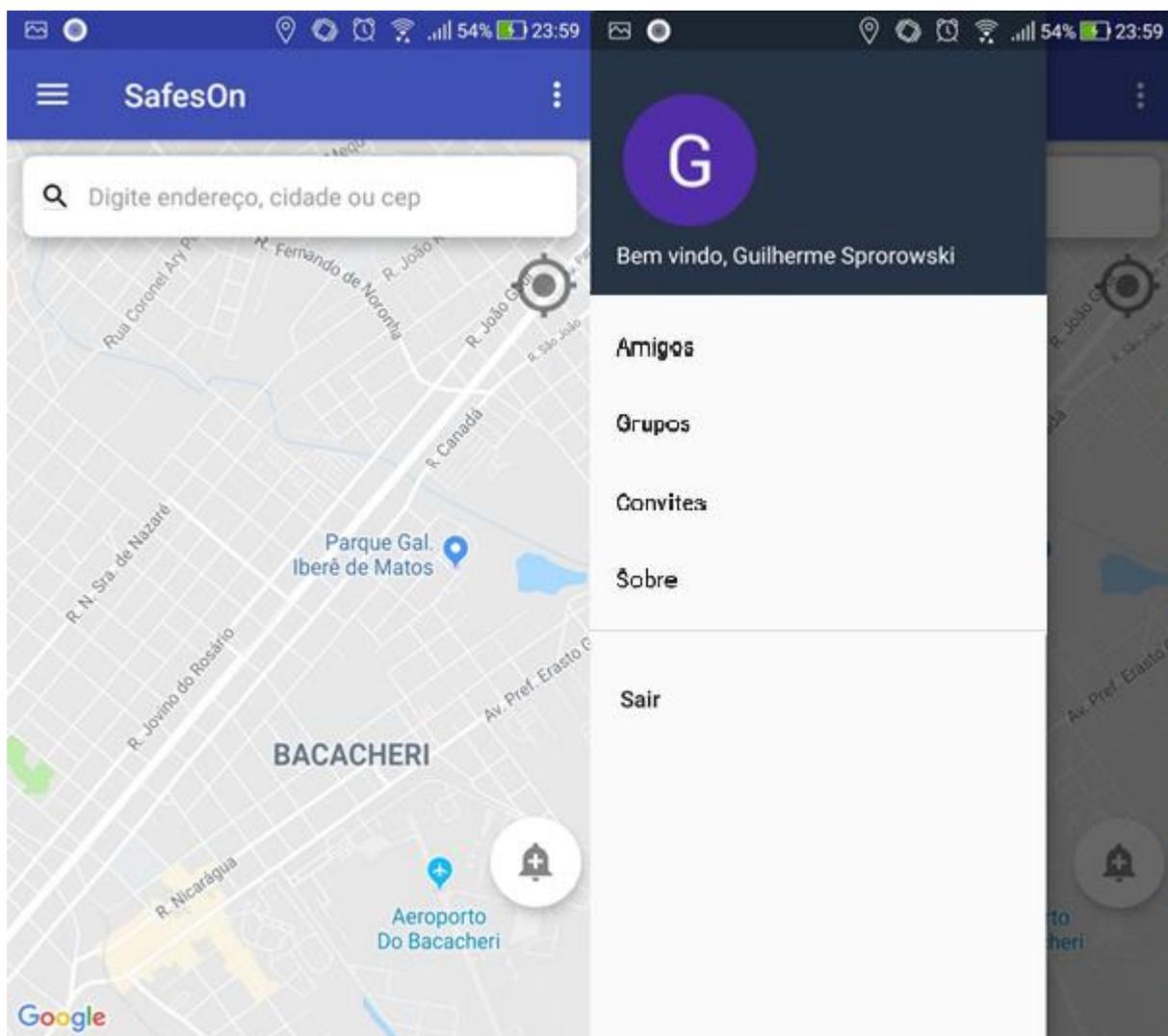


Figura 19 - TELA PRINCIPAL

FONTE: Os autores (2018).

Através do mapa presente na tela principal o usuário pode criar uma rota a partir de sua localização até um ponto desejado, o qual pode ser criado digitando um endereço na barra de busca ou simplesmente clicando no mapa em si. Em seguida o usuário deve pressionar o botão "CRIAR ROTA" (Figura 20).

A descrição do caso de uso desta da tela Gerar Rota está identificado como UC004 no Apêndice B (ESPECIFICAÇÃO DE CASOS DE USO).

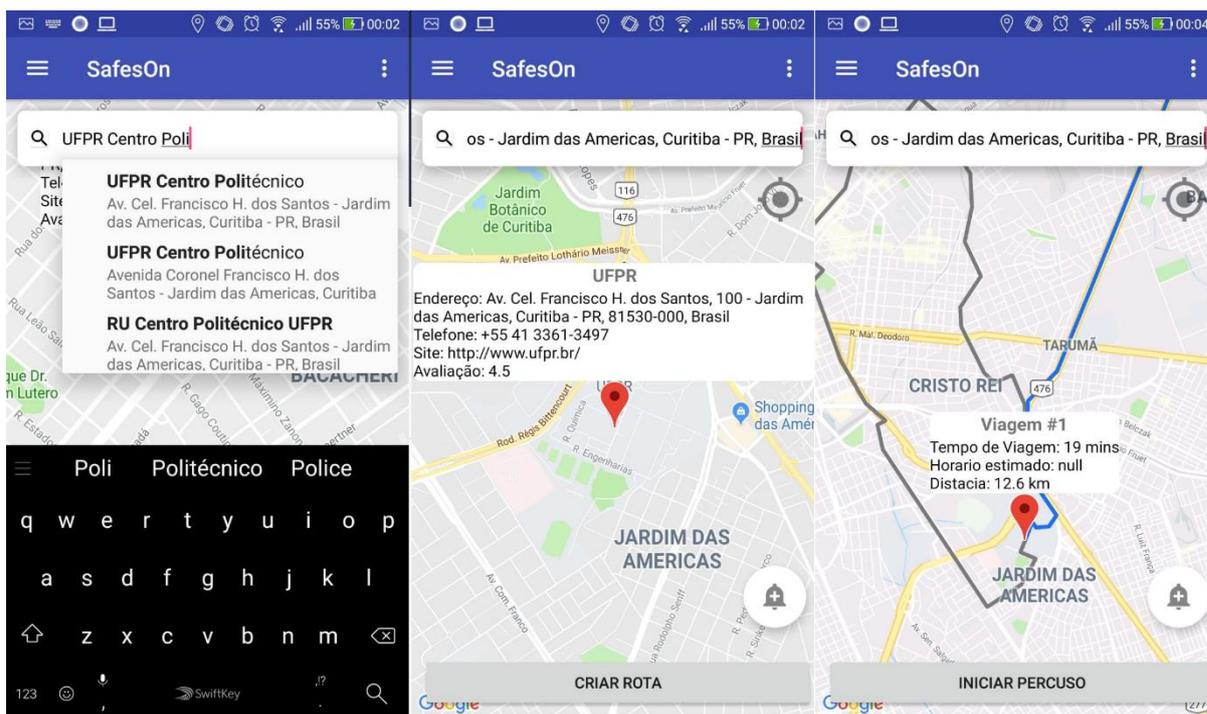


Figura 20 - GERAR ROTA  
 FONTE: Os autores (2018).



Figura 21 - SELECIONAR MEIO DE TRANSPORTE  
 FONTE: Os autores (2018).

A tela selecionar Meio de Transporte (Figura 21) é exibida após o usuário ter pressionado o botão “CRIAR ROTA”. O usuário então deve escolher o meio de transporte entre os 4 apresentados, possibilitando que a rota seja gerada.

A descrição do caso de uso desta tela está identificado como UC004 no Apêndice B (ESPECIFICAÇÃO DE CASOS DE USO).

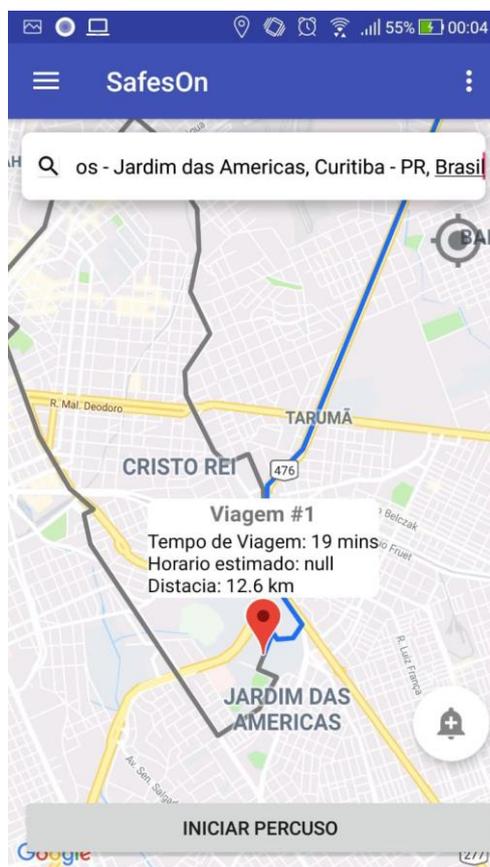


Figura 22 - INICIAR PERCURSO

FONTE: Os autores (2018).

Tendo o usuário selecionado o meio de transporte, ele deve escolher entre as diferentes rotas apresentadas, e em seguida, pressionar o botão “INICIAR PERCURSO” (Figura 22).

A descrição do caso de uso desta tela está identificado como UC004 no Apêndice B (ESPECIFICAÇÃO DE CASOS DE USO).

A tela Selecionar Observadores (Figura 23) é exibida após o usuário pressionar o botão de “INICIAR PERCURSO”. Após esta ação o usuário deve selecionar os amigos que receberão os convites para terem acesso ao percurso em questão.

A descrição do caso de uso desta tela se está identificado como UC004 no Apêndice B (ESPECIFICAÇÃO DE CASOS DE USO).

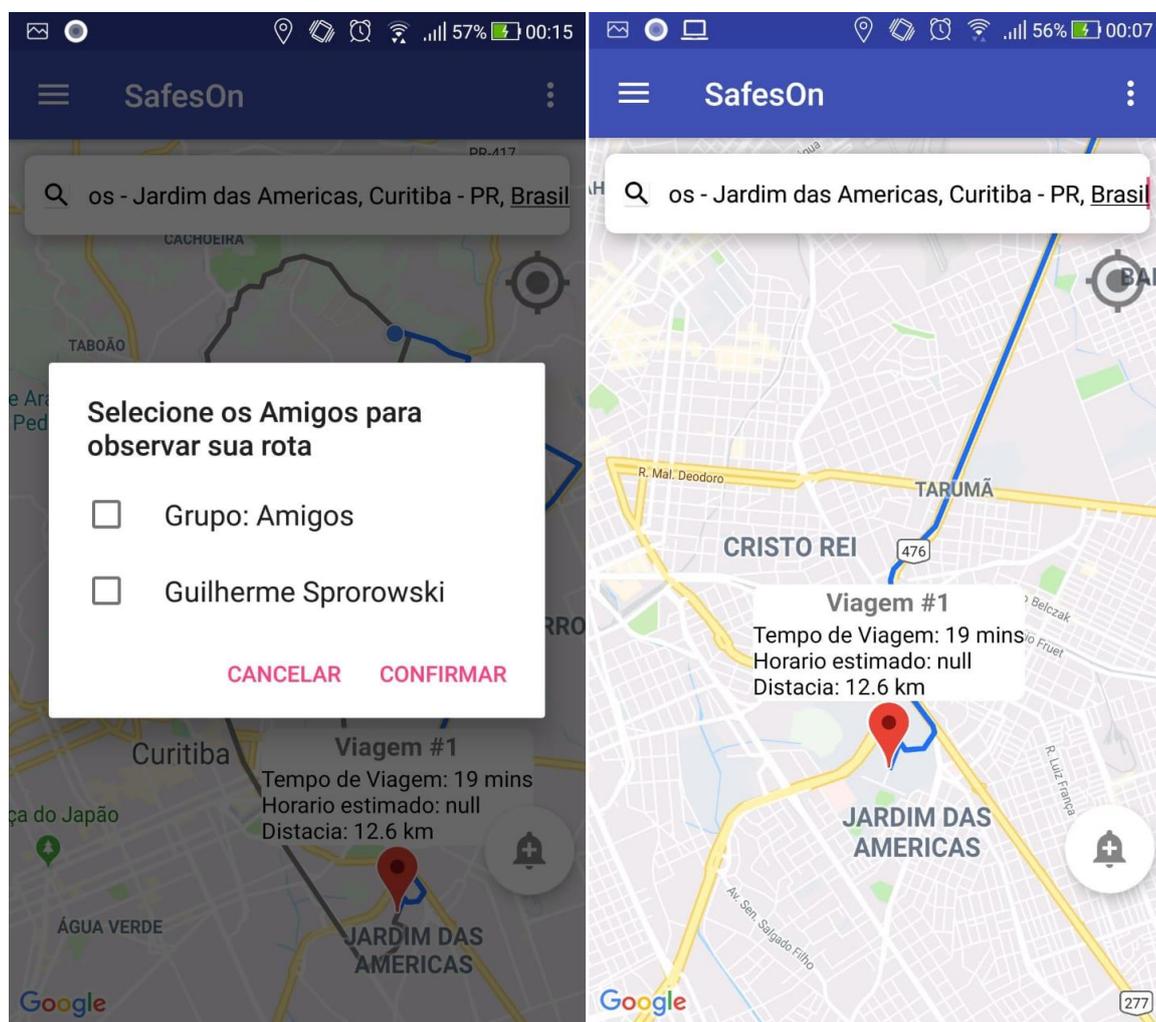


Figura 23 - SELECIONAR OBSERVADORES

FONTE: Os autores (2018).

Quando o usuário clica no ícone de alerta, na tela principal, a tela Adicionar Acontecimento (Figura 24) é exibida, permitindo assim ser adicionada uma ocorrência a ser exibida para todos no mapa.

A descrição do caso de uso desta tela está identificada como UC003 no Apêndice B (ESPECIFICAÇÃO DE CASOS DE USO).

