Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas

Um aplicativo para gerenciamento de aspectos da qualidade do ambiente urbano.

Jean Kenned Teodoro Jeronimo

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:
Prof. Dr. George Henrique Godim da Fonseca

Julho, 2019
João Monlevade–MG

Jean Kenned Teodoro Jeronimo

Um aplicativo para gerenciamento de aspectos da qualidade do ambiente urbano.

Orientador: Prof. Dr. George Henrique Godim da Fonseca

Monografia apresentada ao curso de Engenharia da Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina "Trabalho de Conclusão de Curso II".

Universidade Federal de Ouro Preto

João Monlevade

Julho de 2019

A Ficha Catalográfica é elaborada exclusivamente pela Biblioteca. Substitua esta página pelo documento gerado na versão final da sua monografia.

FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

Um aplicativo para gerenciamento de aspectos da qualidade do ambiente urbano.

Jean Kenned Teodoro Jeronimo

Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI496 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Engenharia da Computação e aprovada pela Banca Examinadora abaixo assinada:

Prof. Dr. George Henrique Godim da Fonseca
Doutor em Engenharia Elétrica
DECSI – UFOP

Prof. Dr. Fernando Bernardes de Oliveira
Doutor em Engenharia Elétrica
Examinador
DECSI – UFOP

Prof. Me. Paganini Barcellos de Oliveira
Mestre em Engenharia de Produção
Examinador
DEENP – UFOP

João Monlevade, 3 de julho de 2019

Este trabalho é dedicado aos meus pais, que pelo apoio e incentivo constante, me permitiu chegar até esta etapa da vida.

Agradecimentos

A Deus, que propiciou que essa vitória acontecesse em minha vida, sempre me guiando, me protegendo e me dando forças, saúde e disposição para enfrentar essa longa jornada.

A toda minha família, especialmente a minha mãe Celina Augusta Teodoro Jeronimo e meu pai Antônio Alves Jeronimo, por todo apoio incondicional, por me ensinarem o valor do trabalho e da dedicação e por sempre estarem ao meu lado nos momentos mais difíceis, não medindo esforços para me ajudar a concluir essa etapa.

A universidade, aos funcionários e a todos os professores que se esforçaram dia após dia para que não apenas o conhecimento, mas também os valores e vivências fossem passados adiante, com amor e dedicação ao trabalho.

A todos os meus amigos, companheiros fiéis durante essa etapa, e que certamente ficarão marcados em minha vida, pelo companheirismo e amizade verdadeira que construímos.

"A força não vem das vitórias. Seus esforços desenvolvem a sua força. Quando você passa por dificuldades e decide não se render, isso é força."

— Arnold Schwarzenegger

Resumo

Este trabalho apresenta uma aplicação Android destinada a prover apoio às autoridades públicas e a população no processo de detecção e resolução de problemas urbanos presentes na cidade de João Monlevade-MG. O trabalho possui como objetivo permitir que habitantes denunciem os problemas urbanos encontrados e que essas denúncias cheguem de fato ao conhecimento dos órgãos responsáveis. A aplicação possui dois modos de operação, um para utilização de habitantes, e outro destinado a pessoas ligadas aos órgãos responsáveis. A partir disso busca-se obter uma melhor comunicação entre ambos por meio de *smartphones*, de maneira simples, eficiente e de baixo custo. Além disso, foi utilizado um banco de dados fornecendo um rápido fluxo de dados entre cliente/servidor. Pode-se concluir que foi obtido um resultado satisfatório mostrando que o sistema pode ser implementado imediatamente, auxiliando na localização de problemas presentes na cidade.

Palavras-chaves: Problemas urbanos. Localização. Android.

Abstract

This work presents an Android application intended to provide support to public authorities and the population in the process of detecting and solving urban problems in the city of João Monlevade-MG. The aim of the work is to allow inhabitants to denounce the urban problems encountered and that these denunciations actually reach the knowledge of the responsible bodies. The application has two modes of operation, one for the use of inhabitants, and the other for people connected to the responsible bodies. From this it is sought to obtain a better communication between both through smartphones, in a simple, efficient and low cost way. In addition, a database was used to provide a fast flow of data between client / server. It can be concluded that a satisfactory result was obtained showing that the system can be implemented immediately, helping to locate problems present in the city.

Key-words: Urban problems. Location. Android.

Lista de ilustrações

Figura 1 – Visão geral do sistema 20
Figura 2 – Diagrama de Casos de Uso 23
Figura 3 – Diagrama de Sequência de Sistema - Inserir problema 32
Figura 4 – Diagrama de Sequência de Sistema - Visualizar mapa de problemas . . 33
Figura 5 – Diagrama de Sequência de Sistema - Visualizar detalhes do problema . 33
Figura 6 – Diagrama de Sequência de Sistema - Editar ou excluir problemas . . . 34
Figura 7 – Diagrama de Sequência de Sistema - Marcar problema como resolvido . 34
Figura 8 – Diagrama de Sequência de Sistema - Notificar usuário 35
Figura 9 – Diagrama de Sequência de Sistema - Notificar administrador 35
Figura 10 – Console do *Firebase* 38

Figura 11 – Telas de login 40
Figura 12 – Visualização do mapa 41
Figura 13 – Detalhes do problema 41
Figura 14 – Telas de cadastro de ocorrência 42
Figura 15 – Ocorrências cadastradas pelo usuário 43
Figura 16 – Opções de edição e exclusão 44
Figura 17 – Edição de ocorrência 45
Figura 18 – Exclusão de ocorrência 45
Figura 19 – Notificando habitantes 45
Figura 20 – Marcar problema como resolvido 46
Figura 21 – Botões adicionais para administradores 47
Figura 22 – Todas ocorrências 47
Figura 23 – Tela de estatísticas 48
Figura 24 – Frequência das categorias afetadas 48
Figura 25 – Bairros afetados 48
Figura 26 – Resolução de ocorrências 48
Figura 27 – Notificando administradores 49
Figura 28 – Ausência de sinal de internet 50
Figura 29 – Mensagem de GPS desabilitado 50
Figura 30 – Problema inserido ultrapassa os limites territoriais 51

Lista de tabelas

Tabela 1 – Comparativo com aplicações relacionadas 18

Lista de quadros

Quadro 1 – Caso de uso - Inserir problema 24
Quadro 2 – Caso de uso - Visualizar mapa de problemas 25
Quadro 3 – Caso de uso - Visualizar detalhes de problemas 26
Quadro 4 – Caso de uso - Editar ou excluir problemas 27
Quadro 5 – Caso de uso - Marcar problema como resolvido. 28
Quadro 6 – Caso de uso - Notificar usuário. 29
Quadro 7 – Caso de uso - Notificar administrador. 30

Lista de abreviaturas e siglas

API *Application Programming Interface*

FGV Fundação Getúlio Vargas

GPS *Global Positioning System*

IDE *Integrated Development Environment*

IPTU Imposto Predial e Territorial Urbano

JSON *JavaScript Object Notation*

NoSQL *Not Only Structured Query Language*

Sumário

1 INTRODUÇÃO 14
1.1 Problema abordado 14
1.2 Objetivos 15
1.3 Justificativa 15
1.4 Organização do trabalho 16

2 REVISÃO BIBLIOGRÁFICA 17

3 METODOLOGIA 19
3.1 Visão geral do sistema 19
3.2 Tecnologias e ferramentas utilizadas 20
3.2.1 Firebase 21
3.3 Diagramas de casos de uso 22
3.4 Casos de uso 23
3.5 Análise de Requisitos 30
3.6 Diagrama de Sequência de Sistema 31
3.7 Administradores 36
3.8 Banco de Dados 37

4 ANÁLISE DOS RESULTADOS 40
4.1 Funcionalidades gerais 40
4.2 Funcionalidades relacionadas aos usuários 41
4.3 Funcionalidades relacionadas aos administradores 45
4.4 Ausência de sinal de GPS ou internet 49
4.5 Inserção de problemas que não pertencem a João Monlevade-MG . 51
4.6 Testes realizados 51

5 CONCLUSÃO 53

REFERÊNCIAS 55

1 Introdução

A popularização dos *smartphones* está cada vez maior, fazendo parte do cotidiano de pessoas de todo o mundo. As facilidades em relação à troca de informações, utilização de serviços, acesso a entretenimento, entre outras, são proporcionadas pelos *smartphones*. Portanto, atualmente essa tecnologia deixa de ser uma ferramenta opcional para se tornar uma necessidade. Uma pesquisa realizada pela Delloite (2018) mostra que o dispositivo foi o mais usado recentemente, superando outros dispositivos como *notebooks*, *desktops* e celulares convencionais, sendo utilizado por 92% dos entrevistados. A Fundação Getúlio Vargas (FGV) também realizou uma pesquisa neste mesmo ano, e nela consta que o Brasil possui mais *smartphones* ativos do que habitantes, chegando a 220 milhões de aparelhos (MEIRELLES, 2018).

Além disso, outra tecnologia em alta é a plataforma Android, que a empresa Statcounter (2017) anunciou como o sistema operacional mais utilizado no mundo, tomando o posto do *windows* em março de 2017. Sempre se atualizando, o sistema busca fornecer cada vez mais ferramentas para que desenvolvedores utilizem o máximo de recursos que os *smartphones* podem oferecer.

Tal cenário de expansão de ambas tecnologias tem atraído desenvolvedores e empresas para esse ramo, e novas soluções são buscadas para as mais diversas áreas. Existem aplicações em áreas corporativas, financeiras, militares, transportes, educação, entre muitas outras. Isso tudo tem facilitado a vida das pessoas e cada vez mais vem surgindo soluções para problemas nos mais diversos âmbitos.

Ao observar as cidades brasileiras é possível encontrar vários problemas urbanos que necessitam de solução. A área da administração pública, a qual atribui responsabilidades e deveres aos governantes, prefeituras e empresas contratadas, tais como resolver os problemas urbanos é outro exemplo a qual a tecnologia pode favorecer. Os problemas poderiam então ser mais facilmente encontrados e resolvidos com a utilização das tecnologias supracitadas, e dessa forma melhorar o cenário caótico em que as cidades se encontram.

1.1 Problema abordado

Atualmente, as cidades brasileiras têm enfrentado muitos problemas urbanos como, por exemplo, vias esburacadas, falta de iluminação em algumas localidades, esgoto a céu aberto, lixo em local indevido, superlotação de transportes públicos, queimadas indevidas, crimes, entre outros. Pode-se ir mais afundo para entender a causa destes problemas. Segundo Ribeiro e Vargas (2015, p. 17) isso ocorre devido o “descompasso entre

crescimento populacional e acesso a serviços e oportunidades, por limitação de recursos naturais e econômicos", o que acaba levando à "degradação do ambiente natural e do construído".

A população reclama com total direito, porém muitas dessas reclamações não chegam de fato aos responsáveis por resolver tais problemas. Há uma clara falta de comunicação entre população e autoridades públicas. O cenário ainda piora quando o número de ocorrência é grande, comum em cidades maiores, devido ao processo de urbanização acelerado. Portanto, visando atenuar essa situação, a criação de um sistema computacional pode ser útil para dar suporte tanto para que os habitantes tenham uma maior facilidade para informar os problemas de sua região, quanto para os responsáveis, durante o processo de localização desses problemas. A aplicação proposta neste trabalho é voltada para a cidade de João Monlevade-MG.

