
**Viagem Segura: Desenvolvimento de
Aplicativo Android para Auxiliar na Redução
de Acidentes de Trânsito em Rodovias de
Minas Gerais**

Eduardo José Rodrigues Silva

PROJETO DE CONCLUSÃO DE CURSO

Data de Depósito: 30/11/2021

Assinatura: _____

Viagem Segura: Desenvolvimento de Aplicativo Android para Auxiliar na Redução de Acidentes de Trânsito em Rodovias de Minas Gerais

Eduardo José Rodrigues Silva

Willyan Michel Ferreira

Monografia apresentada ao Centro Universitário de Formiga UNIFOR/MG, como requisito parcial para obtenção do título de bacharel em Ciência da Computação, sob a orientação do Prof. Willyan Michel Ferreira

Unifor-MG - Formiga

11/2021

*A Deus, a minha família, e
a todos que me apoiaram*

Agradecimentos

Depois dessa jornada desafiadora, a sensação que fica é de dever cumprido e sonho realizado. Agradeço primeiramente a Deus, que sempre esteve comigo e que me capacitou para concluir essa etapa tão importante em minha vida. Agradeço também a minha família, que me deram suporte durante todos esses anos, em especial meu pai e minha mãe, que sempre me apoiaram para que esse sonho fosse possível, essa conquista é nossa. Também deixo meu agradecimento a todos os colegas e amigos que estiveram comigo nessa jornada, depois de uma longa caminhada chegamos ao final e alcançamos a nossa meta, desejo muito sucesso a todos vocês. E por último mas não menos importante, agradeço aos professores, vocês foram verdadeiros mestres e tenho muita admiração e respeito por todos, obrigado por todo conhecimento compartilhado, levarei cada ensinamento para minha vida.

Obrigado a todos!

”A tarefa não é tanto ver aquilo que ninguém viu, mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo vê.”

Arthur Schopenhauer

Sumário

Lista de Figuras	ii
1 Introdução	1
1.1 Considerações Iniciais	1
1.2 Objetivos do Trabalho	2
1.2.1 Objetivos Específicos	2
1.3 Justificativa	2
1.4 Estrutura da Monografia	3
2 Referencial Teórico	5
2.1 Acidentes de Trânsito	5
2.2 Mineração de Dados	6
2.3 Python	6
2.4 Firebase Realtime Database	6
2.5 Android	7
2.6 SDK do Maps para Android	7
3 Estudo da Arte	9
3.1 Desenvolvimento de um aplicativo Android utilizando a classe FlingAnimation para abordagem de conceitos de cinemática	9
3.2 Utilização do Google Maps para o georreferenciamento de dados do Sistema de Informações sobre Mortalidade no município do Rio de Janeiro, 2010-2012	10
3.3 A letalidade dos acidentes de trânsito nas rodovias federais brasileiras em 2016	12
3.4 Considerações Finais	13
4 Metodologia e Desenvolvimento	15
4.1 Bases de Dados	15
4.2 Seleção dos Dados	16
4.3 Armazenamento dos Dados	18

4.4	Implementação do Aplicativo Móvel	19
4.4.1	Diagrama de caso de uso	19
4.4.2	API Google Maps	19
4.4.3	Relatório de Acidentes	20
5	Resultados	21
5.1	Ícone do Aplicativo	21
5.2	Tela Inicial	21
5.3	Dialog Marcar Rota	22
5.4	Dialogs Condições Atuais e Relatório da Rota	22
6	Conclusões	25
6.1	Considerações Finais	25
6.2	Trabalhos Futuros	25

Lista de Figuras

3.1	Tela principal do aplicativo BCinemática – MRU & MRUV	10
3.2	Diagrama representando as três fases do procedimento de georreferenciamento .	11
4.1	Planilha dos acidentes referente ao ano de 2020	16
4.2	Planilha dos municípios brasileiros	16
4.3	Script para seleção de acidentes	17
4.4	Script para seleção de municípios	18
4.5	Estrutura do banco no Firebase	18
4.6	Estruturas dos registros de acidentes e municípios	19
4.7	Diagrama de caso de uso	19
4.8	Método para criação da URL	20
5.1	Ícone do aplicativo Viagem Segura	21
5.2	Tela inicial	22
5.3	Dialog Marcar Rota	22
5.4	Dialog Condições Atuais	23
5.5	Dialog Relatório da Rota	23

Lista de Siglas

PRF - *Polícia Rodoviária Federal*
API - *Application Programming Interface*
IBGE - *Instituto Brasileiro de Geografia e Estatística*
OMS - *Organização Mundial da Saúde*
JSON - *JavaScript Object Notation*
NoSQL - *Not only Structured Query Language*
SDK - *Software Development Kit*
SIM - *Sistema de Informações sobre Mortalidade*
CSV - *Comma-separated values*
SQL - *Structured Query Language*
UF - *Unidade Federativa*
IDE - *Integrated Development Environment*
XML - *eXtensible Markup Language*
HTTP - *Hypertext Transfer Protocol*

Resumo

SILVA, E. J. R. *Viagem Segura: Desenvolvimento de Aplicativo Android para Auxiliar na Redução de Acidentes de Trânsito em Rodovias de Minas Gerais*. Monografia (Graduação) — Centro Universitário de Formiga – Unifor-MG – Formiga, 2021.

Atualmente o Brasil conta com uma frota de mais de 100 milhões de veículos, o que tem acarretado um grande número em acidentes de trânsito. Com base nisso, esse trabalho busca contribuir para a redução desse número em rodovias de Minas Gerais através da criação de um aplicativo *Android*, utilizando técnicas de mineração de dados para selecionar dados relacionados a acidentes de trânsito providos pela Polícia Rodoviária Federal (PRF). A seleção de dados foi realizada através de *scripts* implementados em *Python*, e os dados selecionados foram armazenados na nuvem através do *Firebase*. O aplicativo ainda utiliza a *API Google Maps* para que o usuário trace sua rota, e gere um relatório dos acidentes.

Palavras-chave: Mineração de Dados, Acidentes de Trânsito, Android, API Google Maps

Abstract

SILVA, E. J. R. *Safe Travel: Android Application Development to Help Reduce Traffic Accidents on Minas Gerais Highways*. Monography (University Graduate) — Centro Universitário de Formiga – Unifor-MG – Formiga-MG, 2021.

Currently, Brazil has a fleet of more than 100 million vehicles, which has resulted in a large number of traffic accidents. Based on this, this work seeks to contribute to the reduction of this number in municipalities in Minas Gerais through the creation of an Android application, using data mining techniques to select data related to traffic accidents provided by the Polícia Rodoviária Federal (PRF). Data selection was performed through scripts implemented in Python, and selected data were stored in the cloud using Firebase. The application also uses the Google Maps API for the user to trace their route, and generate a report of accidents.

Keywords: Data Mining, Traffic Accidents, Android, Google Maps API

Introdução

1.1 Considerações Iniciais

De acordo com dados do Instituto Brasileiro de Geografia e Estatística (IBGE) o Brasil conta com uma frota de mais de 107,9 milhões de veículos, considerando todas as categorias (IBGE, 2020). O IBGE também aponta que aproximadamente 58 milhões são automóveis, e segundo o Sindicato Nacional da Indústria de Componentes para Veículos Automotores 46,2 milhões desses automóveis estão em circulação. (SINDIPECAS, 2020)

Como consequência do grande aumento na circulação de veículos, também é perceptível uma elevação em acidentes de trânsito. Segundo a Organização Mundial da Saúde (OMS), acidentes de trânsito resultam em aproximadamente 1,35 milhão de mortes a cada ano. Aponta-se que 93% das fatalidades mundiais nas estradas ocorrem em países de baixa e média renda, sendo que esses países contam com aproximadamente 60% dos veículos do mundo. (OMS, 2020)

Há vários motivos que podem contribuir à ocorrência de acidentes, sendo o principal deles o excesso de velocidade, que está diretamente relacionado à probabilidade de ocorrência de um acidente e à gravidade do mesmo. A cada 1% de aumento na velocidade média, há um aumento de 4% no risco de acidentes fatais. (OMS, 2020)

Segundo (RODY; JÚNIOR; SILVA, 2017), foram criados dispositivos que ajudam a manter o controle da velocidade em vias públicas, sejam eles físicos, como lombadas, ou eletrônicos, como radares. Outras ferramentas que também tem contribuído para redução de acidentes são aplicativos, como por exemplo o *Waze*, e o *MapLink*. (TRANSITO, 2019)

Porém, para a criação de um aplicativo é necessária uma boa fonte de informação, e se tratando de acidentes ocorridos em rodovias federais brasileiras, os dados abertos da Polícia Rodoviária Federal (PRF) são uma excelente opção. Os dados da PRF são disponibilizados anualmente, contendo os registros de acidentes ocorridos no ano anterior.