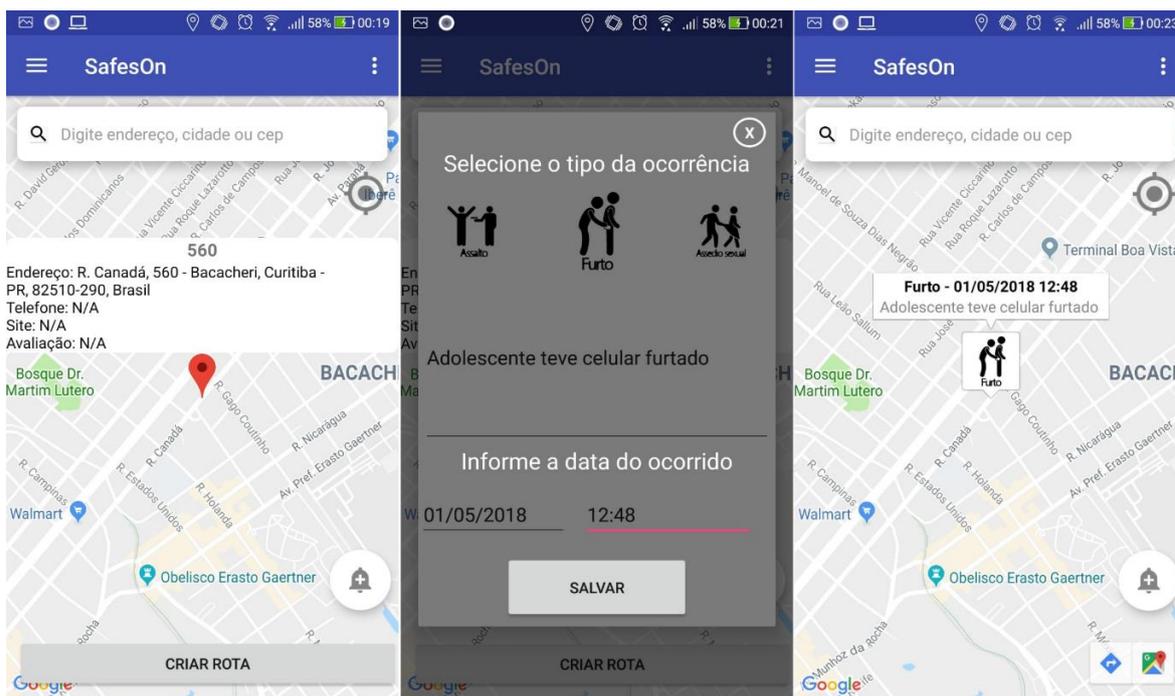


Figura 24 - ADICIONAR ACONTECIMENTO

FONTE: Os autores (2018).

Ao enviar um alerta de pânico (Figura 25), a tela Aviso de Alerta é exibida ao usuário a enviar o alerta, possibilitando-o de cancelar o mesmo.

A descrição do caso de uso desta tela está identificada como UC002 no Apêndice B (ESPECIFICAÇÃO DE CASO DE USO)

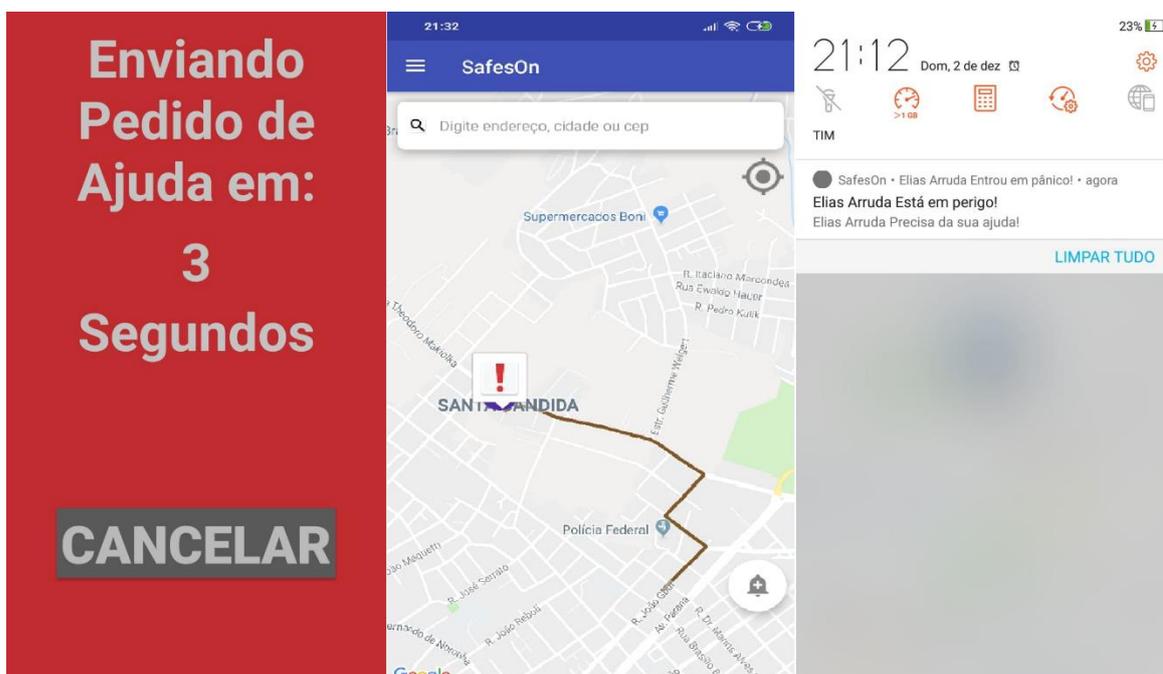


Figura 25 - ENVIAR PEDIDO DE AJUDA

FONTE: Os autores (2018).

A tela Gerenciar Amigos (Figura 26) permite ao usuário ver, remover ou prosseguir no fluxo de adição de amigos. Para remover um amigo basta clicar no ícone “REMOVER”. Caso o usuário deseje adicionar um novo amigo é necessário clicar no ícone “ADICIONAR”, representado pelo sinal de “+”; inserir o e-mail do usuário e pressionar “ADICIONAR”.

A descrição do caso de uso desta tela está identificado como UC006 no Apêndice B (ESPECIFICAÇÃO DE CASOS DE USO).

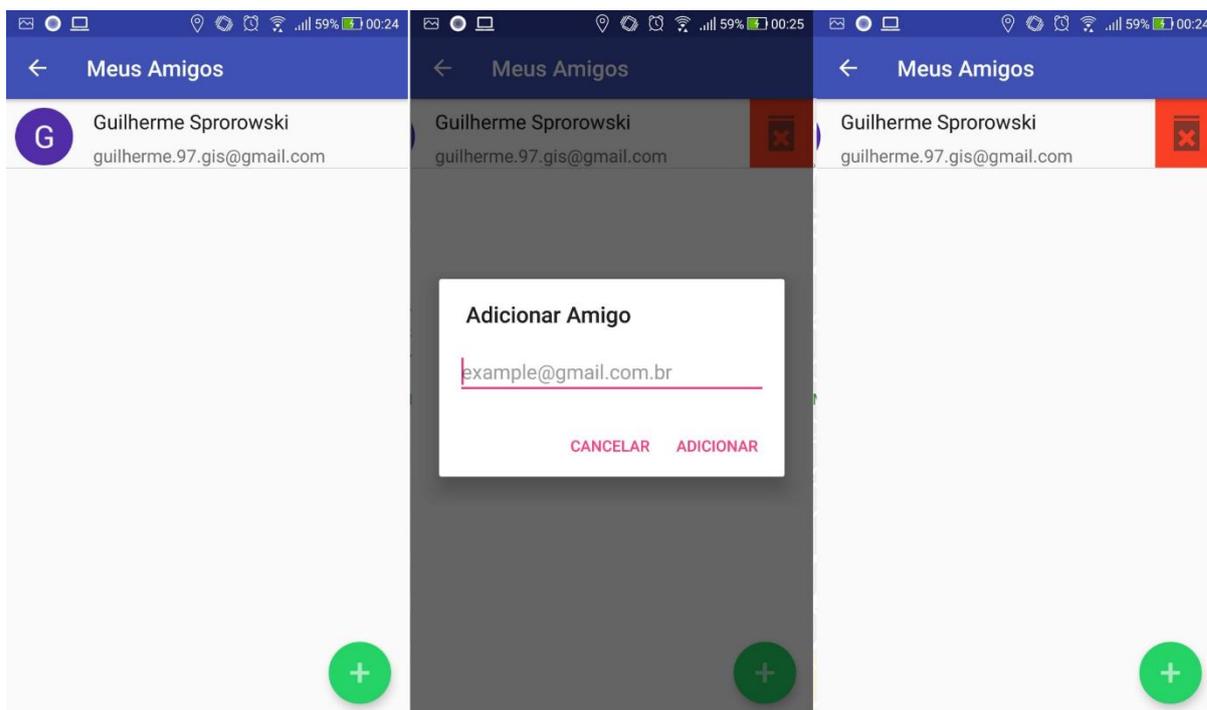


Figura 26 - GERENCIAR AMIGOS

FONTE: Os autores (2018).

A tela Gerenciar Grupos (Figura 27) permite o usuário criar, excluir ou visualizar grupos de amigos. Caso o usuário deseje criar um grupo ele deve inserir o nome e os contatos desejados, para finalizar deve pressionar o botão “ADICIONAR”. O usuário também pode visualizar os integrantes de algum grupo, clicando no item da lista referente ao grupo, e para remover, o usuário arrasta o item para o lado esquerdo, liberando um botão para deletar o mesmo.

A descrição do caso de uso desta tela está identificado como UC007 no Apêndice B (ESPECIFICAÇÃO DE CASOS DE USO).

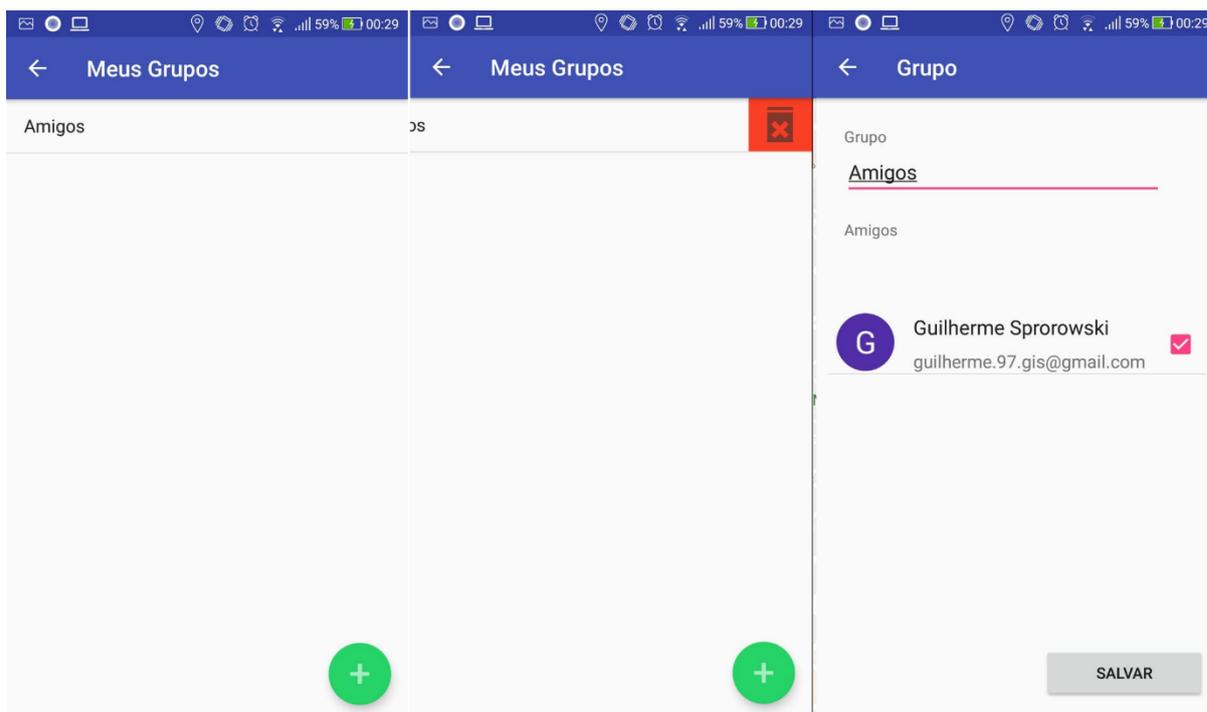


Figura 27 - GERENCIAR GRUPOS

FONTE: Os autores (2018).

A tela Gerenciar Convites (Figura 27) permite ao usuário visualizar e gerenciar os seus convites de rota, sendo que o usuário tem as opções de aceitar ou recusar. A descrição do caso de uso desta tela está identificada como UC008 no Apêndice B (ESPECIFICAÇÃO DE CASOS DE USO).



Figura 28 - GERENCIAR CONVITES

FONTE: Os autores (2018).

## 5 CONSIDERAÇÕES FINAIS

### 5.1 CONCLUSÃO DO PROJETO

A partir da ideia apresentada neste projeto, a aplicação desenvolvida atende os objetivos descritos anteriormente, nas seções referentes a objetivo geral e específicos. E tendo como base a necessidade de auxílio à segurança pública, identificada e analisada anteriormente neste projeto, as funcionalidades desenvolvidas compõem uma ferramenta útil para pessoas em traslado.

Tendo em vista os objetivos específicos apresentados (seção 1.3.2) a aplicação desenvolvida atinge o que foi descrito, desde a parte de análise, onde foi identificada uma demanda por mais segurança pública, até às funcionalidades desenvolvidas, as quais forneceram aos usuários um meio para compartilhamento de informação em tempo real, através das interações de demarcação de áreas de risco no mapa, alerta de pânico para situações de emergência e compartilhamento de rota para amigos ou grupos de amigos.

Com a possibilidade de compartilhamento de acontecimentos em tempo real no mapa, é fornecido mais controle ao usuário na hora de gerar um trajeto, podendo evitar zonas de perigo e, conseqüentemente, seguir por rotas mais seguras. O compartilhamento de rotas e posição permitem que os usuários estejam mais seguros ao transitarem, além de, em conjunto com as funcionalidades de alerta, serem facilitadas ações para prestar auxílio ou socorro às pessoas sendo monitoradas pelo app.