1.2 Objetivos

O presente trabalho consiste em desenvolver e implantar uma aplicação móvel que utiliza a plataforma Android em *smartphones*. Tal sistema possibilitaria aos habitantes de uma cidade informarem onde se localizam os problemas urbanos encontrados e descrevam tais problemas. Além disso, permitiria que todas as informações a respeito desses problemas sejam encaminhadas para os responsáveis, para que seja dado início a sua resolução.

Este trabalho possui os seguintes objetivos específicos:

- Permitir que habitantes informem os problemas encontrados em sua cidade.
- Permitir que os habitantes saibam o quão sua cidade está sendo afetada pelos problemas urbanos.
- Possibilitar que órgãos responsáveis tenham acesso as informações a respeito de todos os problemas e possam dar início as resoluções.
- Estreitar a relação entre órgãos responsáveis e habitantes, aumentando a comunicação entre ambos.

1.3 Justificativa

É possível notar os mais diversos problemas que a cidade João Monlevade-MG enfrenta, tais como vias em péssimas condições, locais mal iluminados, ausência de saneamento básico. Os habitantes da cidade convivem diariamente com essas e outras dificuldades e é dever dos governantes prover melhorias para esse cenário. Logo, o desenvolvimento

deste trabalho tem como justificativa apoiar e dar suporte à população e aos governantes para que tenham um meio mais fácil de localizar os problemas urbanos da cidade.

1.4 Organização do trabalho

Este trabalho é organizado como se segue. O Capítulo 2 apresenta algumas aplicações presentes atualmente no mercado que possuem propostas semelhantes a deste sistema, indicando as principais diferenças entre tais aplicações e o sistema proposto. O Capítulo 3 apresenta a medotologia de pesquisa e desenvolvimento do sistema, apresentando a sua visão geral, as tecnologias e ferramentas utilizadas, bem como a parte de projeto e análise do *software*, e a estruturação do banco de dados. O Capítulo 4 apresenta as telas criadas da aplicação, a descrição das funcionalidades, o fluxo entre as telas, além dos testes realizados. E por fim o Capítulo 5 faz as considerações finais e apresenta futuros trabalhos que podem servir como melhoria da aplicação.

2 Revisão bibliográfica

Existem cada vez mais aplicações sendo utilizadas a favor de causas sociais. Muitas *startups* têm investido nesse ramo e eventos como a *Campus Party* (PEGN, 2016) têm incentivado os desenvolvedores a acolher as mais variadas causas. O crescimento do interesse por esse tipo de ação vem aumentando a qualidade das soluções desenvolvidas e cobrindo um campo maior de soluções, como nas áreas de transporte, sustentabilidade, meio ambiente, saúde, trabalho voluntário, entre muitas outras. Ao se aprofundar mais no escopo deste trabalho, que está relacionado à denúncia de problemas urbanos em geral, é possível encontrar algumas aplicações que possuem funcionalidades e objetivos semelhantes. A seguir são listadas algumas delas.

O "Pelas Ruas", desenvolvido pela (RBS, 2017), é um aplicativo que possibilita a interação entre usuários de uma mesma cidade, possuindo uma *timeline* onde podem ser compartilhadas fotos dos problemas encontrados, além de permitir comentar as publicações feitas ou seguir outros usuários e/ou publicações. O aplicativo é destinado a usuários da cidade de Porto Alegre e região metropolitana.

O "Vigilante", desenvolvido por (MAQHIN, 2014), funciona de forma mais simplificada, possuindo uma interface mais intuitiva e de fácil utilização mesmo no primeiro contato. O usuário pode rapidamente inserir uma ocorrência marcando no mapa sua localização, além de também ser permitido a inserção de fotos e dados a respeito do problema, e dessa forma outros usuários podem ter acesso e visualizar tais marcações. O aplicativo porém, permite que problemas sejam cadastrados em qualquer região do Brasil, e por abranger uma área muito extensa, se torna uma tarefa complicada gerenciar todos os problemas que possam a vir a ser inseridos e encaminhá-los para os responsáveis.

O "Cidadão Online 4R", desenvolvido por (4RSISTEMAS, 2017), é mais uma aplicação com funcionalidades semelhantes ao "Vigilante". Apresenta-se como uma ferramenta de contato com os órgãos públicos, permitindo solicitações de serviços de forma ágil. Além disso, o usuário pode detalhar tal ocorrência e marcar sua localização. Entretanto, para que funcionários das prefeituras das cidades sejam administradores e recebam notificações das solicitações, a cidade deve ser cliente do aplicativo.

O "PBH APP", desenvolvido pela (PRODABEL, 2018), abrange um escopo maior de serviços. Possui uma interface completa e de fácil utilização. Além de permitir solicitações para a resolução de problemas urbanos, possibilita também que o usuário emita a segunda via da guia do IPTU, ou mesmo localize equipamentos públicos presentes na cidade, como academias a céu aberto, por exemplo. Este aplicativo, porém, é limitado à cidade de Belo Horizonte-MG.

A Tabela 1 apresenta as diferenças e as funcionalidades que cada uma dessas aplicações possuem em relação ao sistema proposto. É importante destacar que a funcionalidade de "operar na cidade de João Monlevade-MG", na qual "Vigilante" e "Cidadão Online 4R" possuem, diz respeito ao fato de que ocorrências podem ser cadastradas nessa região, mas não necessariamente as informações sobre esses problemas chegarão aos órgãos responsáveis daquela área. É possível perceber então que o sistema proposto possui algumas particularidades que os outros estudados não possuem. Da mesma maneira, algumas funcionalidades citadas anteriormente como, por exemplo, permitir a interação entre usuários através de comentários, gerar segunda via de IPTU ou acesso a diplomas escolares, não estão presentes no sistema proposto. Essa estratégia foi tomada visando fornecer todas as principais funcionalidades necessárias focando na questão de problemas urbanos especialmente para a cidade de João Monlevade-MG.

Tabela 1 – Comparativo com as aplicações relacionadas.

	Pelas Ruas	**Vigilante**	**Cidadão online**	**PBH APP**	**Sistema proposto**
Cadastro e descrição de problemas	X	X	X	X	X
Órgãos responsáveis com acesso e controle das solicitações			X	X	X
Visualização do mapa da cidade com a localização dos problemas	X	X			X
Órgãos responsáveis e usuários são notificados em caso de ocorrência cadastrada ou resolvida			X	X	X
Opera na cidade de João Monlevade-MG		X	X		X

Fonte: Elaborado pelo autor

3 Metodologia

Este capítulo descreve a estrutura do sistema proposto, bem como os métodos, tecnologias e ferramentas utilizadas no processo de desenvolvimento e implementação. Além disso, são descritos os passos e estratégias tomadas, desde a fase de análise de requisitos até a fase de implementação e testes. A seguinte metodologia de trabalho foi utilizada:

- Revisão da literatura buscando aplicações semelhantes já implementadas, além de ferramentas que poderiam auxiliar no desenvolvimento.
- Desenvolvimento de uma aplicação móvel de interface simples e de fácil utilização que permita aos usuários informarem problemas presentes na cidade de João Monlevade-MG e que possam ser notificados quando suas solicitações forem atendidas. Além disso, possam visualizar o mapa da cidade com todos os problemas cadastrados por outros usuários.
- Desenvolvimento da aplicação voltada também a administradores, que são pessoas que serão responsáveis por dar início a resolução das solicitações e marcá-las como resolvidas. Administradores também devem ser notificados quando novas ocorrências forem adicionadas.
- Estruturação e desenvolvimento do banco de dados para armazenar dados de todas as ocorrências solicitadas, tanto as pendentes, quanto as já resolvidas.
- Processo de validação e realização de testes para encontrar possíveis erros e melhorias que possam ser realizadas.

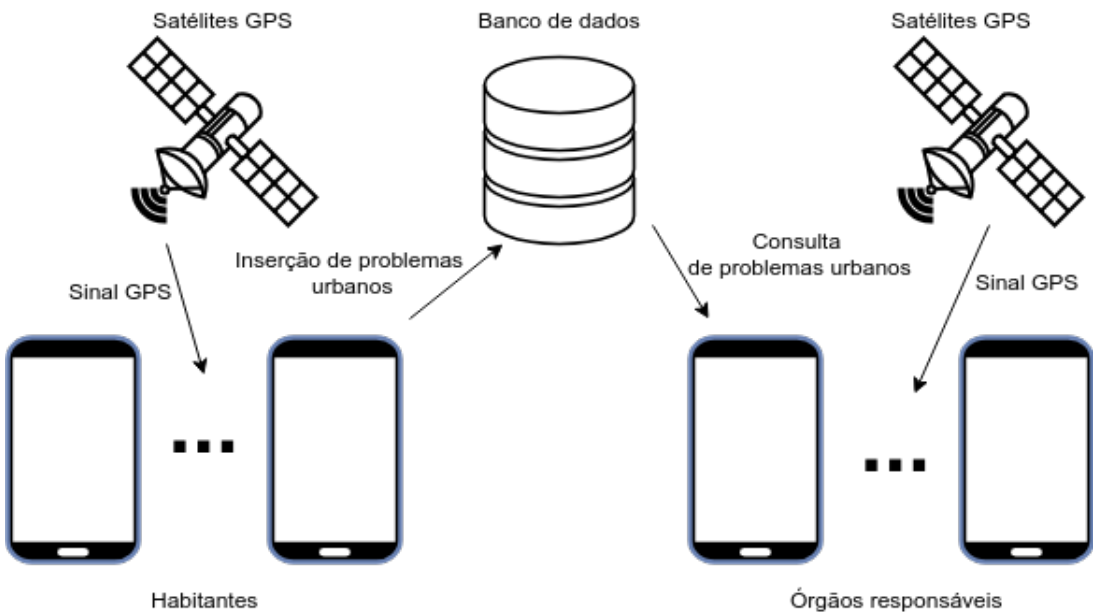
3.1 Visão geral do sistema

O aplicativo possui duas visões diferentes. A primeira delas é dedicada aos habitantes, ou seja, os usuários comuns. Essa visão permite o cadastro de problemas, cujos dados são enviados para um banco de dados.

A segunda visão é destinada aos órgãos responsáveis, os administradores. Essa visão possibilita também a inserção de problemas, mas além disso, permite o monitoramento de todas as ocorrências inseridas pelos habitantes, onde é possível atualizar o status das ocorrências a qualquer momento. Outro diferencial é a apresentação de dados estatísticos a respeito das ocorrências.

A Figura 1 mostra uma arquitetura simplificada do sistema implementado.

Figura 1 – Visão geral do sistema



Fonte: Elaborado pelo autor

3.2 Tecnologias e ferramentas utilizadas

Para a implementação da arquitetura apresentada foram utilizadas algumas tecnologias e ferramentas que auxiliaram na construção das funcionalidades propostas. A IDE utilizada foi o *Android Studio* na versão 3.2 sendo executado em um ambiente *Linux*. O *Android Studio* é uma ferramenta completa para o desenvolvimento de aplicativos Android, possuindo muitos recursos como um avançado emulador de *smartphones*, recursos de testes, uma interface intuitiva, além de ferramentas de depuração e localização de erros relacionados a compatibilidade entre versões, desempenho e outros.

Os códigos fontes escritos foram feitos utilizando a linguagem Java. Porém, aplicativos Android podem ser desenvolvidos utilizando outras linguagens como *Kotlin* e *C++*. Segundo (DEITEL; DEITEL, 2016), o *java* é umas das mais utilizadas quando se trata da construção de aplicações para dispositivos que usam comunicação em rede, que é o caso da aplicação desenvolvida. Além disso, há uma grande quantidade de dispositivos que executam *java* e um enorme número de desenvolvedores que trabalham com esta linguagem.