De acordo com (CASTRO; FERRARI, 2016), para lidar com grandes quantidades de dados,

é necessário a aplicação de técnicas que transformem, de maneira inteligente e automática, dados em informações úteis. Por essa razão, pesquisadores de diversas áreas se dedicam a estudar métodos para análise de dados, e é nesse contexto que surgiu a mineração de dados, que é um processo sistemático, interativo e iterativo, de preparação e extração de conhecimentos em grandes bases de dados.

Com o intuito de auxiliar a prevenção de acidentes, surge a ideia de um aplicativo utilizando da *API Google Maps*, abastecido com dados de acidentes automotivos em rodovias públicas, fornecidos pela PRF, no qual a partir de uma rota definida pelo usuário, os dados serão tratados e será exibido um relatório contendo as probabilidades de acidentes, levando em consideração o cenário durante a solicitação da rota, tais como as condições climáticas, o horário, e o dia da semana.

1.2 Objetivos do Trabalho

Desenvolver um aplicativo móvel para auxiliar na redução de acidentes de trânsito.

1.2.1 Objetivos Específicos

- Associar o local dos acidentes com as coordenadas das cidades de Minas Gerais;
- Filtrar os acidentes com base na rota definida pelo usuário;
- Exibir um relatório dos acidentes a partir da rota definida pelo usuário;

1.3 Justificativa

Segundo a OMS, acidentes de trânsito resultam em aproximadamente 1,35 milhão de mortes a cada ano, deixando ainda 20 a 50 milhões de pessoas com ferimentos não fatais. Em relação a crianças e jovens de 5 a 29 anos, lesões causadas por acidentes de trânsito são a principal causa de morte, sendo que 73% de todas as mortes no trânsito são com vítimas do sexo masculino. (OMS, 2020)

Além de causar graves efeitos psicológicos sobre a vítima e sua família, acidentes de trânsito também implicam em um enorme gasto nas economias nacionais, pois custam aos países 3% de seu produto interno bruto anual. O Instituto de Pesquisa Econômica Aplicada aponta que o Brasil gasta por volta de 132 bilhões por ano com acidentes de transporte. (IPEA, 2020)

Segundo informações da Pesquisa Nacional por Amostra de Domicílios Contínua - Tecnologia da Informação e Comunicação, aparelhos celulares, também chamados de *smartphones*, são o principal meio de acesso à internet no país. A pesquisa aponta que 98,6% das pessoas com 10 ou mais anos de idade acessaram a internet pelo celular. (IBGE, 2019)

Portanto, como tem se tornado cada vez mais comum o uso de *smartphones*, muitas pessoas tem acesso à uma grande variedade de ferramentas contidas no mesmo. Dentre essas ferramentas vale a pena destacar o *Google Maps*, que é um serviço mantido pelo *Google* desde 8 de fevereiro de 2005, que possibilita a visualização de mapas e rotas. (JUNIOR, 2018)

Acidentes estão propensos a acontecerem a qualquer momento e em qualquer lugar, porém algumas rodovias podem apresentar condições que contribuam para a ocorrência do mesmo. Contudo, se o condutor receber a informação sobre a ocorrência de algumas dessas condições

consideradas problemáticas em sua rota, tem-se a expectativa de reduzir a chance de ocorrer alguma fatalidade. E como o uso de *smartphones* tem crescido cada vez mais, a criação de um aplicativo aparenta ser uma alternativa viável.

1.4 Estrutura da Monografia

A estrutura da monografia foi definida da seguinte maneira:

No capítulo 1 foram apresentadas as considerações iniciais, os objetivos e a justificativa para o desenvolvimento do trabalho. No capítulo 2, está presente o referencial teórico, o qual descreve as tecnologias utilizadas no trabalho. No capítulo 3 está contido o estudo da arte, que apresenta alguns trabalhos com temas relacionados ao trabalho em questão, e que serviram como referência para o desenvolvimento do mesmo. O capítulo 4 mostra a metodologia utilizada para a coleta de dados e para o desenvolvimento do aplicativo. No capítulo 5 são apresentados os resultados obtidos. Por fim, no capítulo 6 está presente a conclusão, e sugestões para trabalhos futuros.

Referencial Teórico

Neste capítulo, serão apresentadas as ferramentas e tecnologias utilizadas para o desenvolvimento do projeto.

2.1 Acidentes de Trânsito

De acordo com o Departamento Nacional de Trânsito, acidentes de trânsito são eventos que ocorrem de maneira não intencionais, e que envolva pelo menos um veículo motorizado ou não (ciclistas, motociclistas, veículos em geral, transporte individual, transporte coletivo, transporte de carga), bem como eventos que envolvam atropelamento de pedestres. (MASSAÚ; ROSA, 2016)

Todos os anos, aproximadamente 1,3 milhão de pessoas morrem devido a acidentes de trânsito, sendo que 90% dessas mortes ocorrem em países com baixa e média renda, e quanto às pessoas que sofrem lesões não fatais, o número varia entre 20 e 50 milhões. Além disso, acidentes de trânsito são a principal causa de morte de crianças e jovens entre 5 a 29 anos. (OMS, 2021)

Segundo a (OMS, 2021), dentre os principais fatores de risco em acidentes de trânsito estão a abordagem do Sistema Seguro, que acomoda o erro humano, o excesso de velocidade, dirigir sob a influência de álcool e outras substâncias psicoativas, o não uso de capacetes para motociclistas, cintos de segurança e sistemas de retenção para crianças, direção distraída, infraestrutura rodoviária insegura, veículos inseguros, cuidados pós-acidente inadequados e a aplicação inadequada das leis de trânsito.

Anualmente a PRF disponibiliza microdados de acidentes ocorridos em rodovias brasileiras através dos Dados Abertos.

2.2 Mineração de Dados

A mineração de dados surgiu em meados da década de 1990 como uma área de pesquisa e aplicação independente, ela é responsável pela análise das grandes massas de dados e tem ganhado evidência nos últimos anos após a criação do termo *Big Data*. (CASTRO; FERRARI, 2016)

O *Big Data* é um fenômeno que está associado a produção de grandes volumes de dados, de modo que as tecnologias não conseguiam processá-los. Então a mineração de dados surgiu com o objetivo de transformar o dado, que é uma matéria bruta, em informação e conhecimento. (AMARAL, 2016)

A mineração de dados tem sido aplicada nas mais diversas áreas, como por exemplo a avaliação da reputação de empresas em redes sociais, aplicações em marketing, medicina, educação, finanças, dentre outros. (AMARAL, 2016)

Para a aplicação da mineração de dados, tem se usado muito a linguagem de programação *Python*, que é uma linguagem de sintaxe considerada simples.