Os requisitos apresentados e desenvolvidos surgem, principalmente, a partir de uma consulta com universitários, como descrito no capítulo introdutório. Além disto, o questionário apresentado, considerando as perguntas de 2 a 7, permitiu aos desenvolvedores identificarem qual o real impacto que o projeto teria na sociedade, contudo, a pergunta quanto ao gênero da pessoa a responder, não gerou nenhum resultado significativo para a análise do projeto. Identificadas as principais funcionalidades e necessidades, o desenvolvimento do projeto e sua versão final atendem o que foi proposto nesta documentação.

Considerando os requisitos funcionais apresentados (seção 3.1.1), a aplicação desenvolvida atinge o que foi proposto. O desenvolvimento das funcionalidades seguiu um cronograma tomando como base a prioridade que cada

requisito possui no sistema, possibilitando aos desenvolvedores focarem no que teria mais impacto com relação a ideia principal do sistema.

Da mesma forma, considerando os requisitos não funcionais apresentados, não conseguimos seguir a risca a metodologia do Clean code, não fizemos de maneira efetiva os testes unitários, não foi feito o gerenciamento de erros e não foi usado push notification para alertar das demais notificações do sistema, com excessão desses itens citados acima, foi possível cumprir com todos os outros requisitos funcionais.

Dentro do panorama do projeto proposto, a modelagem de dados foi satisfatória, atendendo as necessidades de desenvolvimento, possibilitando a criação das funcionalidades, com a aplicação sendo implementada a partir da análise feita inicialmente.

Buscando organizar da melhor maneira possível dentro do escopo do projeto, a metodologia de gerência escolhida possibilitou manter flexível o desenvolvimento, em face das possibilidades de existirem incertezas na implementação. Contudo, foi possível prosseguir dentro do planejado, de maneira organizada e sincronizada, e apesar das dificuldades e imprevistos, o projeto foi concluído com sucesso.

As ferramentas utilizadas para o desenvolvimento atenderam às expectativas dos desenvolvedores, possibilitando o desenvolvimento de maneira ágil e limpa, principalmente no âmbito do servidor. Além de permitir refatorações futuras, ou mesmo a implementação da aplicação em diferentes plataformas e tecnologias.

No âmbito da aplicação, o desenvolvimento para *Android*, utilizando a plataforma nativa, possibilitou com que os desenvolvedores contassem com maior suporte para a utilização da API do *Google Maps*, tendo em vista a existência de um kit de desenvolvimento apropriado para o *Android*. Contudo, a necessidade da utilização do pacote SDK da linguagem Java, trouxe grande complexidade para o desenvolvimento, porém a plataforma forneceu as ferramentas necessárias para a implementação.

## 5.2 TRABALHOS FUTUROS

Visando novas funcionalidades futuras, ainda buscando oferecer mais segurança aos usuários, a principal *feature* a ser desenvolvida trata-se da criação de rotas levando em consideração os marcadores de acontecimentos no mapa, fornecendo ao usuário a opção de um trajeto mais seguro ainda, gerado automaticamente pela aplicação. Tal funcionalidade deverá contar com o suporte da

*Google* futuramente (SMITH, 2018), haja vista o suporte da empresa de tal funcionalidade para o aplicativo *Waze*.

Outra funcionalidade a ser implementada trata-se do desenvolvimento de uma parte *Web*, possibilitando que usuários que não possuem a aplicação em um dispositivo móvel tenha acesso, a partir do compartilhamento de um *link* gerado pelo aplicativo, ao mapa e ao usuário a compartilhar a rota, de maneira direta e eficaz. Ainda dentro desta ideia, existe a possibilidade deste *link* ser compartilhado e fornecer acesso direto ao app, caso o usuário já o possua.

Levando em conta as áreas de perigo demarcadas no mapa pelo usuário, outra funcionalidade a ser implementada é a criação de círculos no mapa com diferentes tons de vermelho, com isso, considerando os alertas reportados no mapa, seria determinado um diâmetro onde esse círculo seria elaborado, caracterizando um círculo de tom mais escuro uma região de alto perigo e de tom mais claro uma região de menor perigo.

Outra funcionalidade a ser implementada no sistema é a possibilidade de troca de mensagens entre o usuário que está em trajeto e seus observadores, evitando falsos alertas e podendo esclarecer situações anormais vindas a ocorrer com o usuário.

Considerando a função social do projeto proposto, outra funcionalidade que agregaria para o escopo do projeto é a elaboração de um sistema de relatórios. Com as informações contidas na aplicação, seria útil para a segurança pública uma análise mais aprofundada dos dados existentes a partir das interações do usuário, além da possibilidade de utilização dos relatórios criados em prol de novas funcionalidades e requisitos, ampliando a quantidade de alternativas para tornar mais eficiente a segurança pública para os usuários.

Por fim, outra questão que também agregaria para o entendimento da usabilidade da aplicação, seria a utilização da mesma por parte de diversos usuários no dia-a-dia, possibilitando a identificação de como as funcionalidades propostas interagem em situações reais, buscando coletar feedback a respeito de possíveis melhorias e ajustes.

## REFERÊNCIAS BIBLIOGRÁFICAS

APKPURE. **Fui Assaltado Aqui**. Disponível em: < <https://apkpure.com/br/fui-assaltado-aqui/com.fuiassaltadoaqui.app>> Acesso em 21 nov. 2018

BOOCH, G; RUMBAUGH, J e JACOBSON, I: **UML, Guia do Usuário**: tradução; Fábio Freitas da Silva, Rio de Janeiro, Campus ,2000.

CHRIS, Smith. **Google Maps is about to get Waze's best feature**. 29 de jun. de 2018. Disponível em: <<https://bgr.com/2018/06/29/google-maps-vs-waze-incident-reports-user-interaction/>> Acesso em 21 de nov. 2018.

CIDRAL, Beline. **Afinal, o que é Android?**. 05 de jan. de 2011. Disponível em: <<https://www.techtudo.com.br/artigos/noticia/2011/01/afinal-o-que-e-android.html>> Acesso em 26 nov. 2018.

DEVELOPERS android. **Activity**. Disponível em: <<https://developer.android.com/guide/components/activities>> Acesso em 21 nov. 2018.

DEVELOPER ANDROID. **CRIAÇÃO DE UMA GAVETA DE NAVEGAÇÃO**. Disponível em: <<https://developer.android.com/training/implementing-navigation/nav-drawer>> Acesso em: 21 nov. 2018.

DEVELOPERS android. **Platform Architecture**. Disponível em: <<https://developer.android.com/guide/platform/?hl=en-us>> Acesso em 20 nov. 2018.

DEVELOPERS android. **Services**. Disponível em: <<https://developer.android.com/guide/components/services>> Acesso em 21 nov. 2018.

DEVELOPERS android. **TelephonyManager**. Disponível em: <<https://developer.android.com/reference/android/telephony/TelephonyManager>> Acesso em 20 nov. 2018.

DEVELOPERS android. **Versões**. Disponível em: <<https://developer.android.com/about/versions/pie/?hl=pt-br>> Acesso em 18 nov. 2018.

G1. **Brasil registra mais de 26 mil assassinatos no 1º semestre de 2018**. Disponível em: <<https://g1.globo.com/monitor-da-violencia/noticia/2018/08/28/brasil-registra-mais-de-26-mil-assassinatos-no-1o-semester-de-2018.ghtml>> Acesso em 29 de set. de 2018.

G1. **Monitor da Violência**. Disponível em: <<https://g1.globo.com/monitor-da-violencia/>> Acesso em 29 de set. de 2018.

GLOBALSTATS. **Statcounter**. Disponível em: <<http://gs.statcounter.com/os-market-share#yearly-2009-2018>> Acesso em 21 nov. 2018.

GOOGLE PLAY. **CITZs Polícia Popular**. Disponível em: < [https://play.google.com/store/apps/details?id=com.stark.PoliciaPopular&hl=pt\\_BR](https://play.google.com/store/apps/details?id=com.stark.PoliciaPopular&hl=pt_BR) > Acesso em 21 nov. 2018.

HEUSER, Carlos Alberto. **Projeto de Banco de Dados**. Editora Sagra Luzzato, 4ª edição, 2001.

KANJILAL, Joydip. **Understanding the Android Platform Architecture**. 4 mar. 2016. Disponível em: <<https://www.developer.com/ws/android/understanding-the-android-platform-architecture.html>> Acesso em 19 nov. 2018

LECHETA, Ricardo R. **Google Android - 4ª Edição: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. São Paulo: Novatec Editora, 2015.

LUCIDCHART CONTENT TEAM. **What Is Adaptive Project Framework?** Disponível em: < <https://www.lucidchart.com/blog/what-is-adaptive-project-framework>> Acesso em 21 de nov. 2018.

MANCHANDA, Amit. **Why Kotlin is the Future of Android App Development**. 13 set. 2018. Disponível em: <<https://www.netsolutions.com/insights/why-kotlin-is-the-future-of-android-app-development/>> Acesso em 26 nov. 2018.

MOTTA, Douglas. **Activity: O que é, como é iniciada e ciclo de vida básico**. 18 de maio de 2018. Disponível em: <<https://www.concrete.com.br/2018/05/18/activity-o-que-e-como-e-iniciada-e-ciclo-de-vida-basico/>> Acesso em 21 de nov. de 2018.

PONDER, George. **Life360 for Windows Phone, an easy way to keep track of your Family**. 2014. Disponível em: <<https://www.windowscentral.com/life360-windows-phone-easy-way-keep-track-your-family>>. Acesso em: 21 nov. 2018.

SOBRINHO, Cristiana. **Android Studio (Conceitos Básicos)**. 20 de set. de 2018. Disponível em: <<http://knoow.net/ciencinformtelec/informatica/conceitos-basicos-android-studio/>> Acesso em 21 de nov. 2018.

SOURCE android. **The Android Source Code**. Disponível em: <<http://source.android.com/source/index.html>> Acesso em 21 de nov. de 2018

STATCOUNTER. **Operating System Market Share Worldwide**. Disponível em: <<http://gs.statcounter.com/os-market-share#yearly-2009-2018>> Acesso em 10 de dez. 2018.

STUDYTONIGHT. **Android Architecture - Software Stack of Android**. Disponível em: <<https://www.studytonight.com/android/android-architecture>> Acesso em 20 nov. 2018.

TECHNOPEDIA. **Mobile Application (Mobile App)**. Disponível em: <<https://www.techopedia.com/definition/2953/mobile-application-mobile-app>>. Acesso em 26 nov. 2018.

VEJA ABRIL. **Uber vai mostrar no mapa a cor do carro solicitado**. Disponível em: < <https://veja.abril.com.br/economia/uber-vai-mostrar-a-cor-do-carro-chamado-no-mapa>> Acesso em 21 nov. 2018.

WYSOCKI, Robert K. **Adaptive Project Framework: Managing Complexity in the Face of Uncertainty**. Upper Saddle River, NJ: Addison-Wesley, 2010.

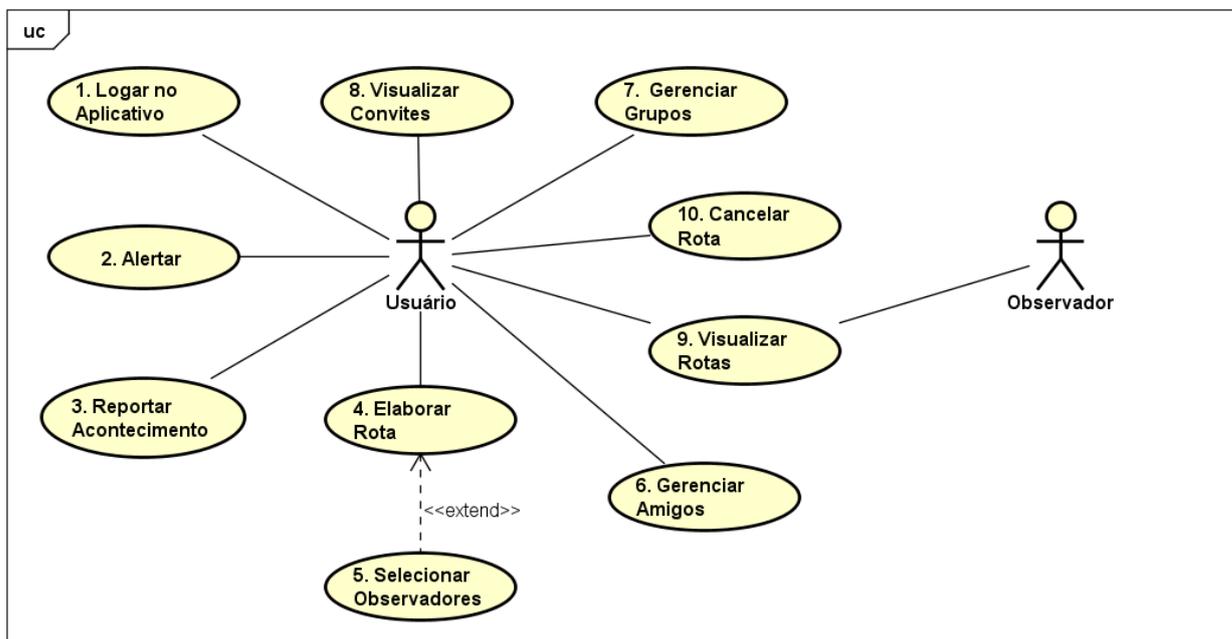
## APÊNDICE A – QUESTIONÁRIO APLICADO NA PESQUISA COM OS UNIVERSITÁRIOS

1. Qual é seu gênero?
  - a. Masculino
  - b. Feminino
  - c. Transexual
  - d. Prefiro não dizer
2. Qual o meio de locomoção que você mais utiliza?
  - ( ) Transporte público.
  - ( ) Carro ou moto.
  - ( ) Carona.
  - ( ) A pé.
  - ( ) Uber ou Táxi.
3. Você se sente seguro(a) em seu percurso de ida e volta da universidade?
  - a. Sim
  - b. Sim, na maioria das vezes
  - c. Não.
4. No percurso de ida ou volta da universidade você já foi assaltado(a) ou sofreu alguma abordagem intimidadora?
  - a. Sim
  - b. Não
5. Você se sentiria mais seguro(a) com algum conhecido monitorando seu trajeto de ida e volta da universidade através de um aplicativo?
  - a. Sim
  - b. Não
6. Você faria um caminho mais longe para o seu destino caso fosse mais seguro?
  - a. Sim
  - b. Não
7. Indique informações e/ou recursos que você considera interessante ter disponível em um aplicativo que visa auxiliar na sua segurança física em momentos de deslocamento, como por exemplo, entre a universidade e a sua casa.