Outro recurso utilizado foi a plataforma do *Google Maps*. É através dela que é feita a exibição do mapa da cidade de João Monlevade-MG com a presença de ícones que demarcam os problemas urbanos adicionados, bem como possibilita que usuários façam buscas por endereços específicos.

Também foi utilizado a plataforma *Firebase* para a criação de algumas funcionalidades. O *Firebase* fornece vários serviços para o auxílio no desenvolvimento de aplicativos móveis, possibilitando que sejam criados de forma mais ágil e eficiente. A criação de servidores próprios para o atendimento de requisições de clientes é um exemplo de tarefa que toma tempo e esforço de programadores. No caso dessa aplicação, o *Firebase Realtime Database* é um dos recursos da plataforma que permitiu que este trabalho fosse poupado, pois possui uma API que sincroniza os dados em tempo real das aplicações móveis no dispositivos e os armazena na nuvem. Logo, todas as informações a respeito das ocorrências cadastradas pelos usuários estão armazenadas nesse banco de dados. A Seção 3.2.1 descreve todas os recursos utilizados da plataforma.

3.2.1 Firebase

Presente em mais de 1,5 milhões de aplicativos ativos (MA, 2018), o *Firebase* possui 18 ferramentas com o intuito de melhorar a qualidade dos produtos desenvolvidos, conforme é apresentado em (FIREBASE, 2019). Nesta aplicação foram utilizadas quatro delas.

- *Realtime Database*: Como já descrito anteriormente, é o banco de dados da aplicação. Permite o armazenamento e sincronização dos dados de maneira eficiente, sendo um banco de dados NoSQL hospedado na nuvem. Os dispositivos podem acessar os dados contidos no banco de qualquer lugar através da internet e realizar modificações em seu conteúdo, onde tais modificações podem ser refletidas em questão de milissegundos em todos os outros dispositivos. Além disso, dispensa a implementação de servidores próprios, já que o *Firebase* trata de forma interna as requisições e processos de sincronização. Outra vantagem é a sua utilização *offline*, visto que aplicativos que utilizam o *Realtime Database* mantém um *cache* local no dispositivo com todos os dados armazenados. Quando o usuário tiver acesso a internet os dados são sincronizados automaticamente com a nuvem.

- *Authentication*: Um sistema confiável de autenticação poderia levar muito tempo para ser implementado. Porém, esta ferramenta facilita este processo e permite que seja feito de forma rápida e segura, além de fornecer uma melhor experiência de login aos usuários. Possui compatibilidade com contas do *Google*, *Twitter*, *GitHub*, *Facebook*, entre outros. Neste trabalho a autenticação é feita utilizando a conta do *Google*.

- *Crashlytics*: Nenhuma aplicação está completamente livre de falhas e muitas delas podem ocorrer durante a utilização do sistema pelo usuário final. É fundamental que o desenvolvedor tenha ciência desses erros para que possa corrigi-los e prover assim uma melhor experiência ao usuário. O processo de rastreio de erros e posteriormente

sua solução, deve ser realizado o mais rápido possível e muitas vezes isso não ocorre, seja devido ao número grande de usuários ou à complexidade do sistema. Com o intuito de facilitar essa tarefa, o *Crashlytics* trabalha na localização de possíveis erros na aplicação, fornecendo um relatório de falhas. Este relatório contém o motivo da falha e o que a gerou, onde cada falha possui um nível de prioridade, dependendo do seu impacto no sistema. É possível ver inclusive a linha de código que está gerando tal problema, diminuindo o esforço para a correção dos erros.

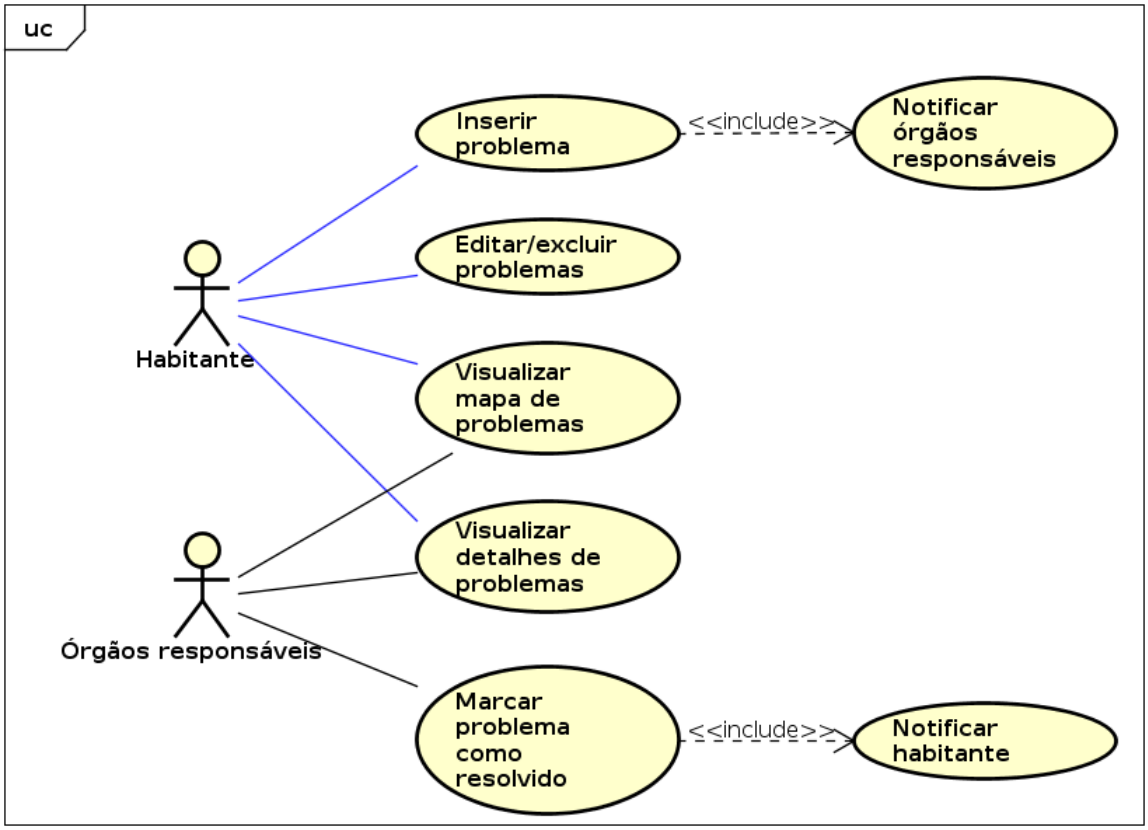
- *Google Analytics*: Com esta ferramenta é possível obter um conjunto completo de informações a respeito da aplicação, entender o comportamento dos usuários, analisar dados sobre falhas, desempenho e eficácia de notificações, por exemplo.

3.3 Diagramas de casos de uso

Utilizando a ferramenta Astah foi possível construir o diagrama de casos de uso do sistema, que é apresentado na Figura 2. É possível observar que há dois atores, os habitantes e os órgãos responsáveis. Além disso, é possível perceber que habitantes e órgãos responsáveis compartilham dois casos de uso, o de visualizar mapa de problemas e o de visualizar detalhes de problemas. As funcionalidades de inserir problema e editar ou excluir problemas são dedicadas aos habitantes, enquanto a de marcar problemas como resolvidos é direcionada aos administradores. Como é possível perceber, a relação de *include* é utilizada para as funcionalidades de notificação. Ou seja, sempre que o caso de uso "Inserir problema" ocorrer, o caso de uso "Notificar órgãos responsáveis" será executado. Da mesma forma, quando o caso de uso "Marcar problema como resolvido" for executado, em seguida o caso "Notificar habitante" também será.

Na Seção 3.4 são detalhados cada um desses casos de uso, bem como seus respectivos cenários.

Figura 2 – Diagrama de Casos de Uso



Fonte: Elaborado pelo autor

3.4 Casos de uso

A presente seção apresenta as estórias textuais dos casos de uso e seus cenários, que foram usados com o objetivo de descobrir e registrar os requisitos funcionais. É importante destacar que nos casos de uso apresentados habitante e órgãos responsáveis são tratados como usuários e administradores, respectivamente. O Quadro 1 apresenta o caso de uso que diz respeito à funcionalidade de adicionar uma nova ocorrência de problema encontrado por um usuário.

Já o Quadro 2 se trata da funcionalidade de visualização do mapa com os problemas sendo indicados no mapa, onde tanto administrador como usuários atuam como atores primários.

Quadro 1 – Caso de uso - Inserir problema

Caso de uso: Inserir problema
Objetivo: Permitir que o habitante cadastre um problema encontrado em sua cidade. Ator primário: Usuário. Interessados: • Usuário: Quer cadastrar o problema encontrado com mínimo esforço. • Administrador: Quer que problemas presentes na cidade sejam cadastrados sempre que encontrados por alguém. Pré-condições: • Usuário ter o aplicativo instalado no *Smartphone*. • Usuário já deve ter feito login no aplicativo.
Cenário de sucesso principal: 1. O usuário abre o aplicativo. 2. O usuário pressiona o botão para inserir nova ocorrência de problema. 3. O usuário insere detalhes a respeito do problema. 4. O usuário confirma no mapa a localização do problema. 5. Uma mensagem de cadastro realizado com sucesso é exibida ao usuário. Extensões: 1a. GPS do *smartphone* desabilitado. 1. Caixa de diálogo é exibida informando que é necessário esse recurso. 2. Em caso afirmativo, a tela de configurações de GPS é aberta e o usuário pode ativá-lo. 4a. Dados móveis do *smartphone* desabilitado. 1. Uma mensagem é exibida informando ao usuário tal situação. 2. O usuário é impedido de confirmar a localização e por consequência cadastrar novo problema até que os dados móveis sejam ativados.

Fonte: Elaborado pelo autor.

Quadro 2 – Caso de uso - Visualizar mapa de problemas

Caso de uso: Visualizar mapa de problemas
Objetivo: Permitir que usuários e administradores tenham uma visão do mapa da cidade de João Monlevade-MG com todos os problemas cadastrados em suas devidas localizações, através de pequenos ícones. Atores primários: Usuário e administrador. Interessados: • Usuário e administrador: Querem ter uma visão completa dos problemas presentes na cidade de João Monlevade-MG. Pré-condições: • Usuário ou administrador ter o aplicativo instalado no *Smartphone*. • Usuário ou administrador já deve ter feito login no aplicativo.
Cenário de sucesso principal: 1. O usuário ou administrador abre o aplicativo. 2. Todos os problemas são consultados no banco de dados e exibidos no mapa em suas respectivas localidades. Extensões: 1a. GPS do *smartphone* desabilitado. 1. Caixa de diálogo é exibida informando que é necessário esse recurso. 2. Em caso afirmativo, a tela de configurações de GPS é aberta e o usuário ou administrador pode ativá-lo.

Fonte: Elaborado pelo autor.

O Quadro 3 por sua vez, apresenta a funcionalidade de exibição de informações sobre um problema. Administrador e usuário podem consultar um problema e obter detalhes relevantes sobre o mesmo.