2.3 Python

Python é uma linguagem de programação desenvolvida pelo holandês Guido Van Rossum, que desenvolveu a linguagem com intuito de resolver grandes falhas presenciadas por ele em sistemas que utilizavam a linguagem *C*. O *Python* é extremamente simples e robusto, e devido a sua arquitetura bem projetada, fornece um bom desempenho e legibilidade do código. (SILVA; SILVA, 2019)

A linguagem *Python* não possui um propósito específico, ela oferece suporte para diversas áreas, como: *desktops*, desenvolvimento *web*, aplicações *mobile*, geoprocessamento, processamento de imagens, robótica, *Data Science*, dentre outras. E além disso, a linguagem já vem instalada de fábrica em diversos sistemas operacionais, tais como *AmigaOS*, *netBSD*, *MacOS* e todas as distribuições *Linux*. (SILVA; SILVA, 2019)

Dentre as principais aplicações do *Python* está o *Scripting*, que geralmente é o nome dado a pequenos programas que foram projetados com objetivo de automatizar tarefas. A sintaxe relativamente simples e facilidade de escrever, torna o Python uma linguagem adequada para esse tipo de tarefa. (PROGRAMATHOR, 2018)

Neste projeto, o *Python* foi utilizado para a implementação dos *scripts*, que possibilitou a filtragem dos microdados fornecidos pela PRF. Os dados resultantes dos *scripts* foram armazenados em um banco na nuvem.

2.4 Firebase Realtime Database

O *Firebase Realtime Database* é um banco de dados hospedado em nuvem, onde os dados são armazenados como *JSON* e sincronizados em tempo real para cada cliente conectado. O *Firebase* é um banco de dados *NoSQL*, o que faz com que ele tenha otimizações e funcionalidades diferentes de um banco de dados relacional. (FIREBASE, 2021)

No *Firebase*, é possível conceder acesso seguro ao banco de dados diretamente do código

do cliente, o que permite a criação de aplicativos avançados e colaborativos. Os dados são mantidos localmente, de modo que mesmo que o cliente esteja *off-line*, os eventos em tempo real continuam sendo acionados, e quando o dispositivo recupera a conexão com a internet, as alterações feitas são sincronizadas com as atualizações remotas, o que proporciona uma experiência responsiva ao usuário final. (FIREBASE, 2021)

O *Firebase* fornece uma linguagem de regras flexíveis baseadas em expressões, que são denominadas como regras de segurança, e são elas que definem como os dados são estruturados e quando podem ser lidos e gravados. Por meio do *Firebase Authentication* os desenvolvedores podem definir quais usuários tem acesso a determinados dados, como esses dados podem ser acessados. (FIREBASE, 2021)

O *Firebase* foi utilizado neste trabalho, para que o aplicativo implementado em Android pudesse consultar os dados coletados dos acidentes de trânsito.

2.5 Android

O *Android* é um sistema operacional que foi desenvolvido em 2003 por Andy Rubin, Rich Miner, Nick Sears e Chris White. A ideia inicial do *Android* era ser um sistema gratuito para o público e de fácil compreensão para os programadores. Os criadores estabeleceram como base de utilização o *kernel* do sistema operacional *Linux*, e a princípio o foco do grupo era o ramo de câmeras digitais, porém devido à escassez no mercado, partiram para o mercado dos dispositivos móveis. (LIMA, 2017)

Os aplicativos *Android* são desenvolvidos através da linguagem *Java*, e a escolha dessa linguagem foi lógica para a plataforma, pois é poderosa, gratuita, de código-fonte aberto e milhões de desenvolvedores já a conhecem. A linguagem *Java* é orientada a objetos, e contém uma grande variedade de bibliotecas que ajudam a desenvolver aplicativos poderosos de forma rápida. (DEITEL; DEITEL; DEITEL, 2015)

No ano de 2005, o *Google* com interesse de entrar no mercado dos smartphones, comprou a *Android Inc* juntamente com sua equipe de desenvolvedores. Após a compra, o *Google* iniciou o projeto *Google Mobile Division*, e manteve a ideologia de um sistema totalmente gratuito e de fácil entendimento, tendo também parte do seu código sob a Licença Apache de código aberto. Dessa forma, companhias que tinham dificuldade em desenvolver seu próprio sistema operacional, só precisavam baixar o código e fazer suas próprias mudanças. (LIMA, 2017)

2.6 SDK do Maps para Android

O *SDK* do *Maps* para *Android*, é uma ferramenta que permite adicionar mapas em aplicativos *Android*, o que inclui aplicativos *Wear OS* usando dados, exibição de mapa e respostas de gestos no mapa do *Google Maps*. Com ele também é possível fornecer outras informações sobre locais, além de permitir que o usuário interaja adicionando marcadores, polígonos e sobreposições em seu mapa. O *SDK* do *Maps* aceita as linguagens de programação *Kotlin* e *Java*, e disponibiliza outras bibliotecas e extensões para técnicas de programação e recursos avançados. (DEVELOPERS, 2021b)

Para utilizar a *SDK* do *Maps* é obrigatório o uso do *Android Studio*, que é o ambiente de desenvolvimento integrado oficial para o desenvolvimento de aplicativos *Android*, sendo baseado no *IntelliJ IDEA*. Além do editor de código e das ferramentas de desenvolvedor avançadas presentes no *IntelliJ*, o *Android Studio* conta com alguns outros recursos para aumentar a produtividade na criação de aplicativos, como um sistema de compilação flexível baseado em *Gradle*, um emulador rápido com inúmeros recursos, um ambiente unificado que possibilita o desenvolvimento para todos os dispositivos *Android*, dentre outros. (DEVELOPERS, 2021a)

Para usar o *SDK* do *Maps* para *Android*, é necessário um projeto com uma conta de faturamento, além de estar com esse *SDK* ativado. E para executar um aplicativo que usa o *SDK*, é necessário implantá-lo em um dispositivo que seja compatível, ou através de um dispositivo virtual com base no *Android* 4.0 ou uma versão mais recente que contenha as *APIs* do *Google*, por meio do *Android Emulator*. (DEVELOPERS, 2021b)

Estudo da Arte

3.1 Desenvolvimento de um aplicativo Android utilizando a classe *FlingAnimation* para abordagem de conceitos de cinemática

Em trabalho apresentado por (JÚNIOR; SIMÕES, 2021), foi desenvolvido um aplicativo *Android* utilizando a classe *FlingAnimation* com objetivo de abordar conceitos de cinemática. O aplicativo foi criado através do ambiente de desenvolvimento *Android Studio*, utilizando a linguagem de programação *Java*.

O aplicativo apresenta a trajetória retilínea de uma bola, que é tratada na simulação como um objeto puntiforme, onde ao clicar no botão Iniciar, é iniciado o movimento da bola. Além disso, o aplicativo também apresenta uma caixa de seleção onde se pode definir quatro opções de desaceleração da bola, sendo as opções: 0 mm/s^2 , 10 mm/s^2 , 20 mm/s^2 e 30 mm/s^2 , e também dois campos onde são apresentados os valores de variação de espaço e tempo. A tela principal do aplicativo pode ser vista na Figura 3.1.

Figura 3.1: Tela principal do aplicativo BCinemática – MRU & MRUV



Fonte: (JÚNIOR; SIMÕES, 2021)

A proposta do aplicativo, é que estudantes explorem e compreendam suas funcionalidades, inicialmente por uma percepção visual. Através do uso do aplicativo, os estudantes podem construir tabelas de dados de espaço em função do tempo, e através disso desenvolver habilidades de aquisição e tabulação de dados obtidos.

Ao final do trabalho, os autores chegaram à conclusão de que é possível utilizar a classe *FlingAnimation* para o desenvolvimento de aplicativos *Android* para a abordagem de conceitos de cinemática no ensino de Física. Segundo eles, o uso do aplicativo demonstrou ser consistente, e apresentou potencial, e que apesar de a classe *FlingAnimation* não ter sido desenvolvida especificamente para abordar conceitos de cinemática, os métodos disponíveis são aceitáveis em aplicações pontuais.