R: (Resposta aberta)

## APÊNDICE B – DOCUMENTAÇÃO TÉCNICA

### 1 DIAGRAMA DE CASOS DE USO



powered by Astah

### 2 ESPECIFICAÇÃO DE CASO DE USO

#### 2.1.1 CASO DE USO: UC001 - LOGAR NO APLICATIVO

##### Descrição:

Este caso de uso é iniciado pelo usuário.

O usuário deve se autenticar no sistema via login do “Google”.

##### DATAVIEW



##### DATAVIEW 1 - LOGAR

Ator Primário:

Todos.

Pré-Condição: Possuir uma conta do Google.

Pós-Condição: Acessar o sistema com o seu respectivo email, gerando o seu perfil.

Fluxo de Eventos Principal:

- 1) O ator clica no botão de logar com a conta do Google.
- 2) O ator informa os seus dados de login da conta do Google.
- 3) O sistema pede permissão para o ator vincular a sua conta do Google com o aplicativo.
- 4) O ator confirma a permissão.
- 5) O sistema redireciona o ator para a tela inicial do aplicativo.
- 6) O caso de uso é finalizado.

Fluxo Alternativo:

Nenhum.

Fluxo de Exceção:

E1: Dados de login incorretos (caso o usuário não tenha uma conta da google vinculada em seu aparelho):

- E1.1) O sistema verifica os dados de login.
- E1.2) O sistema retorna uma mensagem de erro.
- E1.3) O sistema redireciona o ator para preencher o campo novamente.

E2: Ator não cede permissão para o sistema vincular seus dados do Google:

- E2.1) O sistema solicita ao ator permissão para vincular a sua conta do Google com o aplicativo.
- E2.2) O Ator recusa a solicitação.
- E2.3) Sistema não permite o login do ator.

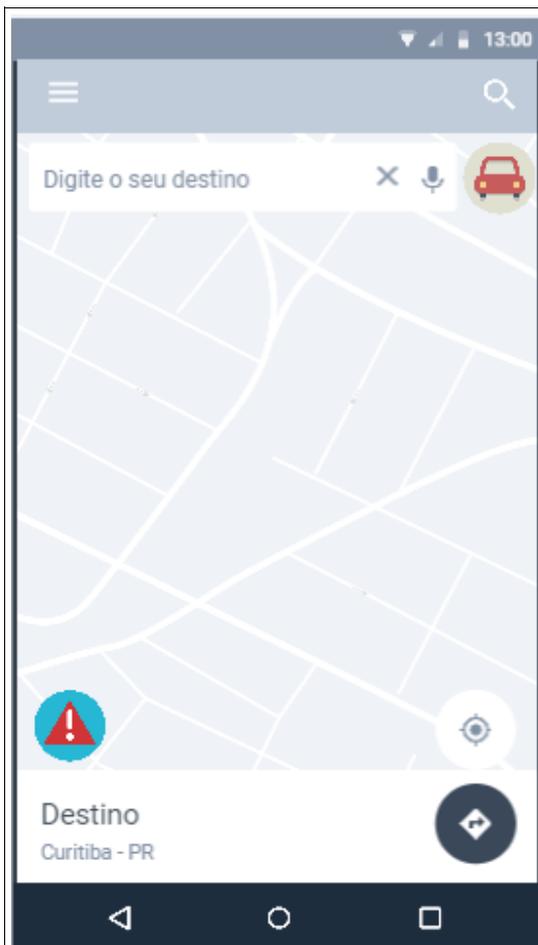
## 2.1.2 CASO DE USO: UC002 - ALERTAR:

Descrição:

Este caso de uso é iniciado pelo usuário.

Permite ao usuário fazer um alerta de perigo a todos os seus amigos e grupos instantaneamente.

*DATAVIEW:*



## DATAVIEW 2 - ALERTAR

Ator primário:

Usuário.

Pré-Condição: Estar em uma rota.

Pós-Condição: Nenhum.

Fluxo de Eventos Principal:

- 1) O Usuário clica 4x no botão de desligar do celular.
- 2) O sistema envia uma mensagem a todos os amigos e grupos do usuário.
- 3) O caso de uso é finalizado.

Fluxo Alternativo:

A1: Usuário pode cancelar o envio do alerta.

- A1.1) Usuário clica no botão de alerta.
- A1.2) Usuário clica no botão de cancelar.
- A1.3) Sistema cancela o envio do alerta.

Fluxo de Exceção:

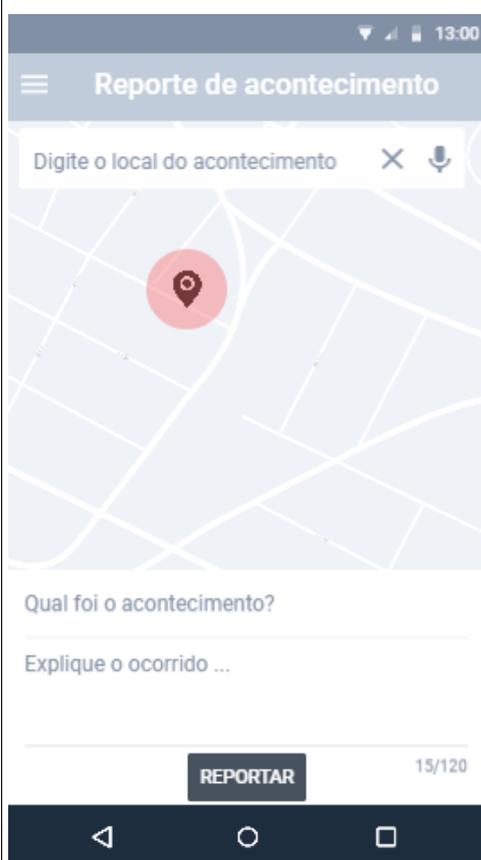
Nenhum.

### 2.1.3 CASO DE USO: UC003 - REPORTAR ACONTECIMENTO:

#### Descrição:

Este caso de uso é iniciado pelo usuário. Permite ao usuário informar qualquer tipo de acontecimento anormal em que a informação venha a ser útil para a comunidade, no qual o próprio usuário demarca no mapa o local do acontecimento, informa qual foi o acontecimento e detalha o que aconteceu.

#### DATAVIEW



#### DATAVIEW 3 - REPORTAR ACONTECIMENTO

#### Ator primário:

Usuário.

Pré-Condição: Estar logado no sistema.

Pós-Condição: Informar todos os dados obrigatórios.

#### Fluxo de Eventos Principal:

- 1) O Usuário demarca no mapa o local do acontecimento.
- 2) O Usuário informa qual foi o acontecimento.

- 3) O Usuário detalha o ocorrido.
- 4) O Usuário clica no botão “Reportar”.
- 5) O sistema salva os dados informados pelo usuário.
- 6) O caso de uso é finalizado.

Fluxo Alternativo:

Nenhum.

Fluxo de Exceção:

E1: O Usuário não marca o local do acontecimento ou deixa algum campo em branco:

- E1.1) O Usuário não informa algum dado obrigatório.
- E1.2) O Usuário clica no botão “Reportar”.
- E1.3) O sistema informa uma mensagem de erro.

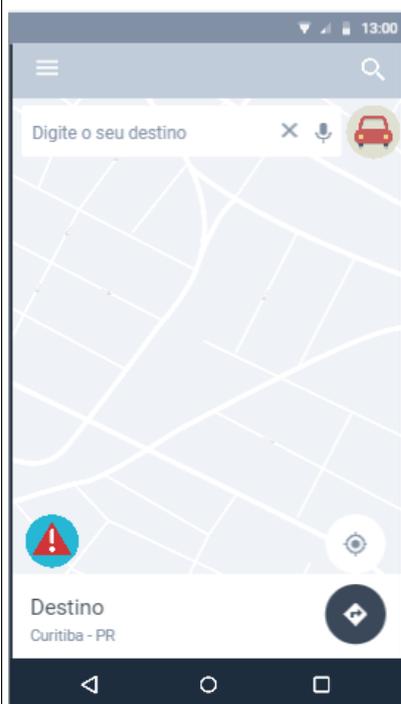
#### 2.1.4 CASO DE USO UC004 - ELABORAR ROTA:

Descrição:

Este caso de uso é iniciado pelo usuário.

Permite ao usuário elaborar a sua rota, no qual ele informa o ponto de partida, o ponto de destino e a forma de locomoção.

*DATAVIEW*



DATAVIEW 4 - ELABORAR ROTA

Ator primário:

Usuário.

Pré-Condição: Estar logado no sistema.

Pós-Condição: Ter informado um destino válido.

Fluxo de Eventos Principal:

- 1) O sistema carrega a posição atual do usuário.
- 2) O Usuário digita o local de destino.
- 3) O Usuário seleciona a forma de transporte.
- 4) O sistema calcula a rota.
- 5) O Usuário clica no botão para começar a rota.
- 6) O caso de uso é finalizado.

Fluxo Alternativo:

Nenhum.

Fluxo de Exceção:

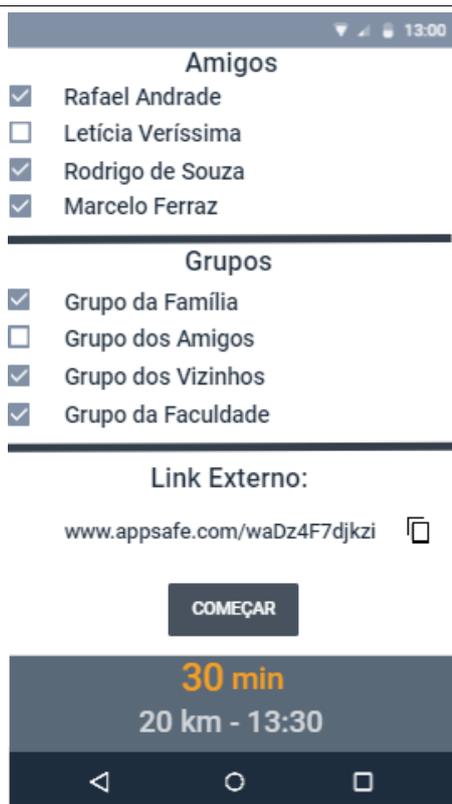
- E1: O Usuário elabora a rota sem digitar o ponto de destino.
- E1.1) O Usuário clica no botão para começar a rota.
- E1.2) O sistema retorna uma mensagem de erro.
- E1.3) O Usuário retorna para a tela inicial.

#### 2.1.5 CASO DE USO UC005 - SELECIONAR OBSERVADORES:

Descrição:

Este caso de uso é iniciado pelo usuário. Permite ao usuário selecionar quais pessoas terão permissão de visualizar o seu trajeto a partir da sua lista de amigos e de grupos.

*DATAVIEW*



#### DATAVIEW 5 - SELECIONAR OBSERVADORES

Pré-condições: Este caso de uso pode iniciar somente se:

Uma rota for gerada pelo usuário (UC004 - ELABORAR ROTA).

Pós-condições: Nenhum.

Ator primário:

Usuário.

Fluxo de Eventos Principal:

- 1) O sistema carrega a lista de amigos e de grupos.
- 2) O Usuário seleciona os amigos desejados.
- 3) O Usuário seleciona os grupos desejados.
- 4) O Usuário clica no botão "Começar".
- 5) O caso de uso é finalizado.

Fluxo Alternativo:

Nenhum.

Fluxo de Exceção:

Nenhum.

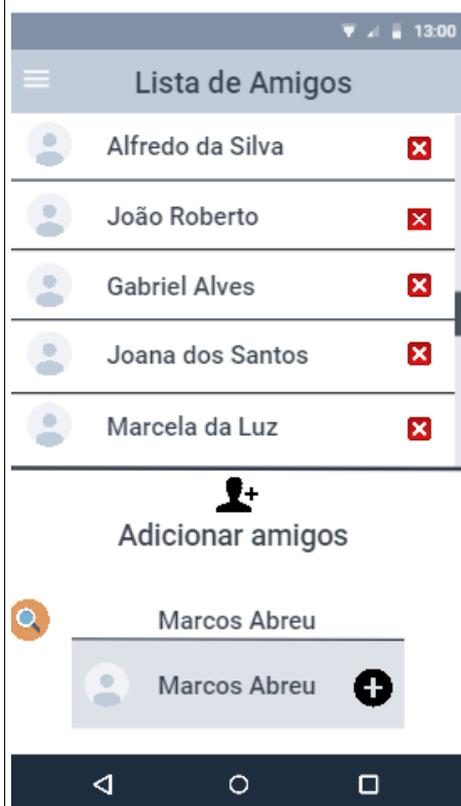
## 2.1.6 CASO DE USO UC006 - GERENCIAR AMIGOS:

### Descrição:

Este caso de uso é iniciado pelo usuário.

Permite ao usuário gerenciar a sua lista de amigos, adicionando e excluindo usuários.

### DATAVIEW



### DATAVIEW 6 - GERENCIAR AMIGOS.

#### Ator primário:

Usuário.

Pré-Condição: Estar logado no sistema.

Pós-Condição: Nenhum.

#### Fluxo de Eventos Principal:

- 1) O sistema carrega os amigos do usuário.
- 2) O Usuário digita o email do usuário a ser adicionado.
- 3) O Usuário clica no botão de adicionar.
- 4) O sistema adiciona o usuário.
- 5) O caso de uso é finalizado.

**Fluxo Alternativo:**

A1: O Usuário deleta um amigo.

A1.1: O Usuário escolhe o amigo a ser deletado.

A1.2: O Usuário clica no botão de deletar.

A1.3: O sistema retira o amigo deletado da lista de amigos.

**Fluxo de Exceção:**

E1: O Usuário pesquisa um outro usuário que é inexistente no sistema.

E1.1: O Usuário digita o email do usuário a ser adicionado.