Quadro 3 – Caso de uso - Visualizar detalhes de problemas

Caso de uso: Visualizar detalhes de problemas
Objetivo: Permitir que usuários e administradores possam ver detalhes de um determinado problema, além de sua localização. Atores primários: Usuário e administrador. Interessados: • Usuário e administrador: Querem ver todas as informações a respeito de um determinado problema. Pré-condições: • Usuário ou administrador ter o aplicativo instalado no *Smartphone*. • Usuário ou administrador já deve ter feito login no aplicativo.
Cenário de sucesso principal: 1. O usuário ou administrador abre o aplicativo. 2. Todos os problemas são consultados no banco de dados e exibidos no mapa em suas respectivas localidades, através de pequenos ícones. 3. Usuário ou administrador clica no ícone do problema que deseja obter mais informações. 4. Uma nova tela é aberta com todas as informações a respeito do problema, tais como horário, data, categoria, descrição, foto, entre outras. Extensões: 1a. GPS do *smartphone* desabilitado. 1. Caixa de diálogo é exibida informando que é necessário esse recurso. 2. Em caso afirmativo, a tela de configurações de GPS é aberta e o usuário ou administrador pode ativá-lo.

Fonte: Elaborado pelo autor.

O Quadro 4 se refere a funcionalidade de edição e exclusão de problemas. O ator primário neste caso é o usuário.

Quadro 4 – Caso de uso - Editar ou excluir problemas

Caso de uso: Editar ou excluir problemas
Objetivo: Permitir que um determinado usuário possa editar informações de problemas que foram cadastrados por esse próprio usuário, ou até mesmo excluí-lo se for de seu desejo. Ator primário: Usuário. Interessados: • Usuário: Quer editar ou excluir um problema cadastrado. Pré-condições: • Usuário ter o aplicativo instalado no *Smartphone*. • Usuário já deve ter feito login no aplicativo. • Usuário já ter cadastrado algum problema no aplicativo.
Cenário de sucesso principal: 1. O usuário abre o aplicativo. 2. O usuário clica no botão que abre a lista de problemas adicionados pelo mesmo. 3. O usuário clica em um desses problemas. 4. São exibidos detalhes do problema, além de dois botões, um que permite editar e outro excluir. 5. O usuário escolhe uma das opções e confirma a operação. 6. Uma mensagem de edição ou exclusão realizada com sucesso é exibida ao usuário. Extensões: 1a. GPS do *smartphone* desabilitado. 1. Caixa de diálogo é exibida informando que é necessário esse recurso. 2. Em caso afirmativo, a tela de configurações de GPS é aberta e o usuário ou administrador pode ativá-lo. 2a. Forma alternativa. 1. Todos os problemas são consultados no banco de dados e exibidos no mapa em suas respectivas localidades, através de pequenos ícones. 2. Usuário clica no ícone de um problema que foi inserido pelo próprio usuário. 3. São exibidos detalhes do problema, além de dois botões, um que permite editar e outro excluir. 4. O usuário escolhe uma das opções e confirma a operação. 5. Uma mensagem de edição ou exclusão realizada com sucesso é exibida ao usuário.

Fonte: Elaborado pelo autor.

O Quadro 5 mostra a funcionalidade de marcar um problema cadastrado como resolvido. Esta funcionalidade é dedicada a administradores, onde é dever do mesmo solicitar a execução do serviço para solução do problema, e após a completa realização do serviço marcá-lo como resolvido.

Quadro 5 – Caso de uso - Marcar problema como resolvido.

Caso de uso: Marcar problema como resolvido.
Objetivo: Permitir que o administrador possa marcar um problema como resolvido, ou seja, atualizar o status do problema de pendente para resolvido. Ator primário: Administrador. Interessados: • Usuário: Quer que seu problema seja resolvido. • Administrador: Quer prover solução ao problema. Pré-condições: • Administrador ter o aplicativo instalado no *Smartphone*. • Administrador já deve ter feito login no aplicativo. • Algum usuário ter cadastrado ao menos um problema no aplicativo.
Cenário de sucesso principal: 1. O administrador abre o aplicativo. 2. O administrador clica no ícone que abre a lista dos problemas cadastrados até o momento. 3. O administrador clica em um desses itens da lista. 4. São exibidos detalhes do problema, além de um botão, que fornece a opção de marcar aquele problema como resolvido. 5. Caso afirmativo o status do problema é alterado de pendente para resolvido. 6. Uma mensagem de operação realizada com sucesso é exibida ao administrador. Extensões: 1a. GPS do *smartphone* desabilitado. 1. Caixa de diálogo é exibida informando que é necessário esse recurso. 2. Em caso afirmativo, a tela de configurações de GPS é aberta e o usuário ou administrador pode ativá-lo. 2a. Forma alternativa. 1. Todos os problemas são consultados no banco de dados e exibidos no mapa em suas respectivas localidades, através de pequenos ícones. 2. Administrador clica no ícone de um problema. 3. São exibidos detalhes do problema, além de um botão, que fornece a opção de marcar aquele problema como resolvido. 4. Caso afirmativo o status do problema é alterado de pendente para resolvido. 5. Uma mensagem de operação realizada com sucesso é exibida ao administrador.

Fonte: Elaborado pelo autor.

O Quadro 6 trata-se de um serviço cujo ator primário é o sistema, sendo responsável por notificar usuários quando problemas são resolvidos. Tal caso de uso ocorre apenas após a execução do caso de uso retratado no Quadro 5.

Quadro 6 – Caso de uso - Notificar usuário.

Caso de uso: Notificar usuário.
Objetivo: Notificar um usuário quando um problema inserido pelo mesmo for resolvido. Ator primário: Sistema. Interessados: • Usuário: Quer ser avisado quando algum de seus problemas for solucionado. Pré-condições: • Usuário ter o aplicativo instalado no *Smartphone*. • Usuário já deve ter feito login no aplicativo. • Usuário já ter cadastrado algum problema no aplicativo.
Cenário de sucesso principal: 1. Caso de uso do Quadro 5 é executado, onde um problema é marcado como resolvido. 2. Uma notificação é enviada ao usuário que cadastrou esse problema. 3. O usuário clica na notificação recebida e é direcionado para a tela com a lista de problemas cadastrados pelo próprio usuário, podendo verificar qual dos problemas foi resolvido. Extensões: 3a. Usuário fecha a notificação. 1. O usuário já foi notificado, porém a tela de problemas cadastrados não será aberta, o que não impede o usuário de verificar posteriormente quando desejar.

Fonte: Elaborado pelo autor.

E finalmente o Quadro 7, que também possui como ator primário o sistema, que agora é responsável por notificar o administrador quando algum problema for cadastrado no sistema. É importante lembrar que tal caso de uso é uma inclusão do caso de uso do Quadro 1, ocorrendo apenas após a execução deste caso de uso.

Quadro 7 – Caso de uso - Notificar administrador.

Caso de uso: Notificar administrador.
Objetivo: Notificar o administrador quando um problema for inserido por algum usuário. Ator primário: Sistema. Interessados: • Administrador: Quer ser avisado quando algum problema for inserido, para que o mesmo possa dar inicio o mais rápido possível a solução deste problema. Pré-condições: • Administrador ter o aplicativo instalado no *Smartphone*. • Administrador já deve ter feito login no aplicativo. • Algum usuário já ter cadastrado ao menos um problema no aplicativo.
Cenário de sucesso principal: 1. Caso de uso do Quadro 1 é executado, dessa forma algum problema encontrado na cidade é então inserido. 2. Uma notificação é enviada ao administrador informando sobre a situação. 3. O administrador clica na notificação recebida e é direcionado para a tela com a lista de problemas cadastrados, podendo verificar qual problema foi cadastrado e consequentemente está com seu status pendente. Extensões: 3a. Administrador fecha a notificação. 1. O administrador já foi notificado, porém a tela de problemas cadastrados não será aberta, o que não impede o administrador de verificar posteriormente quando desejar.

Fonte: Elaborado pelo autor.

3.5 Análise de Requisitos

Através do estudo dos casos de uso foi identificado e definido os requisitos funcionais referentes aos usuários comuns (habitantes) e administradores (órgãos responsáveis). Em relação aos usuários comuns foram definidas as seguintes funcionalidades:

1. Permitir que o usuário insira um problema urbano que será enviado ao banco de dados na nuvem. Deve ser possível selecionar a categoria do problema, marcar sua localização no mapa, e adicionar outras informações relevantes a este problema.

2. Exibir o mapa completo da cidade de João Monlevade-MG sinalizando os problemas urbanos em suas devidas localizações.

3. Possibilitar que todos os usuários possam visualizar os detalhes sobre os problemas urbanos presentes no mapa.
4. Exibir dados das ocorrências registradas pelo próprio usuário, bem como seu status (se já foi resolvida ou ainda está pendente).
5. Permitir que usuários possam editar ou excluir suas próprias ocorrência cadastradas.
6. Exibir notificação para o usuário quando um problema cadastrado pelo mesmo for resolvido pelos órgãos responsáveis.

Para os administradores foram definidas as seguintes funcionalidades:

1. Permitir que administradores marquem as ocorrências como resolvidas.
2. Exibir notificação para administradores quando problemas forem adicionados e pertencerem a sua área de atuação.

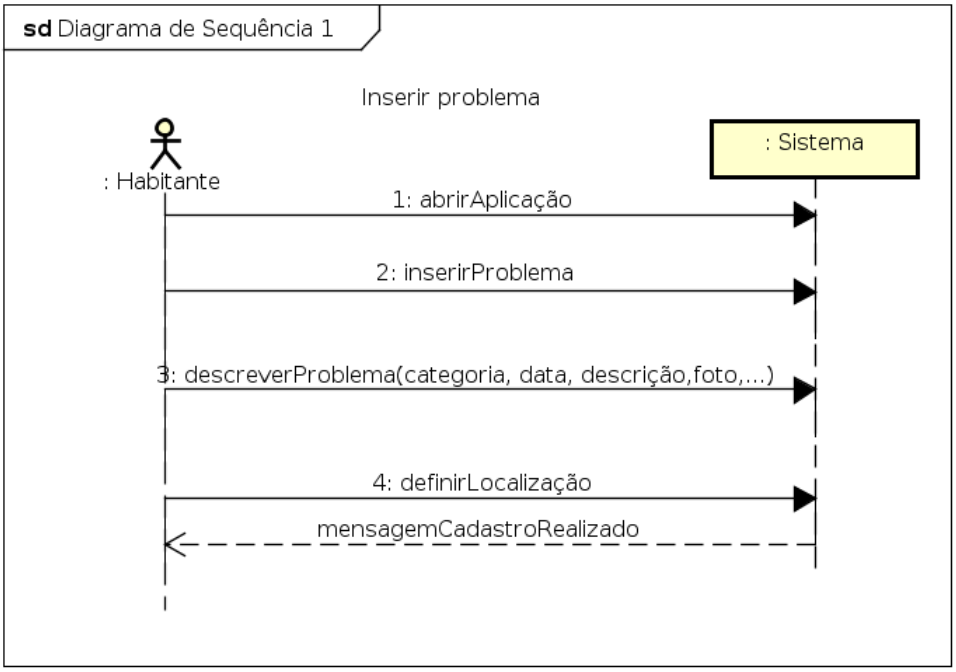
Em relação aos requisitos não funcionais, tanto por parte dos administradores quanto dos usuários comuns, foram destacados os seguintes:

1. Ser simples e intuitivo, no qual habitantes possam adicionar ocorrências de forma rápida e administradores possam facilmente alterar o status desses problemas.
2. Funcionar em todos os dispositivos com sistema operacional *Android* com versão 4.0.3 ou superior a esta.
3. Ter boa disponibilidade na cidade de João Monlevade-MG, utilizando sinal de GPS e internet.