3.2 Utilização do Google Maps para o georreferenciamento de dados do Sistema de Informações sobre Mortalidade no município do Rio de Janeiro, 2010-2012

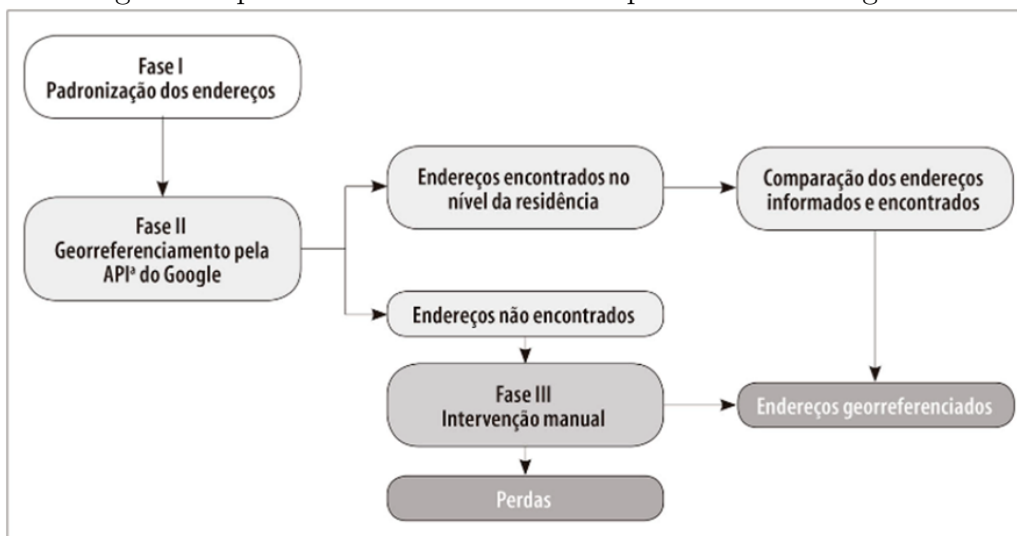
Em trabalho apresentado por (SILVEIRA; OLIVEIRA; JUNGER, 2017), foi implementada uma aplicação baseada em *software* livre, com o objetivo de auxiliar no georreferenciamento de dados do Sistema de Informações sobre Mortalidade (SIM) no município do Rio de Janeiro. O trabalho foi dividido em três etapas: a padronização dos endereços, o georreferenciamento por meio do *Google Maps* e uma intervenção manual.

A padronização dos endereços foi programada em *R*, onde buscou corrigir erros frequentes, remover caracteres estranhos e padronizar componentes relacionados aos tipos e aos títulos dos logradouros.

Quanto a parte do georreferenciamento foi feita através da *API* do *Google*, acessada por meio de um *script* em *R*, onde foi feita uma verificação de qualidade, na qual os endereços encontrados pelo *Google* receberam o mesmo tratamento dos informados. Logo após foi feita uma comparação, onde os endereços não correspondentes foram transferidos para a intervenção manual, juntamente com os endereços não localizados.

Por fim, na etapa manual os erros ortográficos remanescentes foram corrigidos e novamente submetidos ao georreferenciamento automático. Essa etapa de verificação utilizou bases de logradouros e de setores censitários providas pelo IBGE, e em casos mais complexos foram realizadas pesquisas manuais por meio do *Google Street View*, endereços sem número foram georreferenciados nos pontos médios dos logradouros, e quando não contido o endereço foi considerado perda. A seguir pode ser visto o diagrama representando as três fases:

Figura 3.2: Diagrama representando as três fases do procedimento de georreferenciamento



Fonte: (SILVEIRA; OLIVEIRA; JUNGER, 2017)

Foram submetidos 26.081 endereços ao procedimento, onde 18.646 (71,5%) foram georreferenciados automaticamente, os 7.435 restantes passaram pela intervenção manual, onde 5.250 (70,6%) foram recuperados, totalizando ao final 91,6% endereços georreferenciados.

Ao fim, os autores concluíram que o procedimento apresentou alta proporção de acertos automáticos, e apesar de demandar maior tempo, a intervenção manual reduziu consideravelmente as perdas. Eles também apontam que o desempenho da ferramenta depende da qualidade dos dados e do endereçamento municipal, onde quanto maior o detalhamento do endereço, menores serão as perdas, e que sua incorporação em serviços de saúde demanda pessoal com habilidades em informática e disponibilidade para a busca manual.

3.3 A letalidade dos acidentes de trânsito nas rodovias federais brasileiras em 2016

Com o objetivo de identificar os fatores associados à letalidade dos acidentes de trânsito nas rodovias federais brasileiras em 2016, foi feita uma análise por (JUNIOR; BERTHO; VEIGA, 2019), onde levaram em consideração características das vítimas e informações sobre o contexto de ocorrência desses eventos.

Para realizar a análise, foi utilizada a base de dados de acidentes da PRF referente ao ano de 2016, base essa que contém registros de acidentes ocorridos em rodovias federais e que é disponibilizada para download no site da própria instituição.

Para que os dados pudessem atender às necessidades da pesquisa, foram necessários alguns ajustes e tratamentos. Foram retirados registros onde os indivíduos não haviam sido expostos ao risco de morte, o que resultou em um corte de 51,9% dos registros. Foram excluídos também registros que continham dados inválidos para alguma das variáveis que seriam utilizadas na análise, ou que tenham sido preenchidas com “ignorado”, “não identificado”, “NA” ou “indefinido”, e registros com indivíduos com mais de 100 anos de idade ou condutores com menos de dez anos de idade, por serem possíveis erros de preenchimento. Ao final restaram 35,7% do total de registros, que incluem pessoas que se feriram em acidentes, independente da gravidade.

Foram utilizadas 13 das 28 variáveis contidas nos dados da PRF, que são: sexo da vítima, idade da vítima, tipo de envolvimento da vítima no acidente, tipo de veículo, tipo de acidente, tipo de pista, tipo de traçado da via no local do acidente, dia da semana, hora do acidente, uso do solo no local do acidente, unidade da federação, e estado físico da vítima após o acidente. Ainda foram feitas algumas modificações nas 13 variáveis selecionadas, em alguns casos por questão de simplificação e, em outros, para que atendessem melhor ao propósito da análise, para isso foi utilizada a função *glm* do *software R*, função essa que utiliza do método da máxima verossimilhança para estimar os parâmetros do modelo.

As variáveis selecionadas foram inseridas uma a uma em um modelo binomial de regressão logística nulo, e dependendo da sua significância era mantida ou retirada do modelo, e uma nova variável era inserida no passo seguinte. Esse procedimento foi repetido até que todas as variáveis tivessem sido testadas. Após a realização desse procedimento, chegou-se ao modelo final, que foi submetido ao teste de bondade de ajuste de Hosmer e Lemeshow (1980), onde não foram encontradas evidências estatísticas de que o modelo estivesse mal especificado ou mal ajustado aos dados.

Ao final da análise, os autores obtiveram resultados já consolidados na literatura. Como por exemplo, as chances de um acidente ser letal são maiores para homens em relação a mulheres, acidentes ocorridos aos sábados e domingos apresentam maior chance de letalidade em comparação à segunda-feira, e quanto a fase do dia, a letalidade tende a ser maior durante a noite e madrugada quando comparados à manhã.

3.4 Considerações Finais

Através do trabalho de (JÚNIOR; SIMÕES, 2021), foi possível concluir que a utilização de aplicativos *Android* pode trazer muitos benefícios, pois é uma plataforma de fácil acesso e que apresenta diversas funcionalidades. Uma das funcionalidades presentes no *Android*, é a *API Google Maps*, que pode ser vista no trabalho de (SILVEIRA; OLIVEIRA; JUNGER, 2017), onde através dela, os autores fizeram georreferenciamentos de dados do SIM no município do Rio de Janeiro. E se tratando de acidentes de trânsito, uma boa fonte de informações são os dados providos pela PRF, que foram utilizados no trabalho de (JUNIOR; BERTHO; VEIGA, 2019).