E1.2: O sistema retorna uma mensagem de erro.

E1.3: O Usuário é redirecionado para a tela de amigos.

E2: O Usuário tenta adicionar um usuário já adicionado em sua lista de amigos.

E2.1: O Usuário digita o email do usuário a ser adicionado.

E2.2: O sistema retorna uma mensagem de erro.

E2.3: O Usuário é redirecionado para a tela de amigos.

**2.1.7 CASO DE USO UC007 - GERENCIAR GRUPOS:****Descrição:**

Este caso de uso é iniciado pelo usuário.

Permite ao usuário gerenciar a sua lista de grupos, podendo criar um novo e adicionar os usuários de seu desejo.

***DATAVIEW:***



#### DATAVIEW 7 - GERENCIAR GRUPOS.

Ator primário:

Usuário.

Pré-Condição: Estar logado no sistema.

Pós-Condição: Nenhuma.

Fluxo de Eventos Principal:

- 1) O sistema carrega os grupos do usuário.
- 2) O Usuário digita o nome de um grupo para criá-lo.
- 3) O Usuário digita o email dos usuários desejados para adicioná-los para o grupo.
- 4) O sistema adiciona o usuário no grupo.
- 5) O Usuário clica no botão "Criar".
- 6) O sistema cria o grupo para o Usuário
- 7) O sistema carrega o novo grupo na lista de grupos.
- 8) O caso de uso é finalizado.

Fluxo Alternativo:

A1: O Usuário visualiza as informações dos seus grupos.

A1.1: O Usuário clica no botão de informações vinculado ao grupo de seu desejo.

A1.2: O sistema abre uma tela e carrega as informações do grupo visualizado.

A2: O Usuário deleta um determinado grupo.

A2.1: O Usuário clica no botão de deletar vinculado ao grupo desejado.

A2.2: O sistema envia uma mensagem para confirmar a ação.

A2.3: O Usuário confirma a ação.

A2.4: O sistema retira o Usuário do grupo selecionado.

Fluxo de Exceção:

E1: O Usuário cria um novo grupo sem adicionar nenhum outro Usuário.

E1.1: O Usuário digita o nome do grupo.

E1.2: O Usuário clica no botão "Criar".

E1.3: O sistema retorna uma mensagem de erro.

E2: Usuário cria um novo grupo e adiciona um outro Usuário que não existe no sistema.

E2.1: O Usuário digita o nome do grupo.

E2.2: O Usuário digita o email de um Usuário inexistente no sistema.

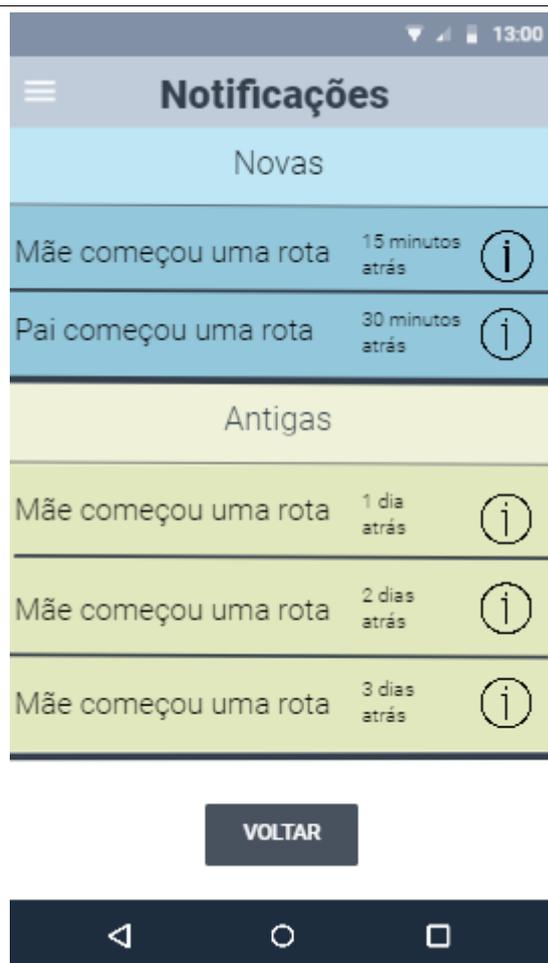
E2.3: O sistema retorna uma mensagem de erro.

#### 2.1.8 CASO DE USO UC008 - VISUALIZAR CONVITES:

Descrição:

Este caso de uso é iniciado pelo usuário. Permite ao usuário visualizar os seus convites de rotas.

*DATAVIEW*



DATAVIEW 8 - VISUALIZAR CONVITES.

Ator primário:

Usuário.

Pré-Condição: Estar logado no sistema.

Pós-Condição: Nenhuma.

Fluxo de Eventos Principal:

- 1) O sistema carrega os convites do Usuário.
- 2) O Usuário clica no botão de aceitar ou recusar.
- 3) O sistema realiza a operação desejada pelo Usuário.
- 4) O caso de uso é finalizado.

Fluxo Alternativo:

Nenhum.

Fluxo de Exceção:

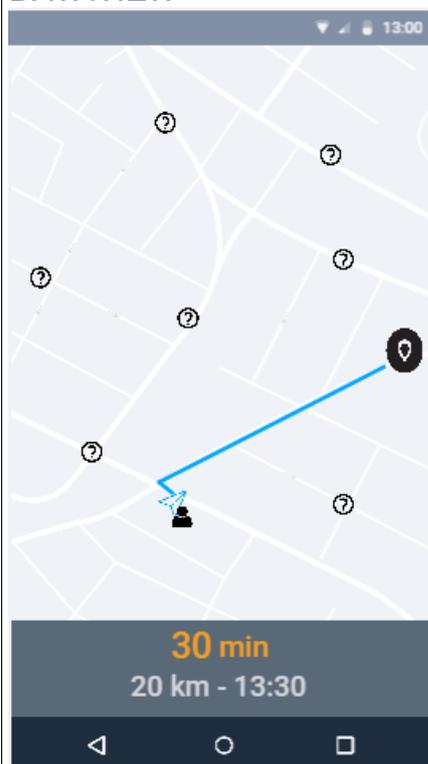
Nenhum.

## 2.1.9 CASO DE USO UC009 - VISUALIZAR ROTAS

### Descrição:

Este caso de uso é iniciado pelo observador. Permite ao observador acompanhar em tempo real a rota disponibilizada por outro usuário através de um convite.

### DATAVIEW



### DATAVIEW 9 - VISUALIZAR ROTA.

Ator primário:

Observador.

Pré-Condição: Ser convidado para acompanhar uma rota em tempo real.

Pós-Condição: Nenhuma.

Fluxo de Eventos Principal:

- 1) Sistema carrega a rota do Usuário.
- 2) Observador acompanha a rota do Usuário.
- 3) O caso de uso é finalizado.

Fluxo Alternativo:

Nenhum.

Fluxo de Exceção:

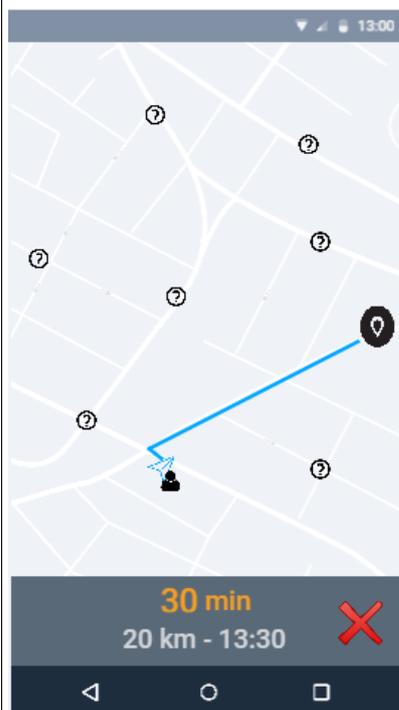
Nenhum.

## 2.1.10 CASO DE USO 010 - CANCELAR ROTA:

### Descrição:

Este caso de uso é iniciado pelo usuário. Permite ao usuário cancelar a sua rota atual assim que desejar.

### DATAVIEW



### DATAVIEW 10 - CANCELAR ROTA

Ator primário:

Usuário.

Pré-Condição: Estar em uma rota.

Pós-Condição: Nenhuma.

Fluxo de Eventos Principal:

- 1) Usuário deseja cancelar a rota.
- 2) Usuário clica no botão de cancelar a rota.
- 3) Sistema finaliza a rota.
- 4) O caso de uso é finalizado.

Fluxo Alternativo:

Nenhum.

Fluxo de Exceção:

Nenhum.

## 3 DICIONÁRIO DE DADOS

<b>ALERTA</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
id	INT(11)	Yes	Chaves primaria alerta
descricao	VARCHAR(500)	No	Descrição do alerta por exemplo, indicando que ocorreu um assalto e as características do assaltante
data_criacao	TIMESTAMP	No	Data da criação do alerta
usuario_id	INT(11)	No	ID do usuário que criou o alerta
tipo_alerta_id	INT(11)	No	Chave estrangeira do tipo de alerta
coordenada_id	INT(11)	No	Chave estrangeira da coordenada do alerta
<b>CONTATO</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
usuario_id	INT(11)	Yes	Chave estrangeira do usuário
contato_id	INT(11)	Yes	Chave estrangeira do contato
<b>COORDENADA</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
id	INT(11)	Yes	Chaves primaria coordenada
latitude	DOUBLE	No	valor da latitude da coordenada
longitude	DOUBLE	No	valor da longitude da coordenada
local	VARCHAR(255)	No	Local, endereço, nome do lugar
<b>GRUPO</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
id	INT(11)	Yes	Chaves primaria grupo
nome	VARCHAR(45)	No	Nome identificador do grupo
usuario_id	INT(11)	No	Criador do grupo
<b>PANICO</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
id	INT(11)	Yes	Chaves primaria panico
data	TIMESTAMP	No	Data que foi acionado o panico
usuario_id	INT(11)	No	Chave estrangeira relativa ao criador do pânico
coordenada_id	INT(11)	No	Chaves estrangeira da coordenada do pânico
trajeto_uuid	VARCHAR(45)	No	Chave estrangeira do trajeto do usuário
<b>regiao_perigo</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
id	INT(11)	Yes	Chaves primaria região de perigo
raio	DOUBLE	No	Raio da região de perigo
intensidade	DOUBLE	No	Intensidade calculada do nivel de perigo
localidade	VARCHAR(255)	No	Descrição da regioao, bairro, localidade
coordenada_id	INT(11)	No	Chaves estrangeira da coordenada da região de perigo

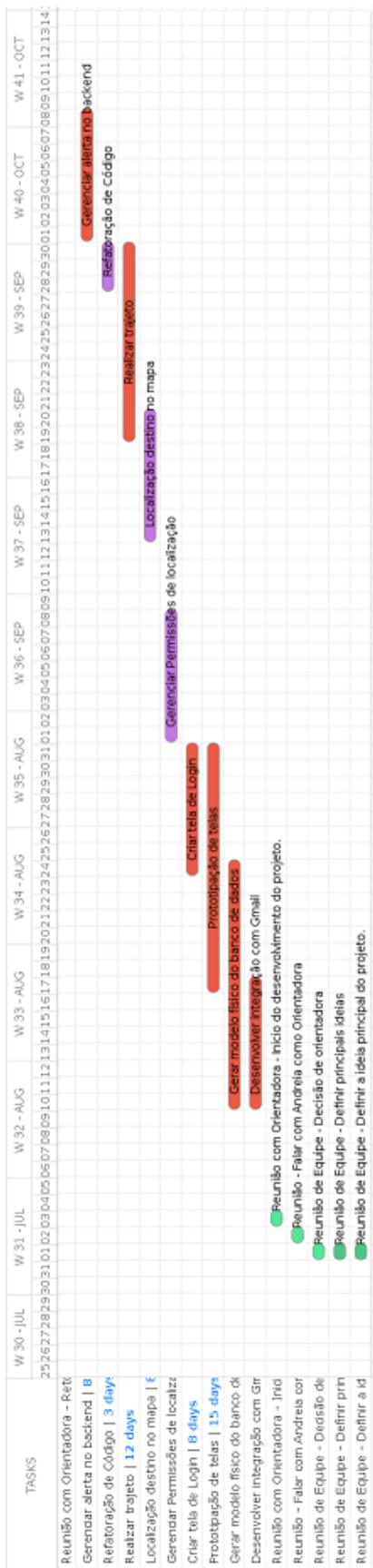
<b>regiao_perigo_alerta</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
alerta_id	INT(11)	Yes	Chaves estrangeira do alerta
regiao_perigo_id	INT(11)	Yes	Chaves estrangeira da região perigo
<b>tipo_alerta</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
id	INT(11)	Yes	Chaves primaria tipo alerta
nome	VARCHAR(45)	No	Nome do tipo por exemplo: Assaltante
gravidade	INT(11)	No	Gravidade do alerta (numero fixo, de 1 a 5 por exemplo)
expiracao	INT(11)	No	Tempo que o alerta irá expirar em minutos
<b>tipo_fim_trajeto</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
id	INT(11)	Yes	Chaves primaria tipo fim trajeto
descricao	VARCHAR(45)	No	Descrição do fim do trajeto (ex: concluído, desligado, interrompido, cancelado)
<b>tipo_trajeto</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
id	INT(11)	Yes	Chaves primaria tipo trajeto
descricao	VARCHAR(45)	No	Tipo do trajeto (ex: Andando, dirigindo, etc..)
modo	VARCHAR(45)	No	Modo de locomoção
<b>trajeto</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
uuid	VARCHAR(45)	Yes	Chaves primaria trajeto
data_inicio	TIMESTAMP	No	Data de início do trajeto
bateria_inicio	DOUBLE	No	Valor da bateria no inicio
bateria_baixa	TINYINT(4)	No	Boolean caso a bateria esteja baixa
distancia	DOUBLE	No	Em metros
duracao_estimada	INT(11)	No	Em segundos
pontos	TEXT	No	Uma array em JSON dos pontos que compoem a rota
usuario_id	INT(11)	No	Chaves estrangeira do usuário criador
partida_coordenada_id	INT(11)	No	Chaves estrangeira da coordenada de partida
destino_coordenada_id	INT(11)	No	Chaves estrangeira da coordenada de destino
tipo_trajeto_id	INT(11)	No	Chaves estrangeira do tipo de trajeto
fim_trajeto_id	INT(11)	No	Chaves estrangeira referente ao fim do trajeto