3.6 Diagrama de Sequência de Sistema

A fim de entender a interação dos atores com o sistema desenvolvido, é apresentado nessa seção os diagramas de sequência de sistema para cada caso de uso. Esta é uma etapa importante da análise do comportamento do sistema, sendo necessário investigar e definir de forma eficaz seu comportamento antes de proceder com o projeto. A Figura 3 apresenta o diagrama de sequência para a funcionalidade de inserir problema, onde o habitante atua interagindo com o sistema.

Figura 3 – Diagrama de Sequência de Sistema - Inserir problema



Fonte: Elaborado pelo autor

A Figura 4 trata da questão da visualização dos problemas presentes no mapa, onde há apenas uma interação por parte do ator com o sistema e o resultado esperado já é obtido.

A Figura 5 apresenta o diagrama referente a funcionalidade de visualizar detalhes do problema, no qual tanto o habitante quanto os órgãos responsáveis interagem com o sistema.

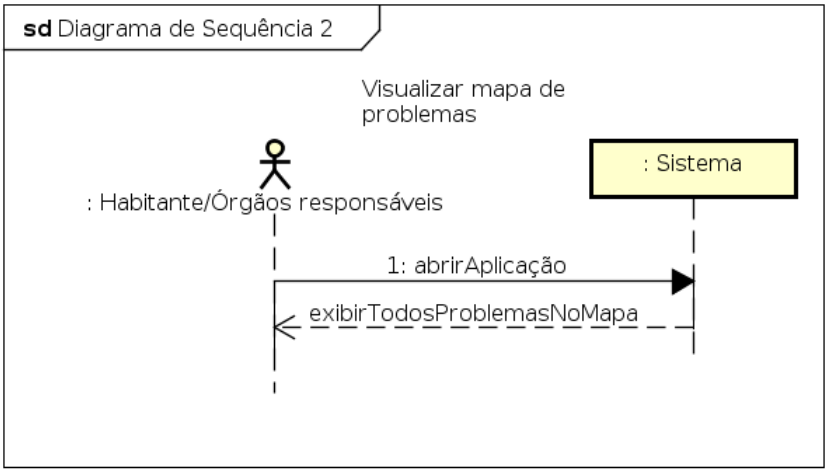
A Figura 6 por sua vez tem como ator apenas o habitante, que faz várias interações com o sistema antes de ter seu objetivo concluído nessa tarefa.

A Figura 7 mostra o diagrama da funcionalidade de marcar problema como resolvido, no qual os órgãos públicos são os atores externos.

A Figura 8 possui dois atores externos, e se comunicam com o sistema em momentos distintos para dessa forma executar a funcionalidade de forma correta, notificando o usuário.

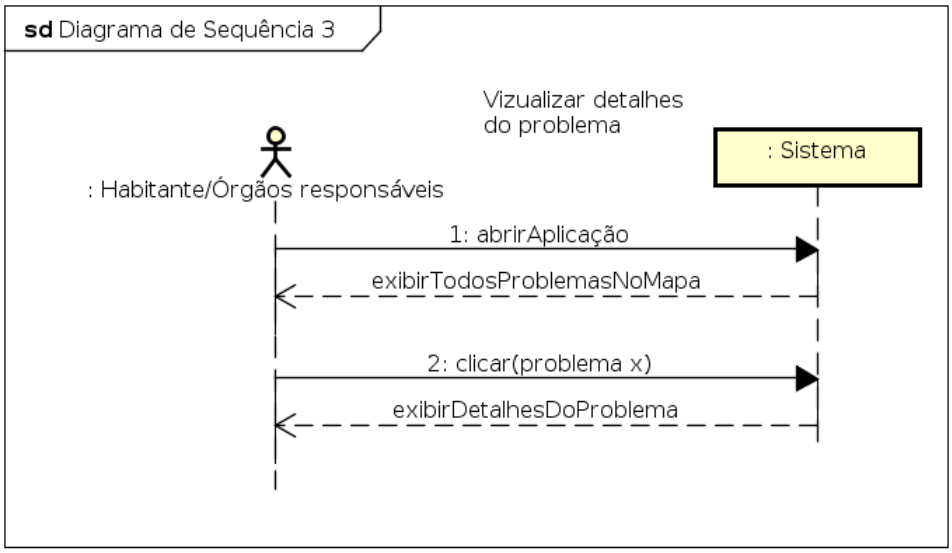
A Figura 9 possui também dois atores externos e com a função de notificar, porém desta vez o notificação é direcionada ao administrador.

Figura 4 – Diagrama de Sequência de Sistema - Visualizar mapa de problemas



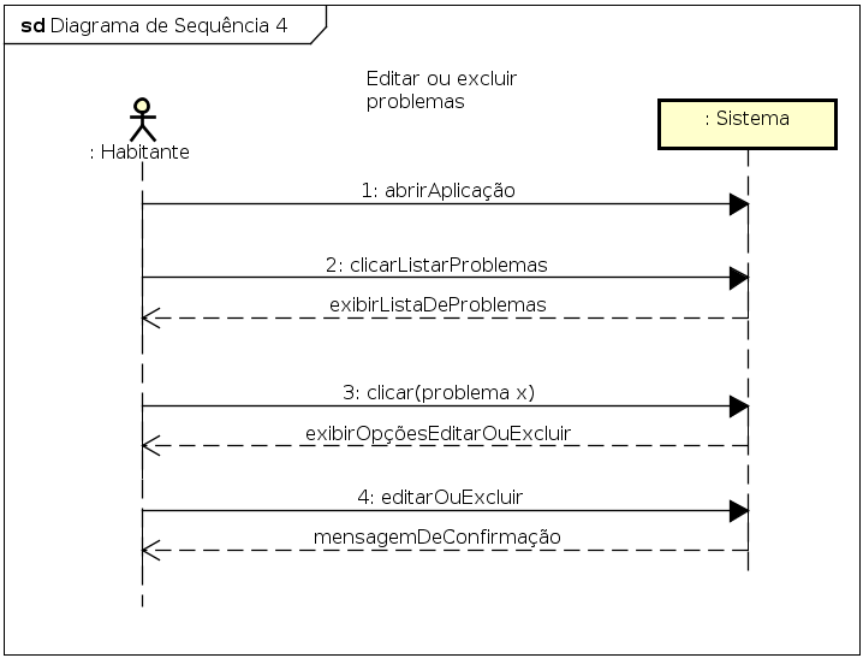
Fonte: Elaborado pelo autor

Figura 5 – Diagrama de Sequência de Sistema - Visualizar detalhes do problema



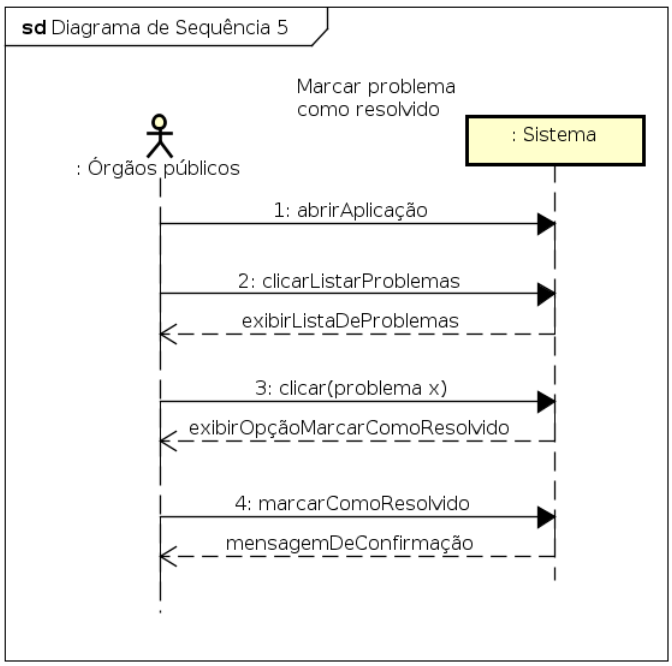
Fonte: Elaborado pelo autor

Figura 6 – Diagrama de Sequência de Sistema - Editar ou excluir problemas



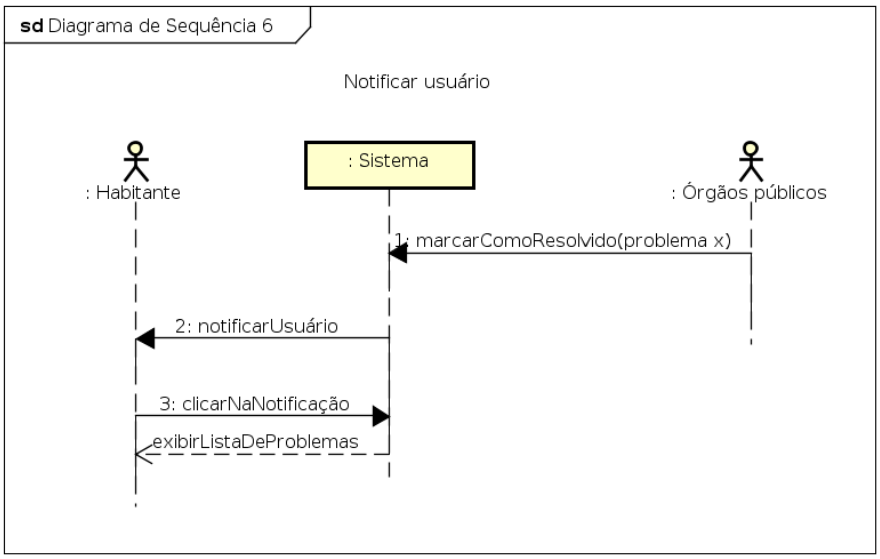
Fonte: Elaborado pelo autor

Figura 7 – Diagrama de Sequência de Sistema - Marcar problema como resolvido



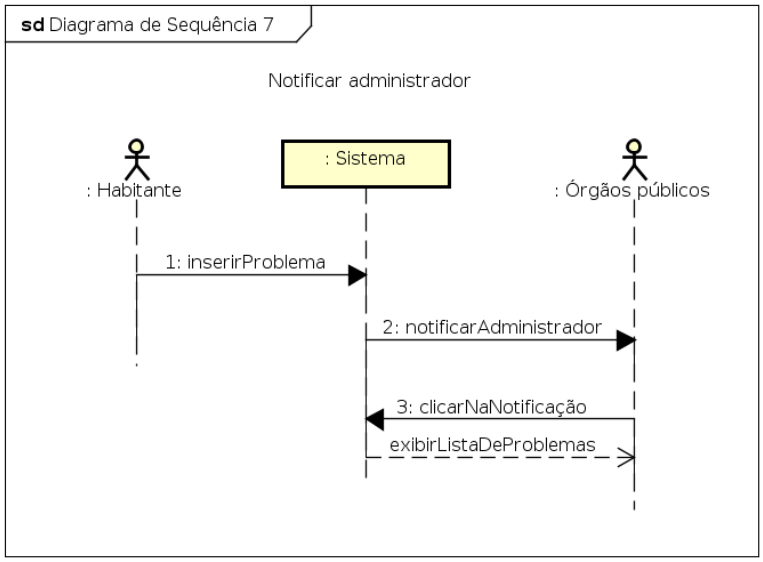
Fonte: Elaborado pelo autor

Figura 8 – Diagrama de Sequência de Sistema - Notificar usuário



Fonte: Elaborado pelo autor

Figura 9 – Diagrama de Sequência de Sistema - Notificar administrador



Fonte: Elaborado pelo autor

3.7 Administradores

A presente seção apresenta a lógica utilizada para o desenvolvimento do aplicativo destinado aos órgãos responsáveis. Primeiramente foi definido que não deveria haver apenas um administrador, devido à grande quantidade de problemas que a cidade poderia ter, o que acabaria sobrecarregando esse único administrador, pois o mesmo seria responsável de mandar executar a resolução de todos os problemas. Sendo assim, a estratégia tomada foi a de elaborar um sistema que permitisse que houvesse múltiplos administradores, no qual cada um deles seria responsável por um conjunto específico de problemas, e cada conjunto seria composto de problemas com particularidades em comum ou fossem relacionados.