Portanto, o objetivo desse trabalho, é contribuir para a redução de acidentes de trânsito através da utilização desses três itens, onde será implementado um aplicativo *Android*, utilizando a *API Google Maps*, e abastecido com dados de acidentes de trânsito providos pela PRF.

Metodologia e Desenvolvimento

Este capítulo tem como objetivo apresentar todos os materiais e métodos utilizados para construção do aplicativo, como por exemplo a tabela dos acidentes de trânsito fornecida pela PRF, a tabela que contém dados de todos os municípios do Brasil adquirida através de um repositório no *GitHub*, bem como os *scripts* que foram desenvolvidos para a filtragem e armazenamento dos dados, e o passo a passo para a implementação do aplicativo. Ambas as bases de dados foram filtradas a registros pertencentes a Minas Gerais.

4.1 Bases de Dados

Segundo (DIGITAL, 2017), bases de dados são um conjunto de dados interrelacionados, organizados de forma a permitir a recuperação da informação. Elas têm como objetivo fornecer informação atualizada, precisa e confiável de acordo com a demanda que usuário necessita.

Para o desenvolvimento deste trabalho foram utilizadas duas bases de dados, onde uma apresentava dados sobre todos os municípios brasileiros, onde foi possível obter as coordenadas geográficas de todos os municípios de Minas Gerais, e a outra continha dados sobre os acidentes de trânsito ocorridos no Brasil em 2020.

A base de dados que contém os acidentes de trânsito é disponibilizada pela PRF, que é uma instituição policial ostensiva brasileira, subordinada ao Ministério da Justiça e Segurança Pública, cuja principal função é garantir a segurança com cidadania nas rodovias federais e em áreas de interesse da União. A cada ano a PRF disponibiliza novas planilhas em formato *CSV*, onde são apresentados os acidentes registrados no ano anterior, além de disponibilizar também um dicionário de dados que tem como objetivo fornecer uma sucinta descrição das variáveis presentes na planilha, auxiliando o usuário na análise as informações. A planilha referente ao ano de 2020 pode ser vista na Figura 4.1.

Quanto aos dados referentes aos municípios brasileiros, foram recuperados através de um

Figura 4.1: Planilha dos acidentes referente ao ano de 2020

id	data_inve	dia_semai	horario	uf	br	km	municipio	causa_aci	tipo_acid	classificac	fase_dia	sentido_v	condicao	tipo_pista	tracado_v	uso_solo	peessoas	mortos	feridos_le	
260031	#####	quarta	01:00:00	TO		153	678,1 GURUPI	Animais n	Atropelan	Com VÃ	ti Plena	Noi Crescente	Nublado	Simples	Reta	NÃ	5	0	0	
260036	#####	quarta	01:00:00	RJ		116	178 NOVA IGU	Falta de A	Atropelan	Com VÃ	ti Plena	Noi Decrescer	CÃ	u Cla Dupla	Reta	Sim	2	1	0	
260037	#####	quarta	01:52:00	SC		101	206,9 SAO JOSE	IngestÃ	ColisÃ	o l Com VÃ	ti Plena	Noi Crescente	Nublado	Simples	Reta	Sim	2	0	1	
260038	#####	quarta	01:15:00	RO		364	236 CACOAL	Velocidad SaÃ	da de Com VÃ	ti Plena	Noi Decrescer	Nublado	Simples	Reta	Sim	1	0	0		
260039	#####	quarta	01:30:00	PR		153	360 REBOUCA	IngestÃ	Capotame	Com VÃ	ti Plena	Noi Crescente	Garoa/Chi	Simples	Curva	NÃ	6	0	2	
260040	#####	quarta	00:40:00	RO		319	64 PORTO VE	IngestÃ	ColisÃ	o t Sem VÃ	ti Plena	Noi Decrescer	CÃ	u Cla Dupla	Reta	Sim	2	0	0	
260041	#####	quarta	02:20:00	SC		101	89,9 BARRA VE	Falta de A	ColisÃ	o c Sem VÃ	ti Plena	Noi Crescente	Garoa/Chi	Dupla	Reta	Sim	1	0	0	
260042	#####	quarta	02:15:00	PR		369	155 LONDRIN	Desobedi,	ColisÃ	o t Com VÃ	ti Plena	Noi Crescente	CÃ	u Cla MÃ	tipla	NÃ	3	0	1	
260043	#####	quarta	02:30:00	AL		104	37 UNIAO DC	Animais n	Atropelan	Com VÃ	ti Plena	Noi Decrescer	Nublado	Simples	Reta	NÃ	1	1	0	
260044	#####	quarta	00:20:00	RS		290	97 PORTO AL	Falta de A	Tombame	Com VÃ	ti Plena	Noi Decrescer	CÃ	u Cla Dupla	Reta	NÃ	2	0	2	
260045	#####	quarta	02:50:00	SC		470	116,3 IBIRAMA	Condutor	ColisÃ	o f Com VÃ	ti Plena	Noi Decrescer	Nublado	Simples	Curva	Sim	4	0	2	
260046	#####	quarta	02:15:00	SC		282	509 XANXERE	Animais n	SaÃ	da de Com VÃ	ti Plena	Noi Decrescer	CÃ	u Cla Simples	Reta	NÃ	3	0	0	
260050	#####	quarta	03:40:00	PE		101	49,5 ABREU E L	Falta de A	ColisÃ	o l Com VÃ	ti Plena	Noi Crescente	CÃ	u Cla MÃ	tipla	Reta	Sim	4	0	0
260051	#####	quarta	02:30:00	MT		364	265,3 SAO PEDR	Falta de A	Queda de	Com VÃ	ti Plena	Noi Crescente	CÃ	u Cla Simples	NÃ	Info Sim	2	0	2	
260052	#####	quarta	03:10:00	BA		101	174 FEIRA DE S	Animais n	Atropelan	Com VÃ	ti Plena	Noi Crescente	Nublado	Simples	Reta	NÃ	2	0	1	
260053	#####	quarta	04:00:00	SC		101	132 BALNEARI	Falta de A	Atropelan	Com VÃ	ti Plena	Noi Crescente	CÃ	u Cla Dupla	Curva	Sim	2	1	0	
260054	#####	quarta	00:05:00	AP		156	406 FERREIRA	Animais n	Atropelan	Sem VÃ	ti Plena	Noi Crescente	CÃ	u Cla Simples	Reta	NÃ	1	0	0	
260055	#####	quarta	03:20:00	GO		60	164,8 GOIANIA	Falta de A	SaÃ	da de Com VÃ	ti Amanheci	Crescente	Nublado	Dupla	Reta	NÃ	4	0	4	
260057	#####	quarta	04:55:00	DF		20	21,8 BRASILIA	IngestÃ	Atropelan	Com VÃ	ti Amanheci	Crescente	Garoa/Chi	Dupla	Reta	Sim	3	0	2	
260060	#####	quarta	05:00:00	SP		116	99,6 PINDAMO	Falta de A	ColisÃ	o c Com VÃ	ti Amanheci	Crescente	CÃ	u Cla Dupla	Reta	NÃ	2	0	2	
260061	#####	quarta	03:00:00	BA		324	616 SALVADOI	Falta de A	Queda de	Com VÃ	ti Plena	Noi Decrescer	Nublado	Dupla	Reta	Sim	2	0	2	

Fonte : PRF (2020)

repositório no *GitHub*, onde seus colaboradores disponibilizaram os dados nos seguintes formatos: *SQL*, *CSV* e *JSON*. Assim como nos dados da PRF, foi utilizado o formato *CSV*, onde a planilha pode ser visualizada na Figura 4.2.