<b>trajeto_convite</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
trajeto_uuid	VARCHAR(45)	Yes	Chaves estrangeira do trajeto
usuario_id	INT(11)	Yes	Chaves estrangeira convidado
data_enviado	TIMESTAMP	No	Data de envio do convite
data_aceito	TIMESTAMP	No	Data no qual o convite foi aceito
data_recusado	TIMESTAMP	No	Data de recusa
<b>trajeto_fim</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
id	INT(11)	Yes	Chaves primaria trajeto fim
data	TIMESTAMP	No	Caso o trajeto tenha sido cancelado
bateria	DOUBLE	No	Porcentagem da bateria no fim do trajeto
qtd_assistindo	INT(11)	No	Quantidade de pessoas assistindo ao fim do trajeto
tipo_fim_trajeto_id	INT(11)	No	Chaves estrangeira do tipo fim
<b>trajeto_observador</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
id	INT(11)	Yes	Chaves primaria trajeto observador
guest_ip	VARCHAR(45)	No	Usuário guest IP
guest_agent	VARCHAR(255)	No	Usuário guest agent
ativo	TINYINT(4)	No	Esconder para a pessoa que está observando o percurso
usuario_id	INT(11)	No	Pessoa que está acompanhando o percurso
data_conectado	TIMESTAMP	No	Data da conexão a rota
data_desconectado	TIMESTAMP	No	Data desconectado
trajeto_uuid	VARCHAR(45)	No	Chaves estrangeira do trajeto
<b>trajeto_verificador</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
id	INT(11)	Yes	Chaves primaria trajeto verificador
data_saida	TIMESTAMP	No	Data que saiu do percurso
data_volta	TIMESTAMP	No	Data que voltou para o percurso
data_parada	TIMESTAMP	No	Data que alertou que ficou tempo demais parado
trajeto_uuid	VARCHAR(45)	No	Chaves estrangeira do trajeto
coordenada_id	INT(11)	No	Chaves estrangeira da coordenada

<b>usuario</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
id	INT(11)	Yes	Chaves primaria do usuario
nome	VARCHAR(45)	No	Nome do usuário
id_social	VARCHAR(45)	No	Id da rede social (facebook, google+)
cpf	VARCHAR(14)	No	Cpf do usuário
data_inclusao	TIMESTAMP	No	Data da inclusão no sistema
data_exclusao	TIMESTAMP	No	Data de exclusão
data_bloqueado	TIMESTAMP	No	Data de bloqueio no sistema
celular	VARCHAR(25)	No	Celular do usuário
celular_responsavel	VARCHAR(25)	No	Celular Responsavel
cep	VARCHAR(11)	No	Cep do usuário
email	VARCHAR(45)	No	Email único do usuário
img_url	VARCHAR(500)	No	URL da imagem de perfil
<b>usuario_grupo</b>			
<b>Name</b>	<b>Data Type</b>	<b>PK</b>	<b>Comment</b>
usuario_id	INT(11)	No	Chaves estrangeira do amigo
grupo_id	INT(11)	No	Chaves estrangeira do grupo

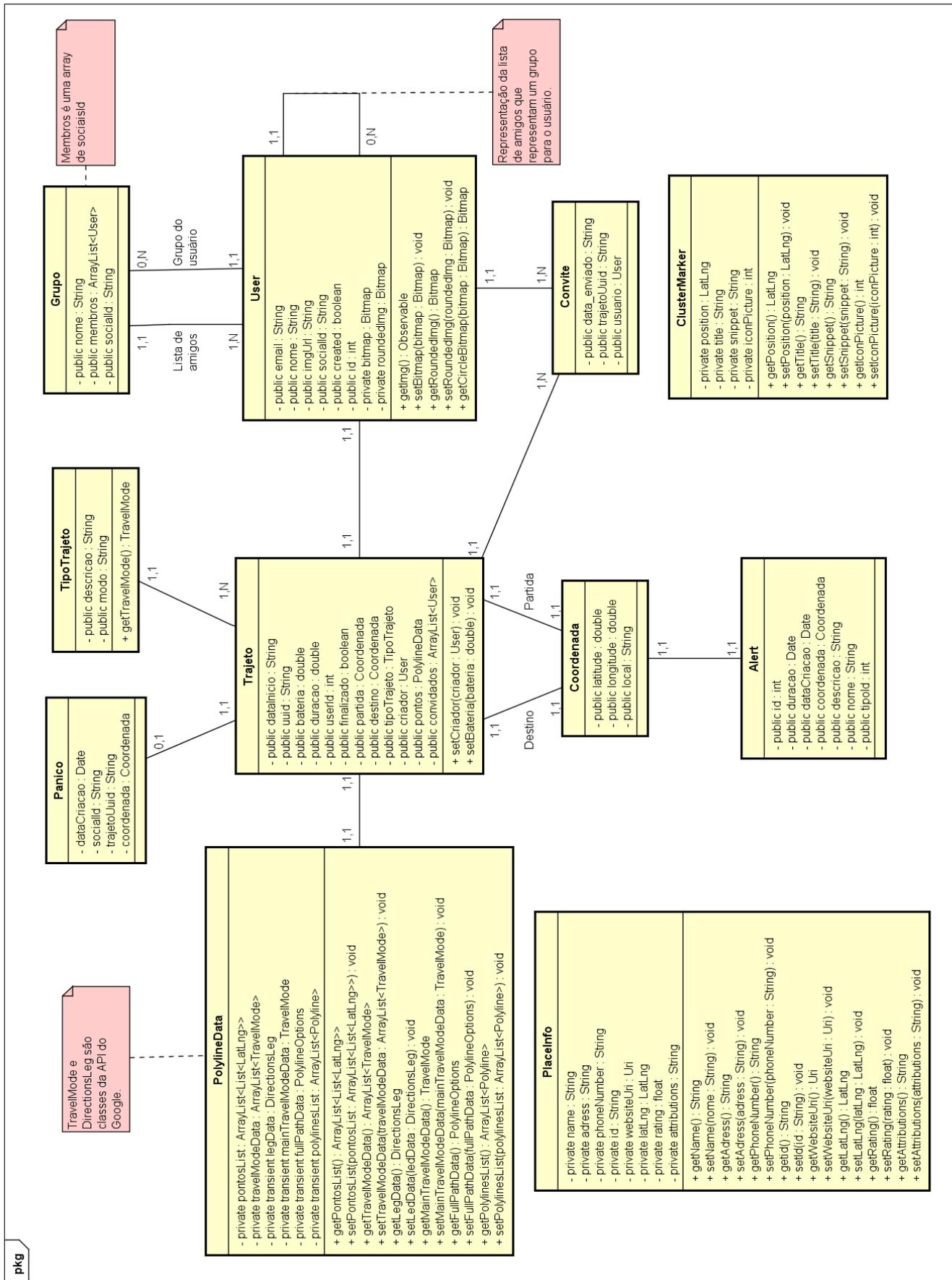


## 5 DIAGRAMA DE GANTT

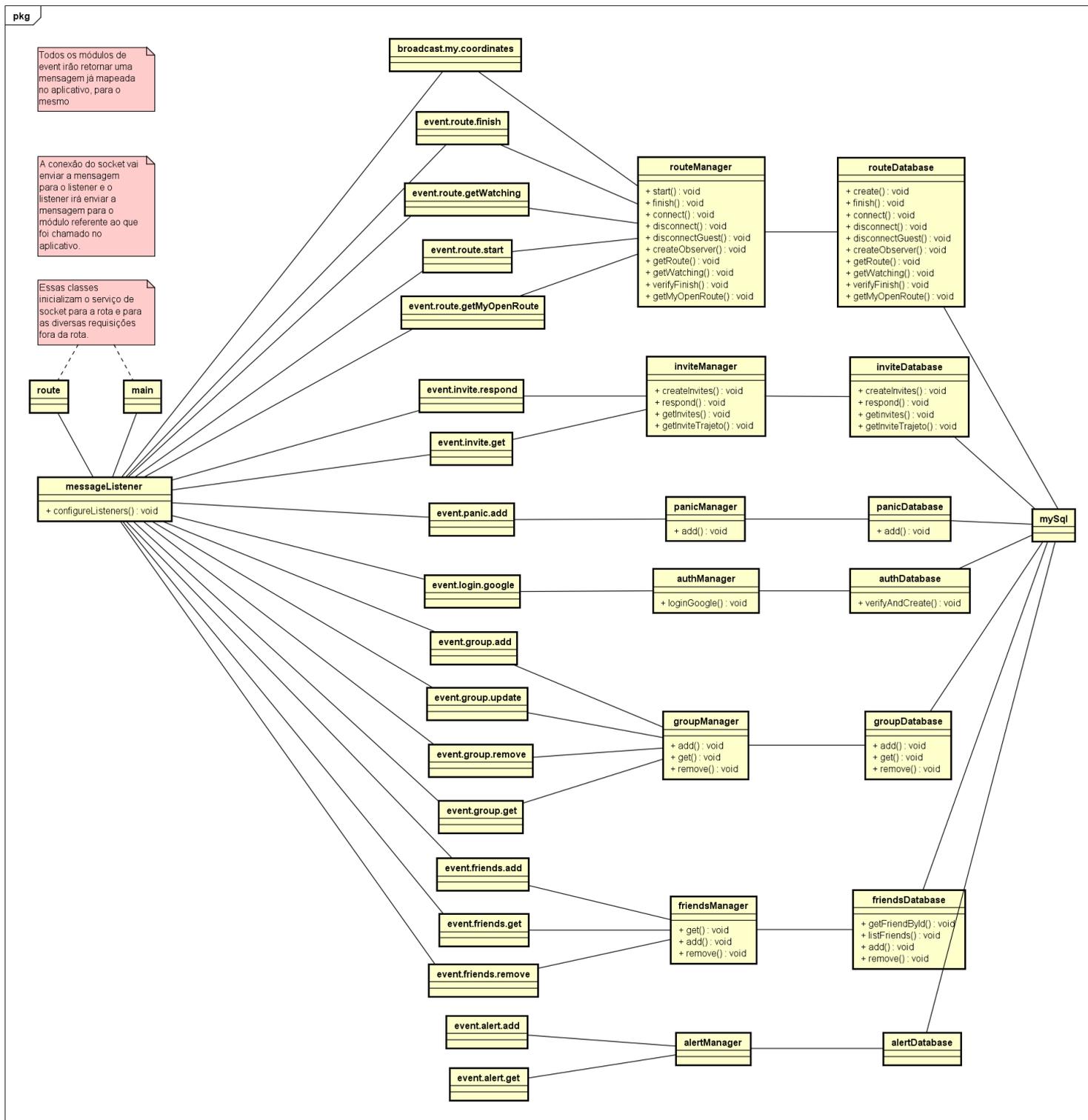




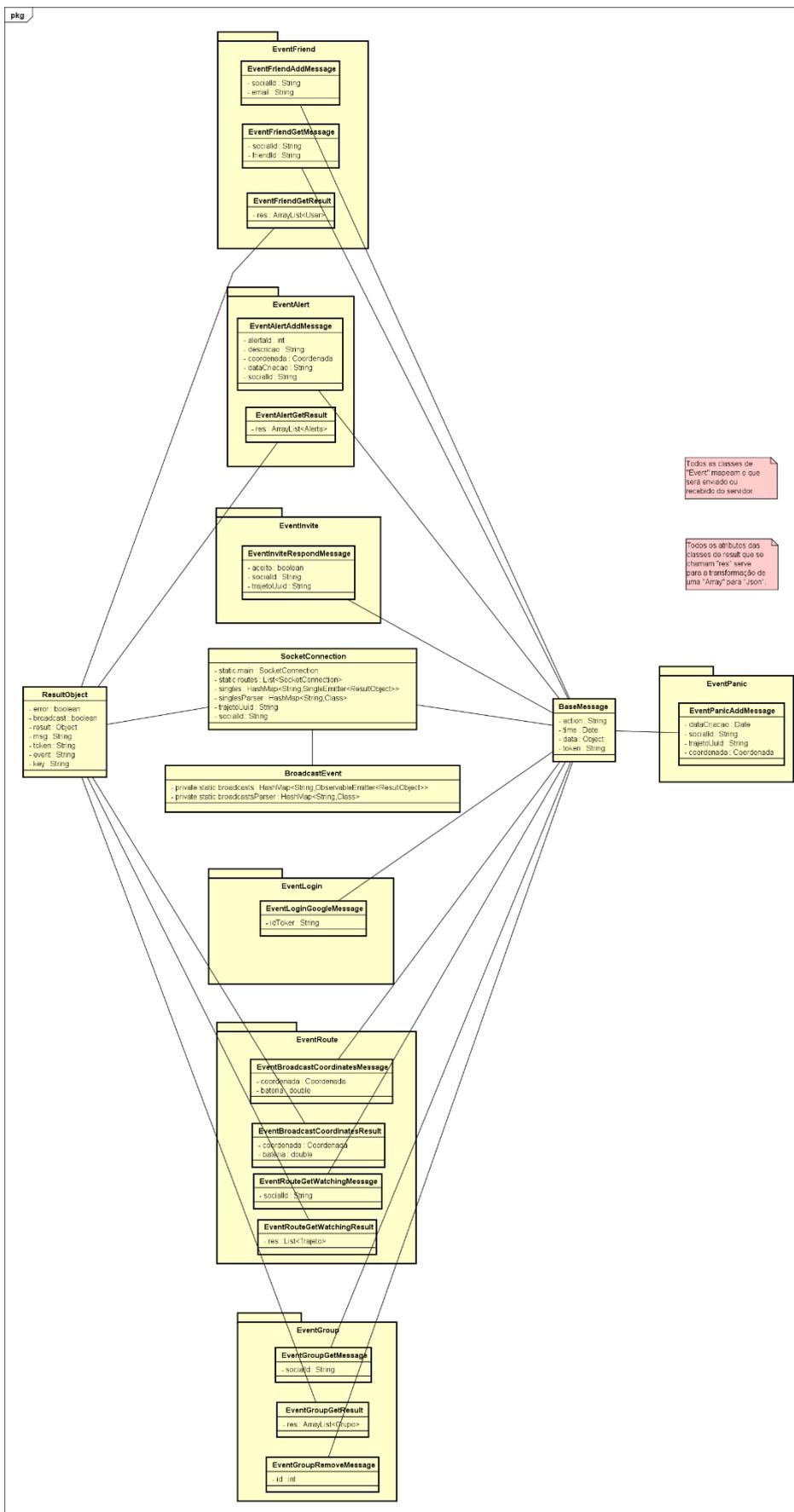
## 6 DIAGRAMA DE CLASSE(APLICATIVO)