Percorrendo todos os bairros da cidade de João Monlevade-MG utilizando um automóvel, foi possível identificar vários problemas. A partir do que foi observado foram definidas as 8 categorias as quais os problemas eram mais frequentes, que são listadas abaixo:

- Crimes
- Estradas
- Iluminação
- Lixo
- Meio ambiente
- Transporte público
- Água e esgoto
- Outros

Essas categorias buscam agrupar os problemas mais comuns de serem encontrados. Além disso, a categoria "outros" permite ao habitante cadastrar algum problema que por ventura não pertença a nenhuma outra categoria específica. O objetivo principal é dividir a carga de problemas entre vários administradores, sendo assim, o ideal seria que houvesse um administrador para cada categoria de problema. Outra vantagem nessa abordagem é que pessoas ligadas a empresas responsáveis por problemas de uma determinada categoria poderiam ser os administradores. Por exemplo, uma pessoa que trabalha na empresa responsável pela coleta de lixo poderia ser administrador da categoria "Lixo". Isso facilitaria o trabalho do administrador visto que estaria mais próximo e poderia fiscalizar no processo de resolução dos problemas cadastrados.

Além disso, como algumas categorias podem estar correlacionadas em alguns aspectos, como por exemplo "Meio ambiente" e "Lixo", um administrador pode ser

responsável por uma ou mais categorias. Da mesma forma, é possível que haja mais de um administrador para uma mesma categoria.

Em suma, administradores responsáveis por uma determinada categoria só podem:

- Marcar como resolvido problemas referentes a categoria.
- Visualizar a lista de problemas nessa categoria.
- Receber notificação de problemas cadastrados nessa categoria.

Através da conta do *Google* é verificado se o usuário é ou não um administrador, caso seja, o aplicativo irá apresentar todas as funcionalidades que são exclusivas de administradores. As contas dos administradores são armazenadas no banco de dados. A Seção 3.8 mostra como está a estrutura desses dados.

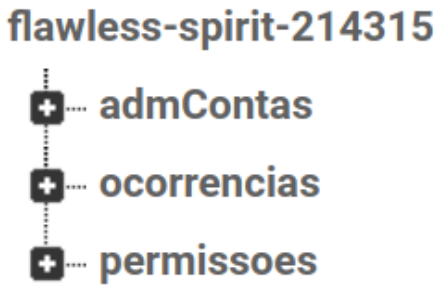
3.8 Banco de Dados

Como já descrito anteriormente, o banco de dados utilizado nessa aplicação foi o "Firebase Realtime Database". Tal banco de dados é NoSQL e utiliza também o formato JSON como estrutura, visando dessa maneira que haja um fluxo rápido de dados entre cliente e servidor. Como é um banco de dados NoSQL, não possui tabelas, portanto os dados são armazenados em uma árvore JSON. Os dados formam os nós desta árvore e cada nó possui uma chave própria de acesso. O *Firebase* possui um console que permite visualizar e manipular os dados contidos no banco de dados de forma direta. A Figura 10 apresenta esses dados e como estão armazenados.

A estrutura é bem simples, com apenas três nós principais, onde um deles é responsável por conter as contas de todos os administradores, outro por armazenar todas as ocorrências cadastradas e por último o nó que guarda informações a respeito das permissões que cada administrador possui. A Figura 10a evidencia cada um deles.

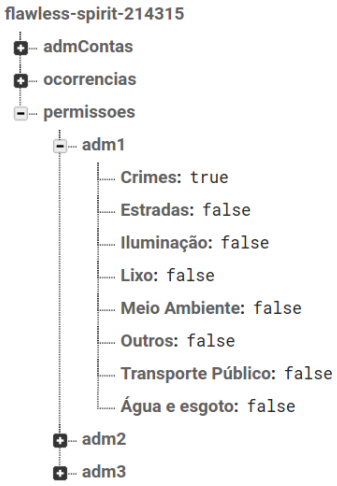
A Figura 10b expande o nó “admContas”, responsável pelas contas dos administradores, e mostra seus nós filhos. Nesse caso há 8 administradores, onde apenas o *e-mail* de cada administrador é armazenado. Este dados são consultados no momento do login para identificar se o usuário que está acessando a aplicação é ou não um administrador, e caso seja, funcionalidades específicas dos administradores são liberadas.

A Figura 10c expande o nó responsável por informações referentes às permissões. Quando um novo administrador é adicionado ao nó “admContas”, um novo nó é adicionado ao nó “permissões” também, com o intuito de especificar por quais categorias de problemas aquele administrador é responsável, e dar as devidas permissões ao mesmo. Portanto, cada nó filho do nó permissões possui 8 campos, referentes às categorias de problemas. O valor booleano *true* é atribuído ao campo pelo qual o administrador deve ser responsável.

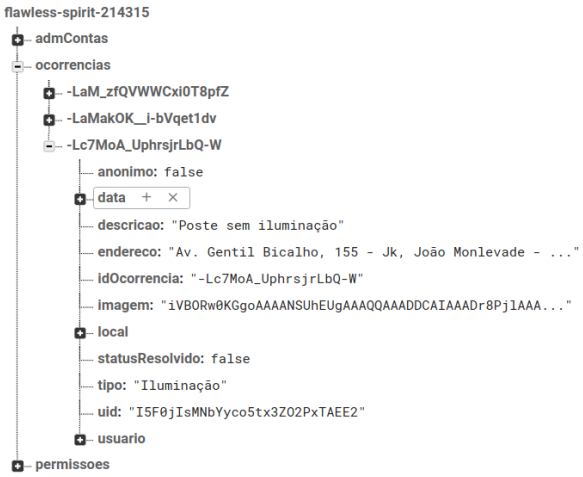


(a) Nós principais

(b) Filhos do nó "admContas"



(c) Filhos do nó "permissões"



(d) Filhos do nó "ocorrências"

Figura 10 – Console do *Firebase*

E por fim a Figura 10d mostra o nó “ocorrencias”, que armazena todas as ocorrências cadastradas. Cada ocorrência possui uma chave única como identificador, que é uma sequência de caracteres. É possível observar que uma ocorrência possui então os seguintes atributos:

- Anônimo: tipo *boolean*, indicando se a ocorrência cadastrada será anônima ou não.
- Data: tipo *Date*, contendo informações sobre horário e data em que a ocorrência foi visualizada.
- Descrição: tipo String, contendo simples descrição da ocorrência do problema.
- Endereço: tipo String, contendo o endereço do problema.
- IdOcorrência: tipo String, sendo o identificador da ocorrência.

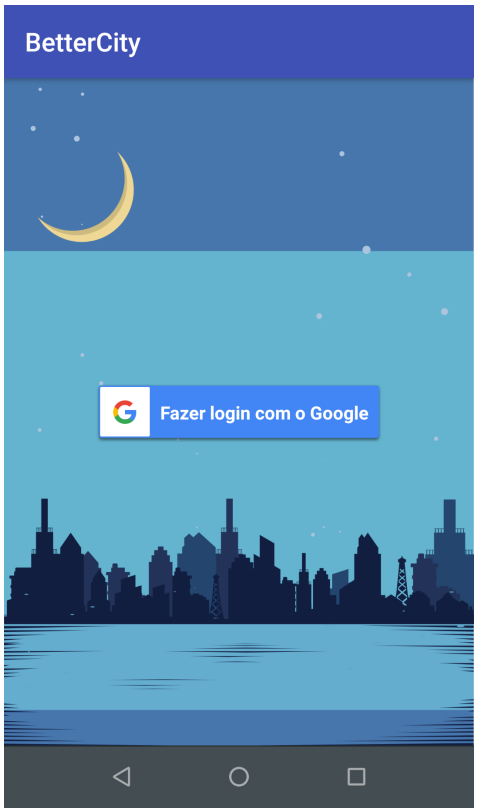
- Imagem: tipo String, contendo uma codificação dos bits da foto registrada da ocorrência.
- Local: tipo objeto, possuindo os atributos latitude e longitude, ambos float. Responsável pela localização da ocorrência.
- StatusResolvido: tipo *boolean*, indicando se a ocorrência já foi resolvida ou não.
- Tipo: tipo String, especifíca a qual categoria a ocorrência pertence.
- Uid: tipo String, contém o identificador do usuário que cadastrou a ocorrência.
- Usuário: tipo objeto, contendo os campos *e-mail*, foto e nome, sendo todos do tipo String.

4 Análise dos resultados

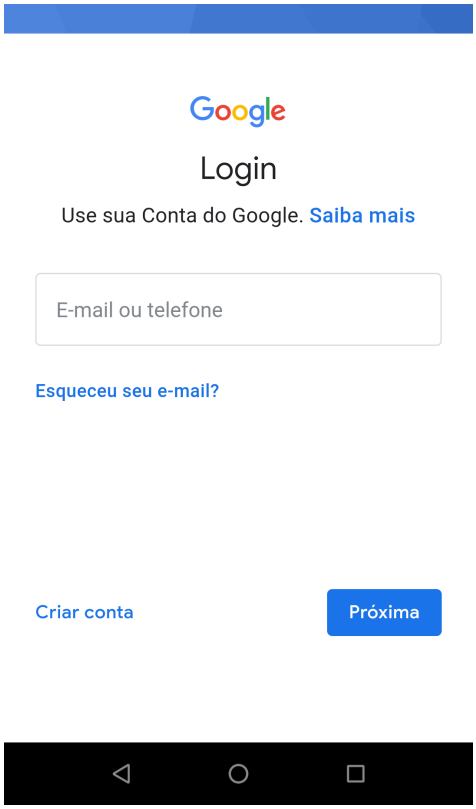
Neste capítulo serão apresentados os resultados obtidos, bem como os testes realizados. Tanto habitantes quanto órgãos responsáveis possuem papel ativo, com funcionalidades destinadas a ambos, sendo tratados como usuários e administradores, respectivamente.

4.1 Funcionalidades gerais

Esta seção apresenta as funcionalidades gerais da aplicação e é destinada tanto aos usuários quanto para administradores. A Figura 11 mostra a funcionalidade de login, requisitando que usuários e administradores acessem a aplicação utilizando a conta do *Google*, para dessa maneira serem identificados. A Figura 11a é a tela inicial da aplicação, após clicar no botão presente na tela, será exibida a tela mostrada na Figura 11b. Caso já tenha sido feito login e a opção "sair" não tenha sido clicada, usuários e administradores são automaticamente direcionados para o mapa da cidade, que é apresentado na Figura 12.