Figura 4.2: Planilha dos municípios brasileiros

codigo_ibge	nome	latitude	longitude	capital	codigo_uf	siafi_id	ddd	fuso_horario
5200050	Abadia de Goiás	-16.7573	-49.4412	0,52	1050	62	America/Sao_Paulo	
3100104	Abadia dos Dourados	-18.4831	-47.3916	0,31	4001	34	America/Sao_Paulo	
5200100	AbadiÃ	-16.197	-48.7057	0,52	9201	62	America/Sao_Paulo	
3100203	AbaetÃ	-19.1551	-45.4444	0,31	4003	37	America/Sao_Paulo	
1500107	Abaetetuba	-1.72183	-48.8788	0,15	0401	91	America/Sao_Paulo	
2300101	Abaiara	-7.34588	-39.0416	0,23	1301	88	America/Sao_Paulo	
2900108	AbaÃ	-13.2488	-41.6619	0,29	3301	77	America/Sao_Paulo	
2900207	AbarÃ	-8.72073	-39.1162	0,29	3303	75	America/Sao_Paulo	
4100103	AbatiÃ	-23.3049	-50.3133	0,41	7401	43	America/Sao_Paulo	
4200051	Abdon Batista	-27.6126	-51.0233	0,42	9939	49	America/Sao_Paulo	
1500131	Abel Figueiredo	-4.95333	-48.3933	0,15	0375	94	America/Sao_Paulo	
4200101	Abelardo Luz	-26.5716	-52.3229	0,42	8001	49	America/Sao_Paulo	
3100302	Abre Campo	-20.2996	-42.4743	0,31	4005	31	America/Sao_Paulo	
2600054	Abreu e Lima	-7.90072	-34.8984	0,26	2631	81	America/Sao_Paulo	
1700251	AbreulÃ	-9.62101	-49.1518	0,17	0337	63	America/Sao_Paulo	
3100401	Acaiaca	-20.359	-43.1439	0,31	4007	31	America/Sao_Paulo	
2100055	AÃ	-4.94714	-47.5004	0,21	0961	99	America/Sao_Paulo	
2900306	Acajutiba	-11.6575	-38.0197	0,29	3305	75	America/Sao_Paulo	
1500206	AcarÃ	-1.95383	-48.1985	0,15	0403	91	America/Sao_Paulo	
2300150	Acarape	-4.22083	-38.7055	0,23	1231	85	America/Sao_Paulo	

Fonte : Github (2020)

4.2 Seleção dos Dados

Para que os dados contidos nas duas bases de dados apresentadas anteriormente pudessem ser utilizadas de maneira mais objetiva, foram implementados *scripts*, que percorreram os arquivos originais, e filtrou apenas as colunas que continham informações que seriam utilizadas no trabalho. As informações filtradas foram armazenadas em um arquivo de formato *JSON*.

Como pode ser visto na Figura 4.3, o script para a seleção de dados da planilha de acidentes, foi realizado com auxílio da biblioteca *csv* do *Python*. Como os dados eram divididos em colunas, foi possível acessá-los apenas com um laço de repetição, onde foram filtrados os dados em que o valor referente a UF era igual a ‘MG’. Em seguida foram armazenados no arquivo *JSON* os dados referentes as colunas dia_semana, horário, condicao_meteorologica, latitude e longitude.

Figura 4.3: Script para seleção de acidentes

```
import csv

with open('datatran2020.csv', encoding='utf-8') as arq:
    planilha = csv.reader(arq, delimiter=',')

    acidentes = open('acidentes.json', 'w')
    acidentes.write('{"acidentes":{')
    id = 0

    for l in planilha:
        if(l[4] == 'MG'):
            id = id + 1
            if(id != 1):
                acidentes.write(',')
            acidentes.write('"' + str(id) + "':{')
            acidentes.write("diaSemana": "' + l[2] + '",')
            acidentes.write("horario": "' + l[3] + '",')
            acidentes.write("condicaoMeteorologica": "' + l[13] + '",')
            acidentes.write("latitude": "' + l[25] + '",')
            acidentes.write("longitude": "' + l[26] + '",')
            acidentes.write('}')

    acidentes.write('}}}')
    acidentes.close()
```

Fonte : Autoria própria

Já o *script* referente a seleção dos municípios, foi feito de maneira semelhante ao de acidentes, porém como os dados estavam agrupados em uma única coluna, foi necessário a utilização da função *split* para separá-los e armazená-los em diferentes posições de um *array*. Nesse caso, também foram filtrados apenas os municípios referentes a Minas Gerais, cujo valor referente a coluna código_uf era igual a ‘31’, e foram armazenados no arquivo *JSON* os valores referentes as colunas nome, latitude e longitude. O *script* pode ser visualizado através da Figura 4.4.

Figura 4.4: Script para seleção de municípios

```
import csv

with open('municipios.csv', encoding='utf-8') as arq:
    planilha = csv.reader(arq, delimiter=';')

    municipios = open('municipios.json', 'w')
    municipios.write('{"municipios":{'
    id = 0

    for l in planilha:
        linha = l[0].split(',')

        if(linha[5] == '31'):
            id = id + 1
            if(id != 1):
                municipios.write(',')
            municipios.write('"' + str(id) + "':{'
            municipios.write('"nome": "' + linha[1] + "',')
            municipios.write('"latitude": "' + linha[2] + "',')
            municipios.write('"longitude": "' + linha[3] + "'')
            municipios.write('}')

    municipios.write('}')
    municipios.close()
```

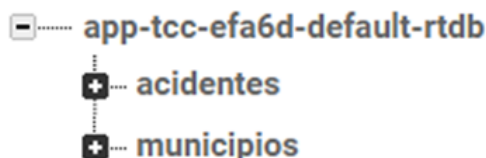
Fonte : Autoria própria

4.3 Armazenamento dos Dados

A etapa que sucedeu a seleção dos dados, foi o armazenamento no banco de dados. O banco escolhido para o aplicativo foi o *Firebase Realtime Database*, que é um banco de dados em nuvem. Essa escolha se deu devido a uma avaliação feita, onde se pôde concluir que apesar de os dados serem estáticos, ocupariam um grande espaço de armazenamento, e que se fosse armazenado em um banco local poderia causar uma queda de performance no aplicativo.

Os dados foram transferidos para o banco através da ferramenta de importação de *JSON* presente no console do *Firebase*. Como essa ferramenta sobrescreve todos os registros do banco com os dados importados, o conteúdo dos dois arquivos *JSON* resultantes dos *scripts* de seleção foram copiados para um único arquivo. A estrutura final do banco pode ser vista através da Figura 4.5.

Figura 4.5: Estrutura do banco no Firebase



Fonte : Autoria própria

No total, foram armazenados 8373 registros de acidentes no nó de acidentes, e 853 registros no nó de municípios. As estruturas dos registros de acidentes e municípios podem ser vistas na Figura 4.6.

Figura 4.6: Estruturas dos registros de acidentes e municípios



Fonte : Autoria própria

4.4 Implementação do Aplicativo Móvel

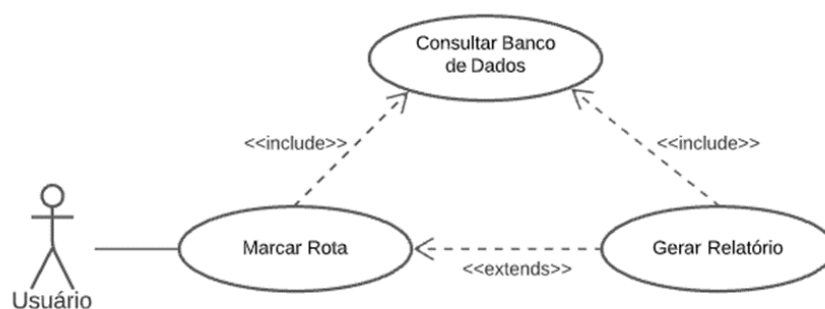
A implementação do aplicativo Viagem Segura foi realizada através do *Android Studio*, que é a *IDE* oficial do *Android*, utilizando a linguagem de programação *Java*. O *Android* utiliza para a criação de suas interfaces a tecnologia *XML*, e conta com um sistema de compilação flexível baseado em *Gradle*.