## 7 DIAGRAMA DE CLASSE(SERVIDOR)

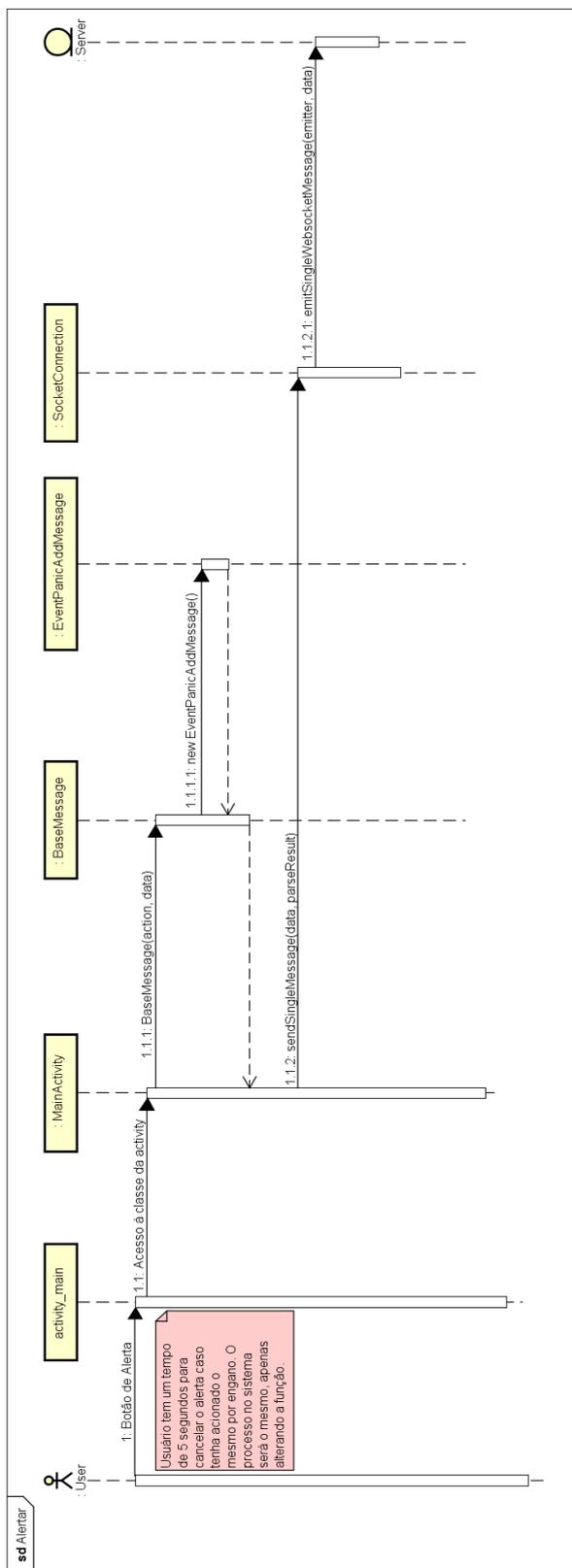


## 8 DIAGRAMA DE CLASSE (CONEXÃO AO SERVIDOR)

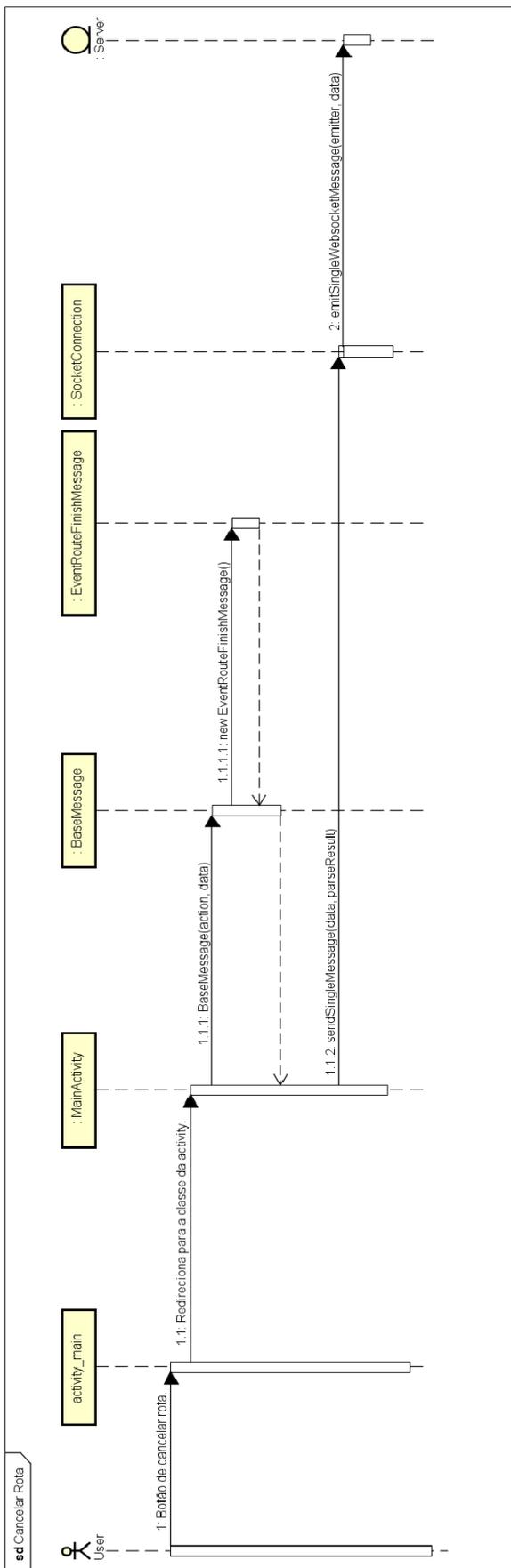


## 9 DIAGRAMA DE SEQUÊNCIA

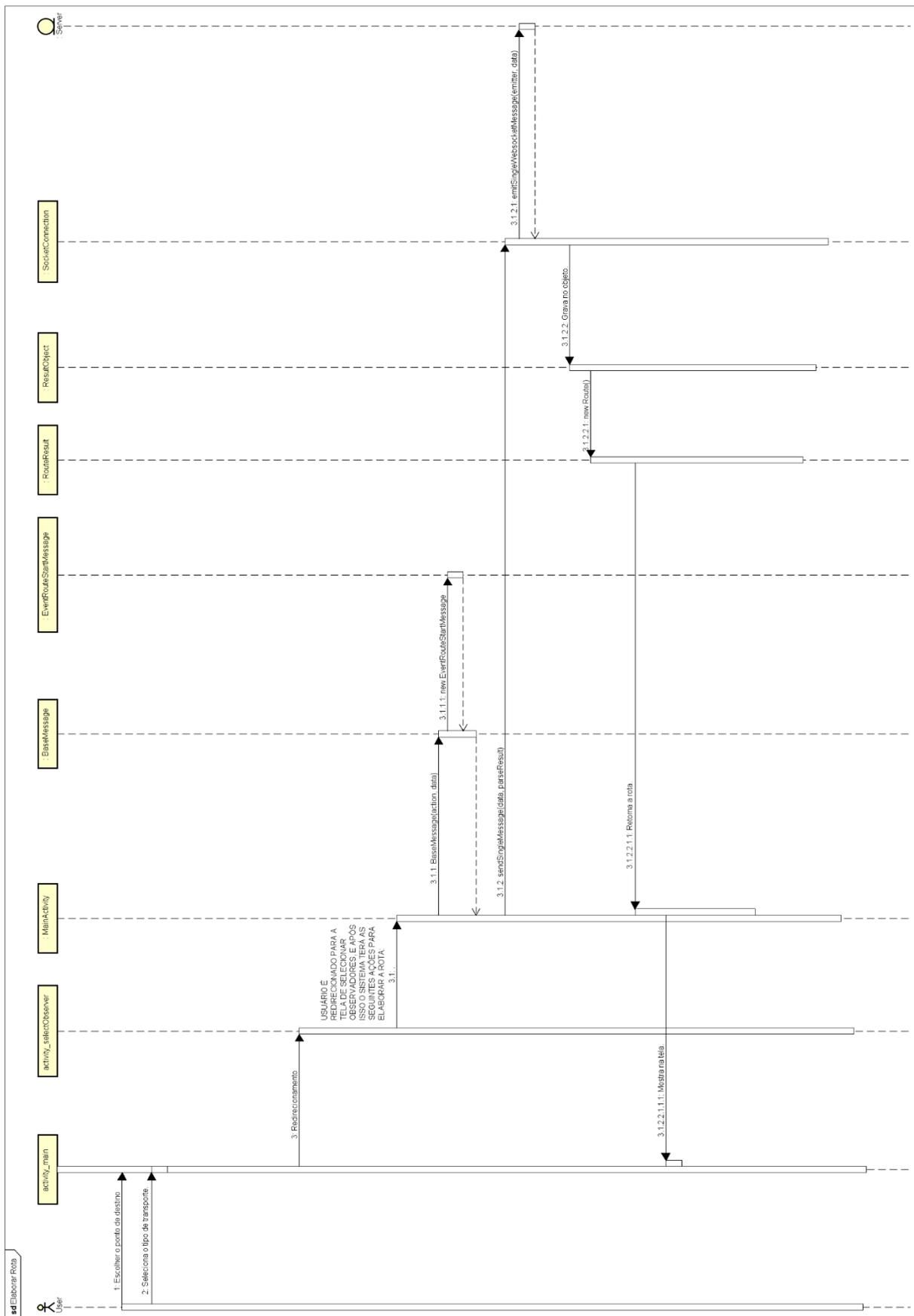
### 9.1 ALERTA



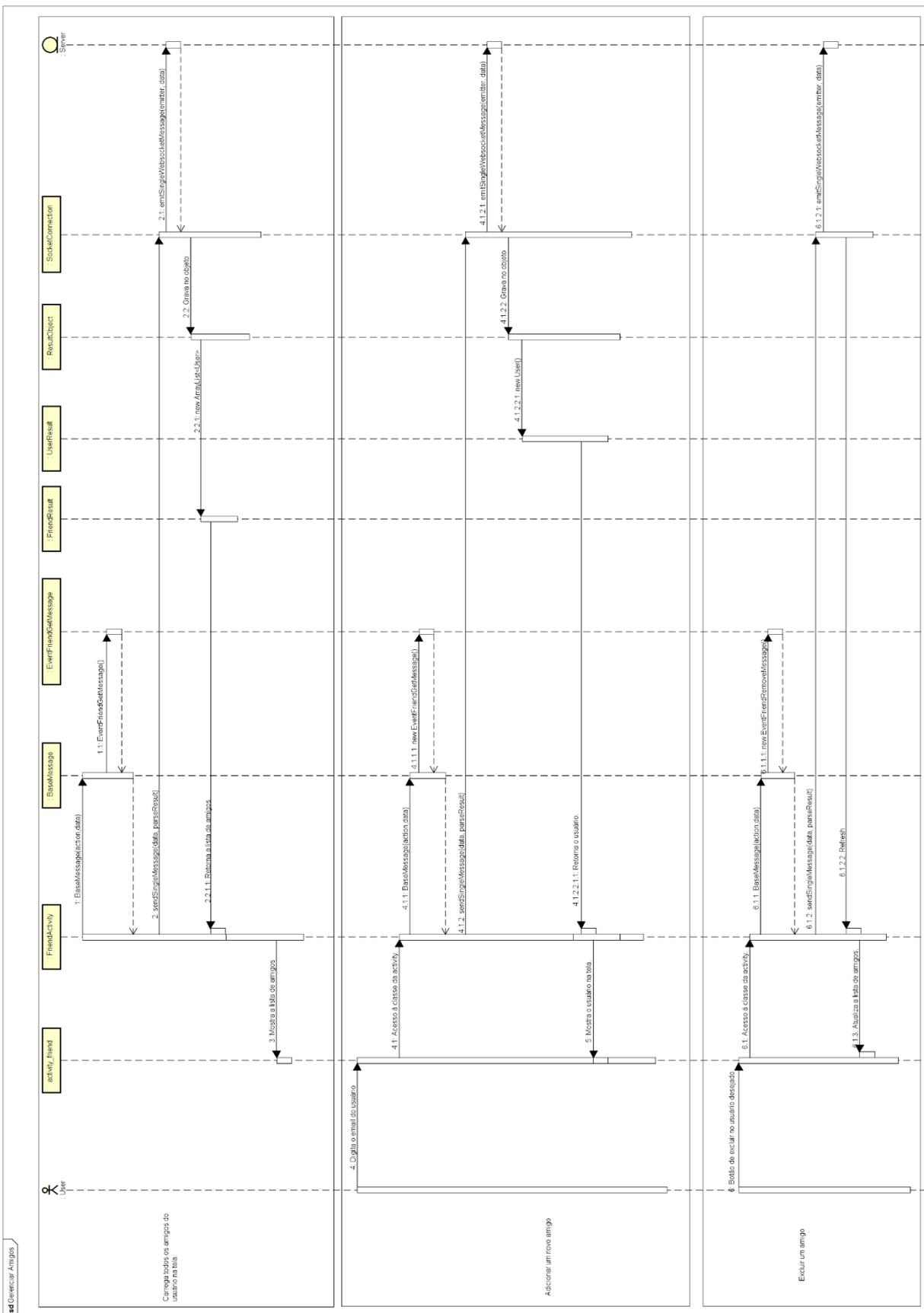
## 9.2 CANCELAR ROTA



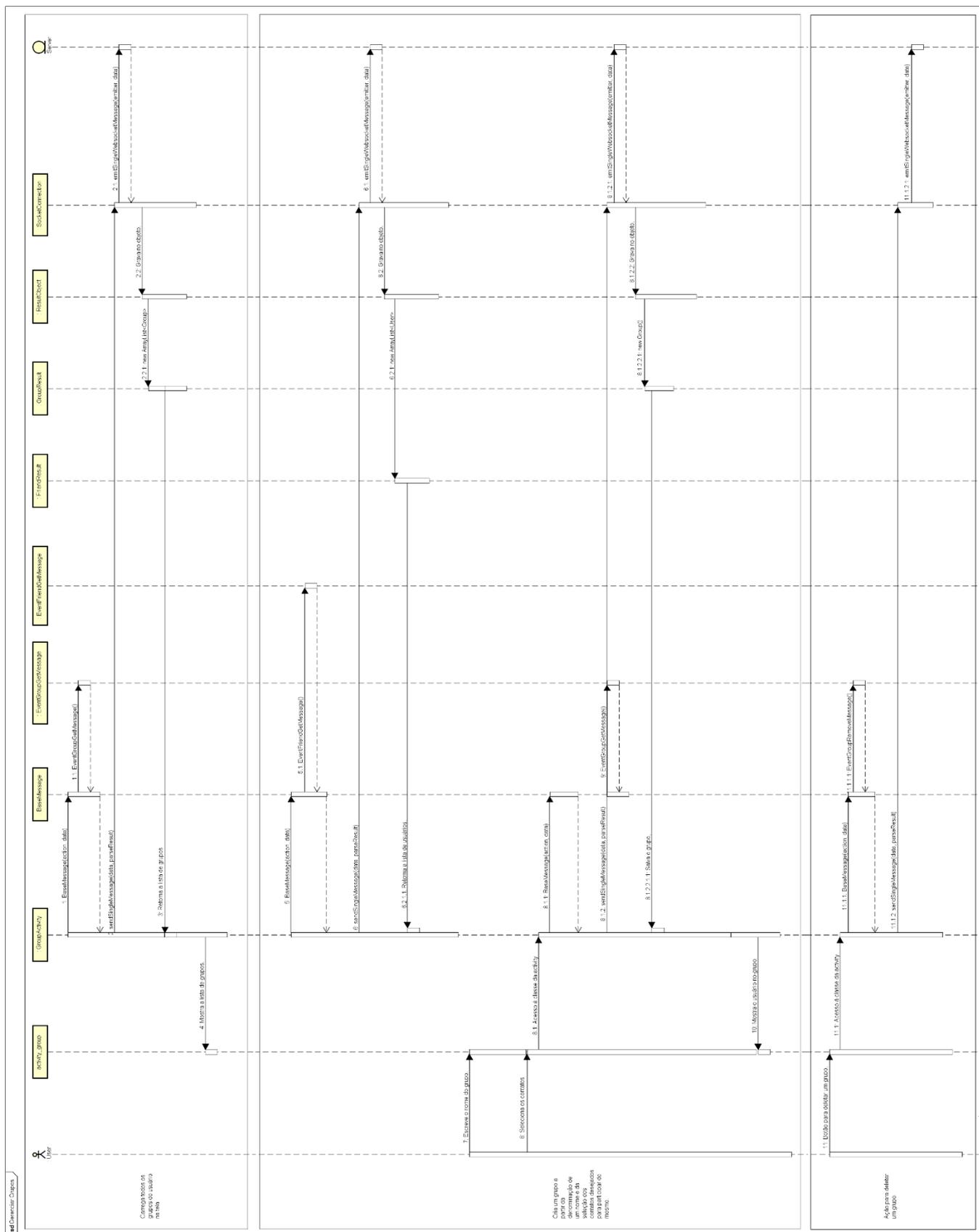
### 9.3 ELABORAR ROTA



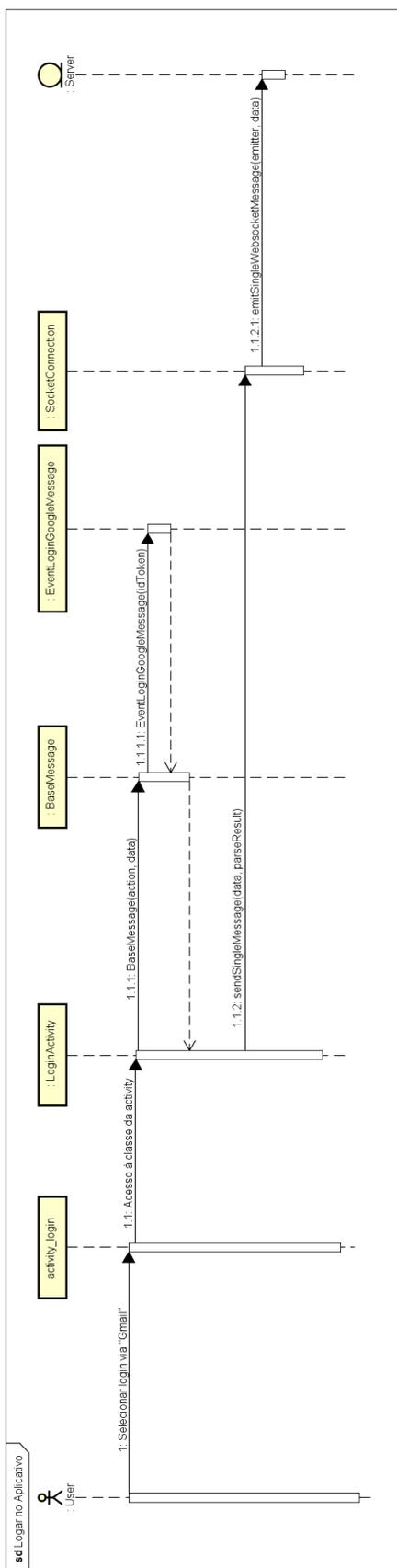
### 9.4 GERENCIAR AMIGOS



## 9.5 GERENCIAR GRUPOS

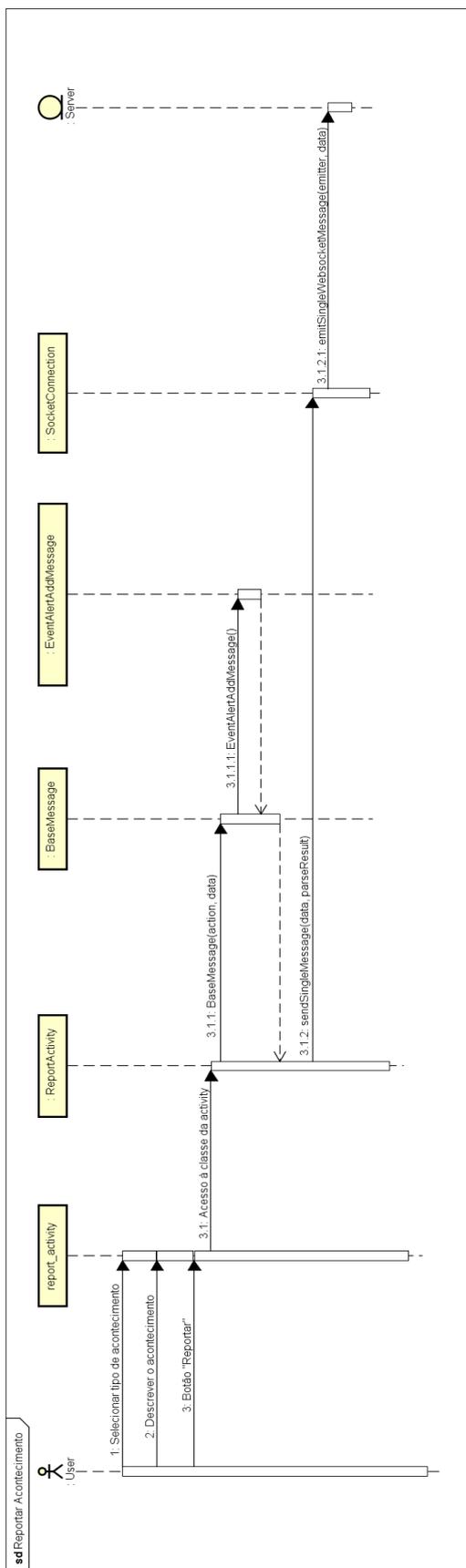


## 9.6 LOGAR NO APLICATIVO



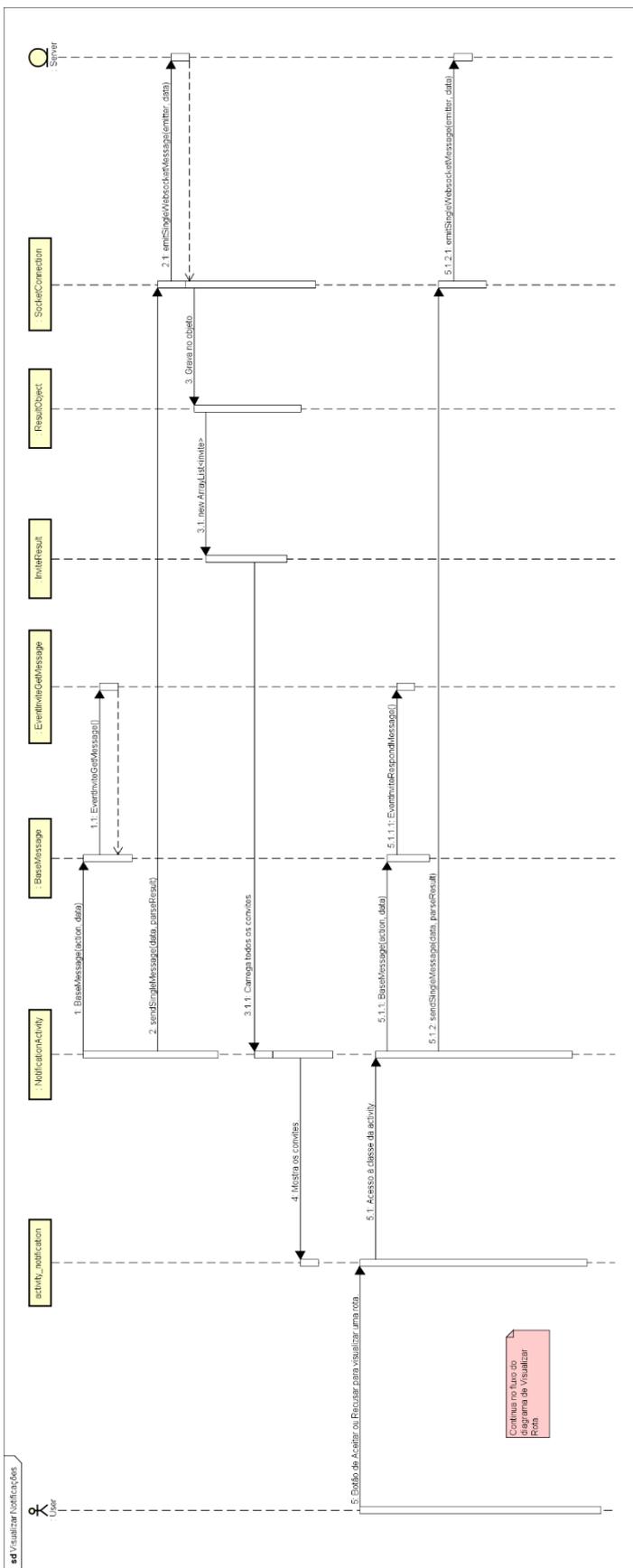
powered by Astah