(a) Tela inicial de login



(b) Utilização da conta do Google para login

Figura 11 – Telas de login

A opção "sair" é exibida após clicar no botão de "três pontos verticais" que fica no canto direito superior da tela. Esta tela é bem intuitiva, contendo principalmente o

mapa da cidade com os ícones indicando os problemas. Além disso, possui um campo de busca para encontrar endereços específicos, que é localizado logo acima do mapa, e um botão para centralizar o mapa na atual posição geográfica do usuário, este situado no canto direito superior do mapa. Os usuários e administradores podem navegar por todo o mapa da cidade e observar por exemplo, onde estão os problemas, quais os bairros mais afetados ou a categoria de problema que é mais frequente. É importante também destacar que apenas problemas pendentes aparecem no mapa.

Cada categoria de problema possui seu ícone específico, como é possível perceber na Figura 12, na qual são apresentados alguns deles. Ao clicar sobre alguns desses ícones uma nova tela é aberta, como é possível ver na Figura 13. Através dela são exibidos todos os detalhes da ocorrência, como endereço, data e hora, descrição, usuário e foto.

Figura 12 – Visualização do mapa

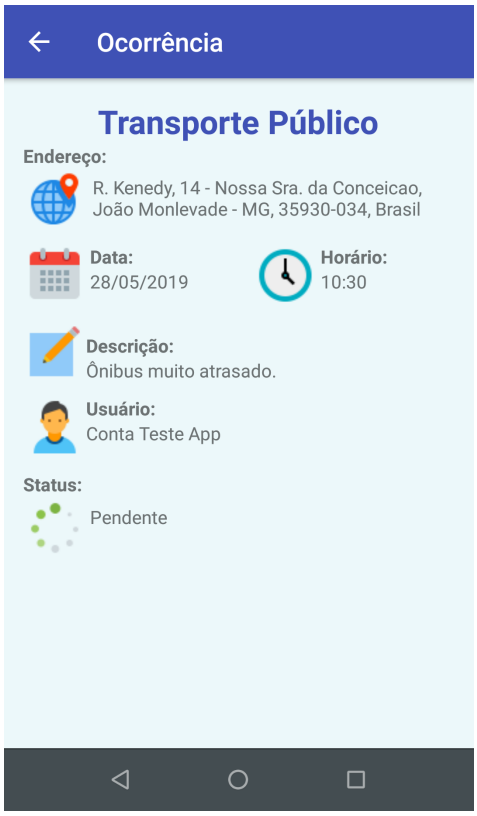


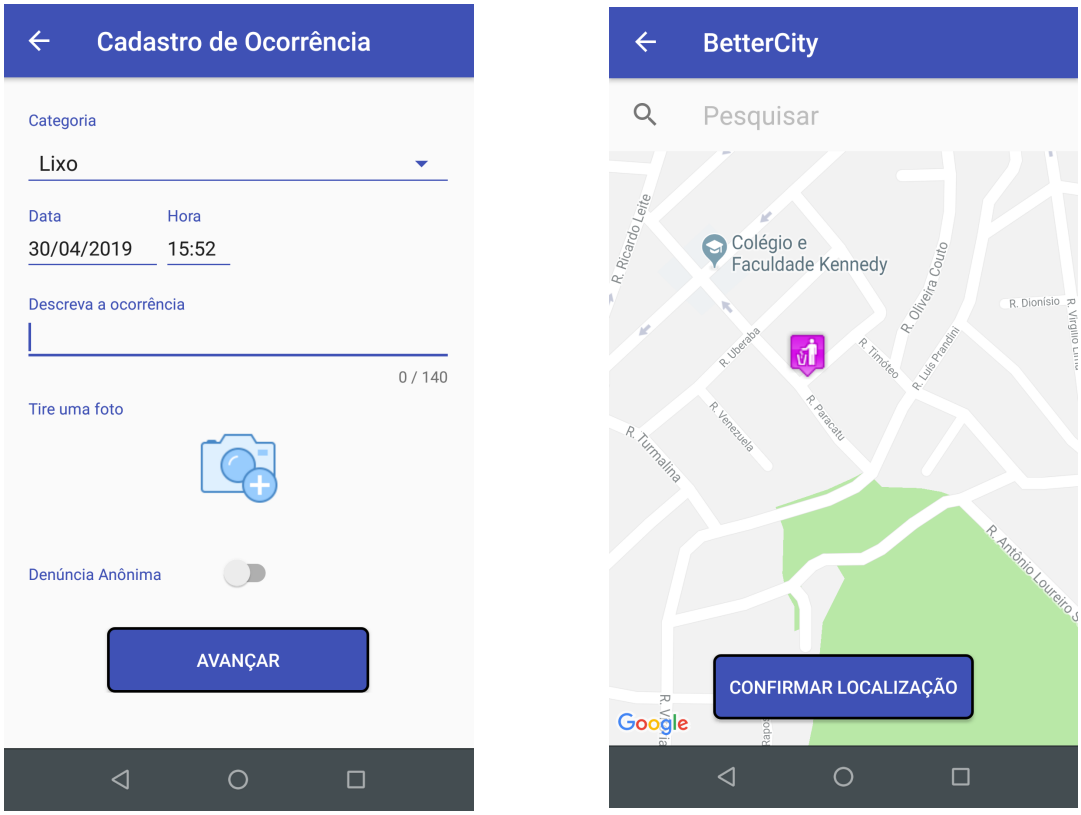
Figura 13 – Detalhes do problema

4.2 Funcionalidades relacionadas aos usuários

Essa seção apresenta as funcionalidades destinadas aos usuários. A Figura 14 mostra a primeira delas, que ocorre em duas etapas. A primeira etapa é informar os dados referentes à ocorrência, como pode ser visto na Figura 14a. Esta tela pode ser acessada clicando no botão circular que se encontra no canto direito inferior da tela na Figura 12. O usuário pode então um selecionar sua categoria, marcar a data e hora, descrever e tirar

uma foto, se desejar. O usuário também tem a opção de não se identificar, selecionando a opção "Denúncia anônima".

A segunda etapa é informar a localização do problema. Ao clicar no botão "Avançar" o usuário é direcionado para a tela da Figura 14b na qual pode clicar em algum lugar no mapa, ou arrastar o ícone para o lugar desejado. O ícone a princípio é posicionado no lugar em que o usuário está naquele momento, utilizando seu sinal de GPS. Além disso, o ícone que aparecerá será referente à categoria selecionada na tela anterior. Após clicar em "Confirmar localização" uma mensagem de confirmação é exibida e o processo de cadastro de ocorrência é concluído. Esta é a principal funcionalidade por parte dos habitantes, pois é onde de fato os problemas encontrados na cidade são denunciados.



(a) Cadastro de ocorrência (b) Confirmar Localização

Figura 14 – Telas de cadastro de ocorrência

A outra funcionalidade é a de exibir os dados das ocorrências já cadastradas pelo próprio usuário. A tela da Figura 15 é acessada ao clicar no ícone de "+" presente na Figura 12, no canto superior direito da tela. Nela uma lista é exibida informando todas as ocorrências e alguns dados adicionais, como a categoria do problema, que é indicada pelo ícone e também pelo título, o endereço, data e horário. Além disso, através dessa funcionalidade é possível acompanhar os status das ocorrências, verificando se as mesmas já foram solucionadas ou não.

Figura 15 – Ocorrências cadastradas pelo usuário

Outra funcionalidade destinada aos usuários é a de exclusão e edição de ocorrências. Quando uma ocorrência é cadastrada por um determinado usuário, o mesmo pode excluir ou editá-la se assim desejar. Essa funcionalidade é importante pois informações equivocadas podem ser inseridas. Outro caso possível é o de cadastro de ocorrências duplicadas, onde o mesmo problema é denunciado por dois ou mais usuários. Sendo assim, a aplicação dá autonomia para que os próprios usuários possam corrigir esses casos.

Para editar ou excluir um determinada ocorrência o usuário deve acessar a tela que exibe informações a respeito da mesma, como mostrado na tela da Figura 13. As opções de editar e excluir serão exibidas conforme mostrado na Figura 16. Lembrando que há duas formas de acessar essas informações, a primeira é através da tela de "Minhas ocorrências", da Figura 15. A segunda é clicando nos ícones do mapa, mostrado na Figura 12. Obviamente, a segunda opção só permite tal funcionalidade se a ocorrência tiver sido cadastrada pelo próprio usuário.

Outro ponto importante é que uma ocorrência só pode ser editada ou excluída caso ainda esteja com status pendente. Foi definido dessa forma para que os administradores tenham um maior controle dos problemas que já foram resolvidos, podendo até mesmo contabilizá-los.

Figura 16 – Opções de edição e exclusão

Ao selecionar a opção "Editar", a tela da Figura 17 é mostrada ao usuário. O processo de edição é semelhante ao de cadastro de ocorrência. Portanto, após clicar em "Avançar" o usuário deverá novamente passar pela tela de confirmação de localização e só então confirmar a edição.

A exclusão funciona de forma mais simples. Ao clicar na opção de "Excluir", a mensagem da Figura 18 aparecerá ao usuário. Caso afirmativo, a exclusão é realizada e uma mensagem de êxito é exibida.

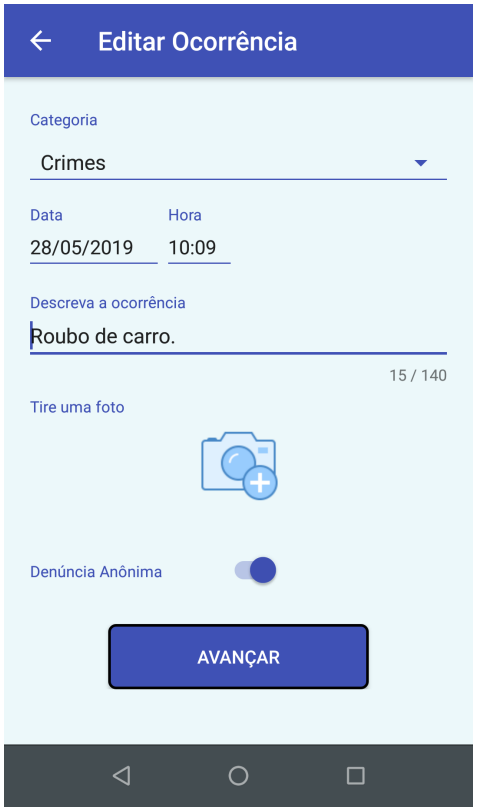


Figura 17 – Edição de ocorrência

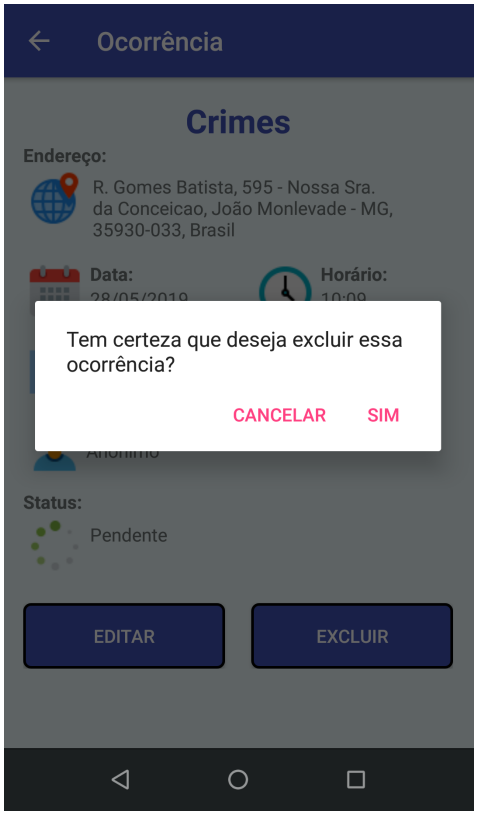
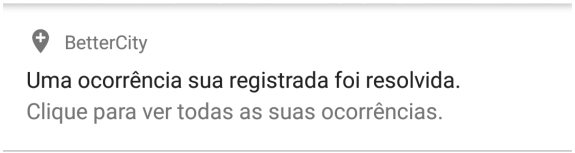


Figura 18 – Exclusão de ocorrência

E por fim a funcionalidade que notifica o usuário caso a ocorrência cadastrada pelo mesmo seja resolvida. A Figura 19a mostra o ícone da notificação, sendo o primeiro ícone do lado esquerdo, contendo um sinal de "+" em seu interior. Ao expandi-lá aparecerá conforme mostrado na Figura 19b. Ao clicar sobre a notificação o usuário é direcionado para a tela de "Minhas ocorrências" da Figura 15, onde pode verificar qual ocorrência foi marcada como resolvida.