4.4.1 Diagrama de caso de uso

Segundo (JÚNIOR, 2020), um modelo de caso de uso é a representação das funcionalidades externamente observáveis de um sistema, ou de elementos externos que ele interaja. Ou seja, ele é representação visual das interações entre o usuário e as funcionalidades do sistema.

Como pode ser visto na Figura 4.7, o diagrama do aplicativo Viagem Segura contém três casos de usos e um único ator. O ator (Usuário) interage diretamente com o caso de uso Marcar Rota, e após essa interação ele tem a opção de Gerar Relatório. Ambos os casos de uso realizam uma consulta no banco de dados.

Figura 4.7: Diagrama de caso de uso



Fonte : Autoria própria

4.4.2 API Google Maps

O foco principal do aplicativo está na *API* do *Google Maps*, que possibilitou ao usuário marcar e visualizar sua rota. Para utilizá-la foi necessário criar uma conta de faturamento, pois cada uma das *APIs* tem seu preço de utilização, variando conforme a quantidade de requisições feitas. O *Google* disponibiliza US\$300 por um período de teste de 90 dias, e ao final

desse período, ainda são disponibilizados US\$200 de créditos mensais para serem utilizados em requisições de *APIs* na plataforma, onde cada *API* tem seu preço, ainda podendo variar pela quantidade de solicitações feitas.

Os pontos de origem e destino da rota foram definidos pelo usuário, e logo após foram recuperadas do banco de dados da aplicação as informações sobre latitude e longitude dos pontos, para que assim eles pudessem ser marcados no mapa.

A marcação de rota foi realizada através *API Directions* presente no *Google Maps*, que ao ser solicitada através de uma solicitação *HTTP* retorna a rota formatada em *JSON* ou *XML*. O método utilizado para criar a *URL* da solicitação pode ser visto na Figura 4.8.

Figura 4.8: Método para criação da URL

```
private String getUrl(LatLng origin, LatLng dest, String directionMode) {
    String str_origin = "origin=" + origin.latitude + "," + origin.longitude;
    String str_dest = "destination=" + dest.latitude + "," + dest.longitude;
    String mode = "mode=" + directionMode;
    String parameters = str_origin + "&" + str_dest + "&" + mode;
    String output = "json";
    String url = "https://maps.googleapis.com/maps/api/directions/" + output + "?" + parameters + "&key=" + getString(R.string.google_maps_key);
    return url;
}
```

Fonte : Autoria própria

Esse método recebe dois parâmetros, que são as coordenadas geográficas dos pontos de origem e destino da rota. Já dentro do método são definidos outros dois parâmetros, sendo o primeiro deles o modo de direção, que para esse trabalho foi definido sempre como “*driving*”, o que indica que a rota marcada será para um veículo, e o segundo o tipo de retorno da solicitação, que foi definido como “*json*”. Ao final todos os parâmetros são tratados e concatenados a *URL* da *API*.

4.4.3 Relatório de Acidentes

Após a marcação da rota foi implementado o relatório dos acidentes, que através de uma consulta no banco de dados buscou registros de acidentes que estivessem presentes na rota.

Ao relatório ser solicitado pelo usuário, através das classes *LocalTime* e *LocalDateTime* do *Java* o próprio aplicativo irá definir a fase do dia e o dia da semana referente a solicitação, sendo necessário que o usuário informe apenas a condição climática. O atributo fase do dia foi dividido em quatro categorias, sendo: madrugada (00:01 a 06:00), manhã (06:01 a 12:00), tarde (12:01 a 18:00), e noite (18:01 a 00:00).

Após os dados referentes a rota serem obtidos, são exibidos no relatório: a quantidade total de acidentes, a quantidade de acidentes ocorridos nas condições especificadas, e a porcentagem relativa aos dois itens anteriores.

Resultados

Neste capítulo estão presentes as telas do aplicativo Viagem Segura, juntamente com suas funcionalidades.

5.1 Ícone do Aplicativo

Na Figura 4.1 é possível visualizar o ícone criado para o Viagem Segura, que se assemelha a uma placa de trânsito onde ao centro contém um veículo sendo atingido.

Figura 5.1: Ícone do aplicativo Viagem Segura

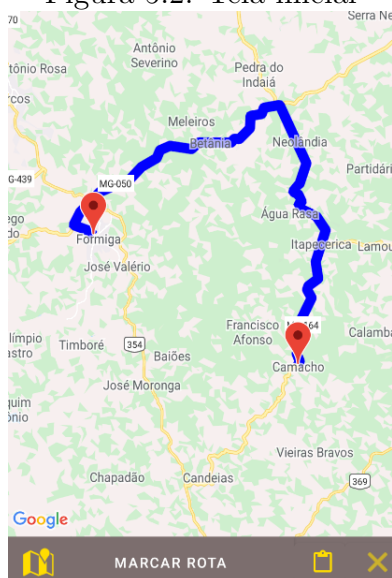


Fonte : Autoria própria

5.2 Tela Inicial

A tela inicial do aplicativo contém a mapa obtido através da *API Google Maps* junto com 3 botões, onde um tem a função de marcar a rota, outro tem a função de gerar o relatório da rota marcada, e o último tem como função limpar a rota marcada no mapa. A tela pode ser vista na Figura 5.2.

Figura 5.2: Tela inicial

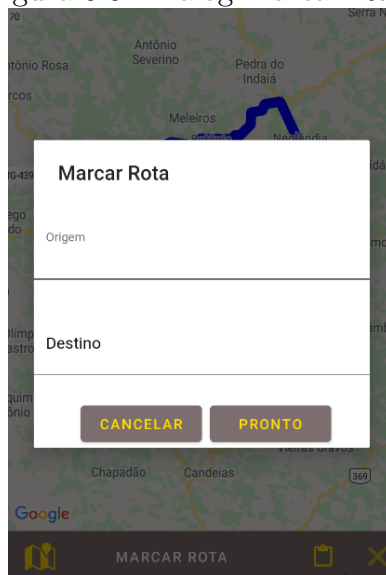


Fonte : Autoria própria

5.3 Dialog Marcar Rota

Ao clicar no botão de marcar rota, o aplicativo abre um *dialog* que contém dois campos, que devem ser preenchidos com a cidade de origem e a cidade destino. A tela também apresenta dois botões, um para confirmar os dados e marcar a rota, e outro para fechar a janela caso o usuário não queira marcar nenhuma rota. O *dialog* pode ser visualizado na Figura 5.3.

Figura 5.3: Dialog Marcar Rota



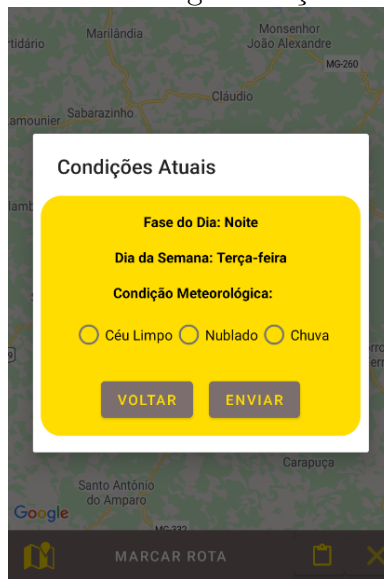
Fonte : Autoria própria

5.4 Dialogs Condições Atuais e Relatório da Rota

Quanto ao botão de gerar relatório, ao ser clicado abre um *dialog* onde o usuário deverá informar sobre a condição meteorológica durante a solicitação da rota, sendo que o próprio

aplicativo faz a verificação sobre a fase do dia e o dia da semana. Como visto na Figura 5.4, a tela ainda conta com um botão de confirmação, e outro para fechar a janela.