## 9.7 REPORTAR ACONTECIMENTO

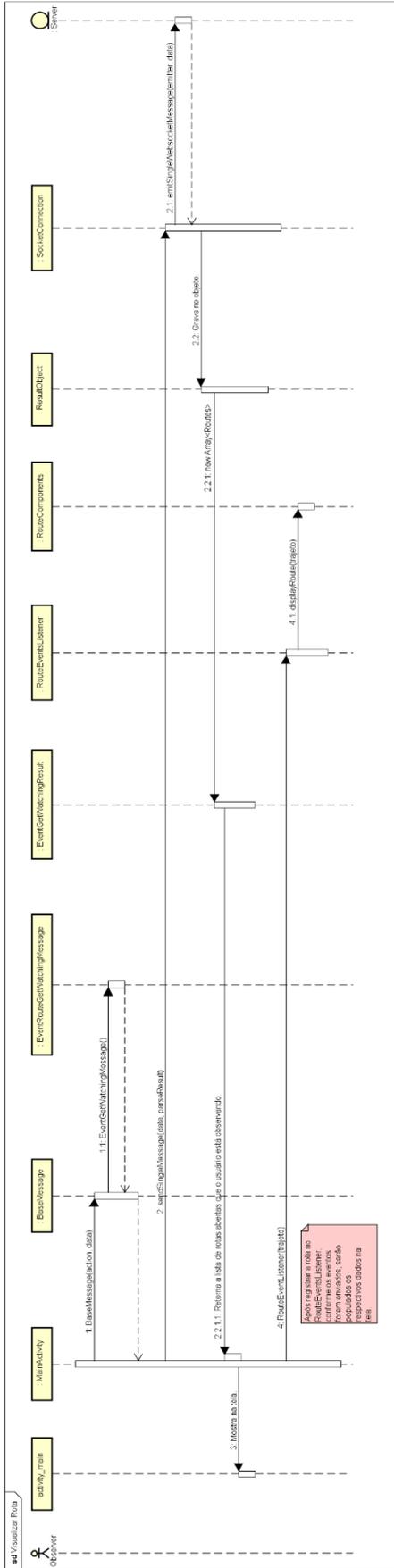




## 9.9 VISUALIZAR CONVITES



### 9.10 VISUALIZAR ROTAS



powered by Act1

## 10 TABELA DE BUGS CONHECIDOS

Tabela 4 TABELA DE BUGS

FONTES: os autores(2018).

Descrição	Como resolver
Ao receber o segundo convite de uma rota enquanto a aplicação está aberta, o aplicativo da erro e quebra	Retirar a notificação de dentro do listener do "route.invites" na MainActivity
Ao aceitar convite na aba de convites do menu lateral, não aparece no mapa a rota que foi aceita	Garantir que está sendo executado denovo o onMapReady da MainActivity para executar o "event.route.getWatching" novamente