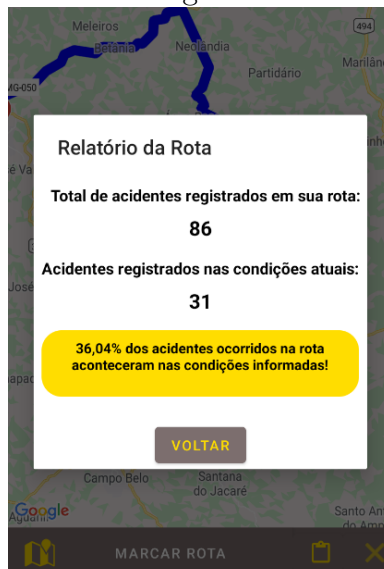
Figura 5.4: Dialog Condições Atuais



Fonte : Autoria própria

Ainda na tela de condições, ao clicar o botão Enviar o aplicativo abrirá outro *dialog*, onde será possível visualizar o relatório da rota, que conta com informações como o total de acidentes registrados na rota, o total de acidentes ocorridos nas condições especificadas. A tela pode ser vista na Figura 5.5.

Figura 5.5: Dialog Relatório da Rota



Fonte : Autoria própria

Conclusões

6.1 Considerações Finais

Como pode ser visto no trabalho, acidentes de trânsito tem ocasionado milhões de mortes anualmente, o que evidencia a necessidade de soluções para o problema. Já foram desenvolvidas diversas ferramentas que auxiliam nesse âmbito, o que contribuiu para a redução do mesmo.

Ao final da implementação do projeto, foi possível concluir que o aplicativo desenvolvido tem um grande potencial para contribuir na redução de acidentes de trânsito, pois através das informações recebidas, o usuário estará ciente da ocorrência de condições problemáticas em sua viagem, e conseqüentemente ficará mais atento.

A mineração de dados se mostrou muito relevante para o trabalho, pois facilitou bastante a obtenção de informações utilizadas na aplicação. Vale também ressaltar a utilização do *Python*, que devido a sua grande variedade de bibliotecas facilitou bastante a implementação dos *scripts*.

Por fim, devido ao frequente uso de dispositivos móveis pela sociedade, sendo grande parte dos aparelhos com sistema operacional *Android*, a plataforma se mostrou viável para o projeto, tornando o aplicativo de fácil acesso.

6.2 Trabalhos Futuros

Para maior simplicidade do sistema, o aplicativo desenvolvido no trabalho apresentou apenas funcionalidades básicas, e como trabalhos futuros podem ser implementadas novas funcionalidades, como por exemplo:

- A utilização de uma *API* que colete dados do clima, facilitando ainda mais a utilização do aplicativo por parte do usuário;
- A opção de agendar uma viagem, possibilitando que o usuário obtenha informações não só de viagens imediatas;

- A utilização de notificações *push*, possibilitando a emissão de alertas sobre trechos mais problemáticos durante a viagem;

Referências Bibliográficas

AMARAL, F. *Introdução à Ciência de Dados: mineração de dados e big data*. [S.l.]: Alta Books Editora, 2016.

CASTRO, L. N. d.; FERRARI, D. G. *Introdução à mineração de dados: conceitos básicos, algoritmos e aplicações*. São Paulo: Saraiva, 2016.

DEITEL, H.; DEITEL, P.; DEITEL, A. *Android: Como Programar*. [S.l.]: Bookman Editora, 2015.

DEVELOPERS. *Conheça o Android Studio*. 2021. Disponível em: <<https://developer.android.com/studio/intro>>.

DEVELOPERS. *Visão geral do SDK do Maps para Android*. 2021. Disponível em: <<https://developers.google.com/maps/documentation/android-sdk/overview?section=start>>.

DIGITAL, B. *BASES DE DADOS: conceito, classificações, critérios, aspectos importantes e exemplos*. 2017. Disponível em: <<https://biblioteconomiadigital.com.br/2017/12/bases-de-dados-conceito-classificacoes.html>>.

FIREBASE. *Firestore Realtime Database*. 2021. Disponível em: <<https://firebase.google.com/docs/database?hl=pt>>.

IBGE. *Uso de internet, televisão e celular no Brasil*. 2019. Disponível em: <<https://educacao.ibge.gov.br/jovens/materias-especiais/20787-uso-de-internet-televisao-e-celular-no-brasil.html#subtitulo-1>>.

IBGE. *Frota de veículos*. 2020. Disponível em: <<https://cidades.ibge.gov.br/brasil/pesquisa/22/28120>>.

IPEA. *Brasil gasta 132 bilhões por ano com acidentes de transporte*. 2020. Disponível em: <https://www.ipea.gov.br/portal/index.php?option=com_content&view=article&id=36658>.

JÚNIOR, A. J. de O.; SIMÕES, R. P. Desenvolvimento de um aplicativo android utilizando a classe flinganimation para abordagem de conceitos de cinemática. *Research, Society and Development*, v. 10, n. 1, p. e28410111710–e28410111710, 2021.

JÚNIOR, E. S. Diagrama de caso de uso. 2020.

JUNIOR, G. M. D. M. Implementação de um protótipo de aplicativo móvel com a integração com a ferramenta google maps e o componente gps dos smartphones voltado ao auxílio no transporte público da cidade de criciúma. 2018.

JUNIOR, G. T. B.; BERTHO, A. C. S.; VEIGA, A. d. C. A letalidade dos acidentes de trânsito nas rodovias federais brasileiras em 2016. *Revista Brasileira de Estudos de População*, SciELO Brasil, v. 36, 2019.

LIMA, W. D. Android e a influência do sistema operacional linux. *Tecnologias em Projeção*, v. 8, n. 1, p. 100–111, 2017.

MASSAÚ, G. C.; ROSA, R. G. da. Acidentes de trânsito e direito à saúde: prevenção de vidas e economia pública. *Revista de Direito Sanitário*, v. 17, n. 2, p. 30–47, 2016.

OMS. *Road traffic injuries*. 2020. Disponível em: <https://www.who.int/health-topics/road-safety#tab=tab_1>.

OMS. *Road traffic injuries*. 2021. Disponível em: <<https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>>.

PROGRAMATHOR. *Principais aplicações da linguagem Python*. 2018. Disponível em: <<https://programathor.com.br/blog/aplicacoes-linguagem-python/>>.

RODY, H. A.; JÚNIOR, S. J. de M.; SILVA, R. C. P. da. Radares e lombadas eletrônicas-a “moderação de tráfego” nas rodovias estaduais. *Projectus*, v. 1, n. 2, p. 81–87, 2017.

SILVA, R. O. da; SILVA, I. R. S. Linguagem de programação python. *TECNOLOGIAS EM PROJEÇÃO*, v. 10, n. 1, p. 55–71, 2019.

SILVEIRA, I. H. d.; OLIVEIRA, B. F. A. d.; JUNGER, W. L. Utilização do google maps para o georreferenciamento de dados do sistema de informações sobre mortalidade no município do rio de janeiro, 2010-2012. *Epidemiologia e Serviços de Saúde*, SciELO Brasil, v. 26, p. 881–886, 2017.

SINDIPECAS. *Frota Circulante*. 2020. Disponível em: <<https://www.sindipecas.org.br/area-atuacao/?co=s&a=frota-circulante>>.

TRANSITO, P. do. *Cinco aplicativos para ajudar no trânsito*. 2019. Disponível em: <<https://www.portaldotransito.com.br/noticias/5-aplicativos-para-ajudar-no-transito-2/>>.