(a) Notificação



(b) Notificação expandida

Figura 19 – Notificando habitantes

4.3 Funcionalidades relacionadas aos administradores

Esta seção apresenta as funcionalidades dedicadas aos administradores. A primeira delas é a de marcar ocorrências como resolvidas. Como já foi explicado na Seção 3.7, os administradores só podem marcar ocorrências que pertençam ao setor responsável pelo mesmo. Sendo assim, caso o administrador acesse a tela da Figura 13 para uma determinada

ocorrência, e essa condição seja válida, a opção "Marcar como resolvida" estará disponível conforme é mostrado na Figura 20. Ao clicar neste botão o status da ocorrência mudará de "pendente" para "resolvida" e o usuário que cadastrou a mesma será notificado. Esta é portanto a mais importante funcionalidade por parte dos administradores, onde acontece de fato o processo de resolução de ocorrências no sistema.

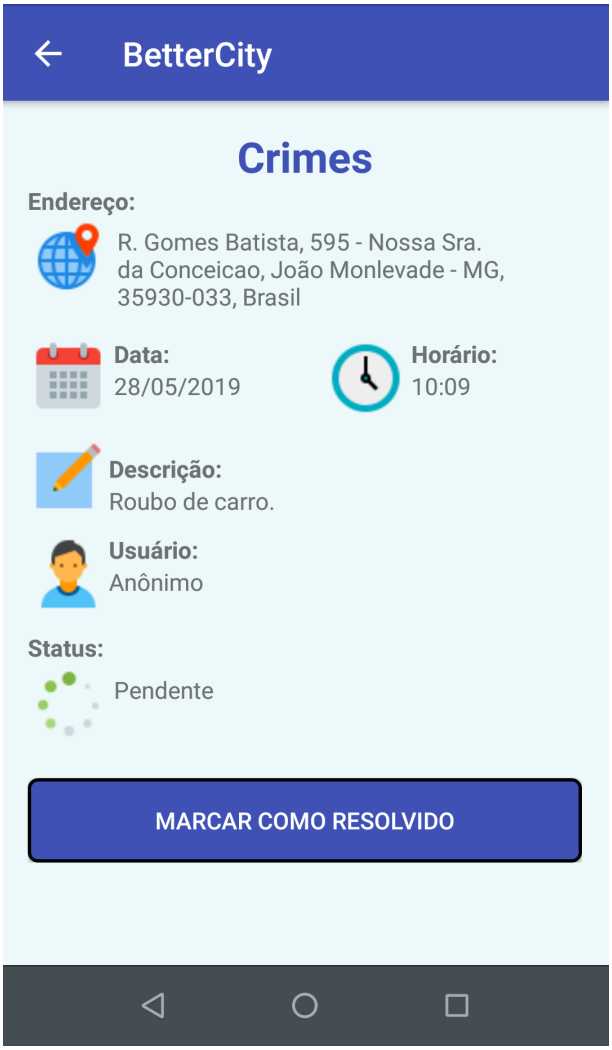


Figura 20 – Marcar problema como resolvido

Quando um usuário é identificado como administrador, dois novos botões surgem na tela inicial, um contendo a imagem do planeta Terra, e outro com uma imagem de um gráfico de linhas, ambos localizados entre os botões de "três pontos verticais" e o botão com sinal de "+" como é apresentado na Figura 21. O botão representado pelo planeta Terra dá acesso a tela mostrada na Figura 22, que contém todas as ocorrências cadastradas e que pertençam a área de atuação do administrador. No exemplo da Figura 22 o administrador é responsável pela categoria de crimes. Através dessa tela é possível acompanhar quais ocorrências ainda estão pendentes e ao clicar sobre algum problema dessa lista é possível acessar a tela da Figura 20 para marcar os problemas como resolvidos.

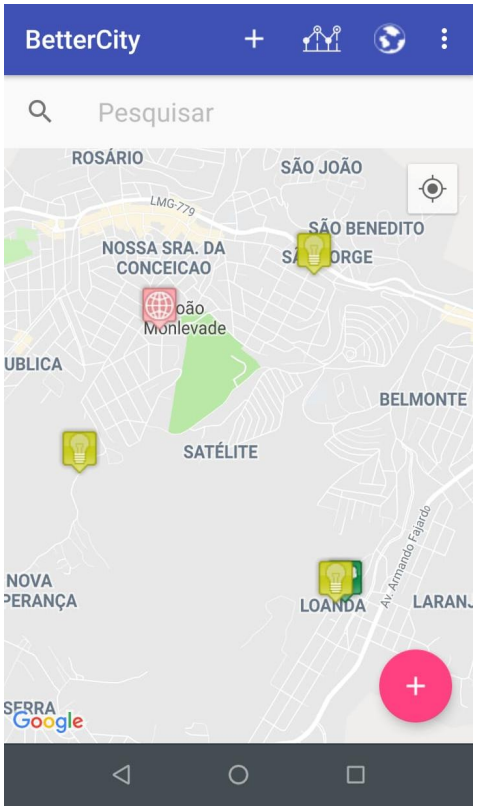


Figura 21 – Botões adicionais para administradores

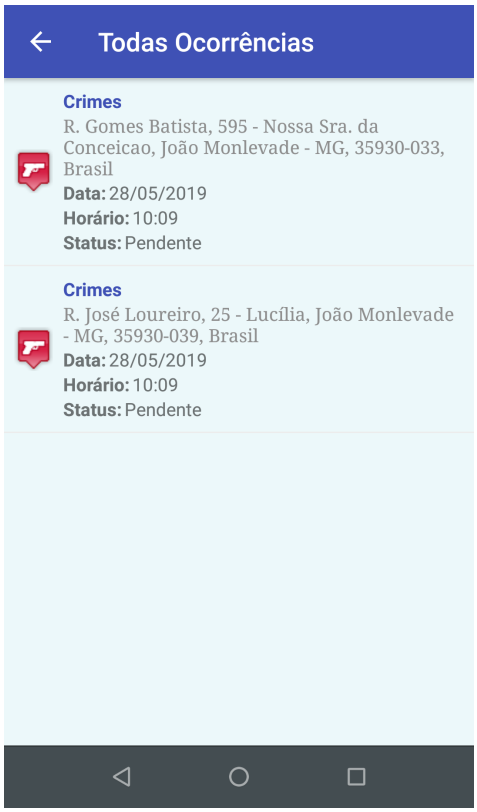


Figura 22 – Todas ocorrências

O outro botão direciona o administrador para tela que é apresentada na Figura 23. Esta tela é responsável por dar acesso a dados estatísticos. Tais dados são importantes pois podem ajudar a identificar fatores de impacto como os bairros mais afetados, indicando quais carecem de mais estrutura e investimento, por exemplo. Ou mesmo analisar a eficiência dos administradores e órgãos responsáveis na etapa de resolução de problemas. Além de outros dados que podem ser alvo de estudo e análise.

A Figura 24 apresenta um gráfico de setores com a porcentagem de ocorrências cadastradas por categoria de problema, além de apresentar o número bruto de ocorrências cadastradas por categoria. Esses dados não distinguem as ocorrências resolvidas das não resolvidas, logo apresenta dados de todas que já foram cadastradas até o momento.

A Figura 25 mostra um gráfico de barras que fornece o número de ocorrências cadastradas em cada bairro. Assim como o anterior, este apresenta dados de todas as ocorrências cadastradas.

E a Figura 26 exibe vários gráficos de barras apresentando a relação entre ocorrências cadastradas e resolvidas por categoria. Através das abas localizadas na parte superior da tela é possível navegar entre os gráficos de cada categoria.

Para criação dos gráficos apresentados neste trabalho foi utilizada a biblioteca gráfica “MPAndroidChart” (JAHODA, 2019).

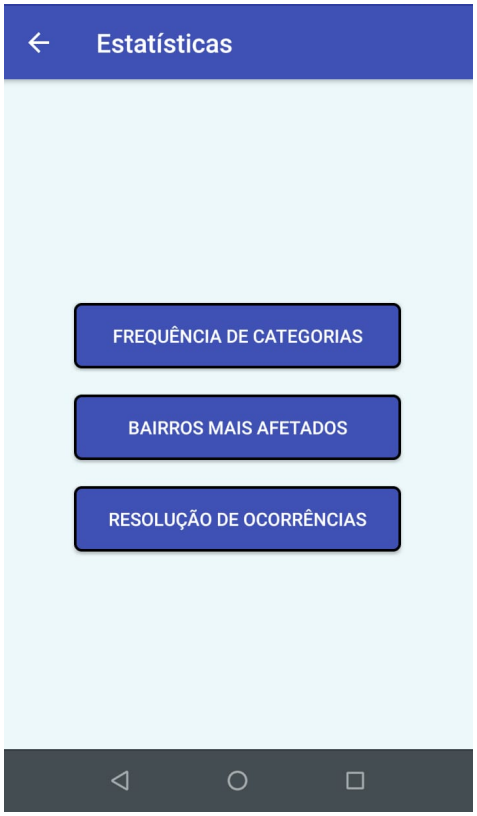


Figura 23 – Tela de estatísticas

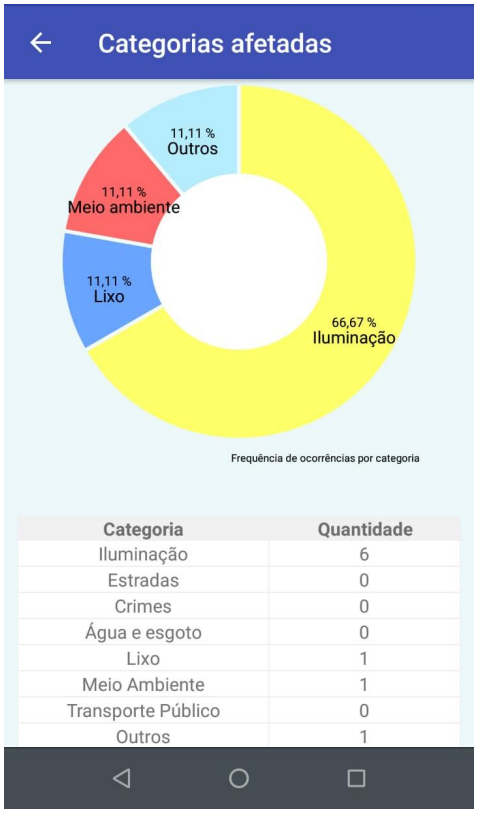


Figura 24 – Frequência das categorias afetadas

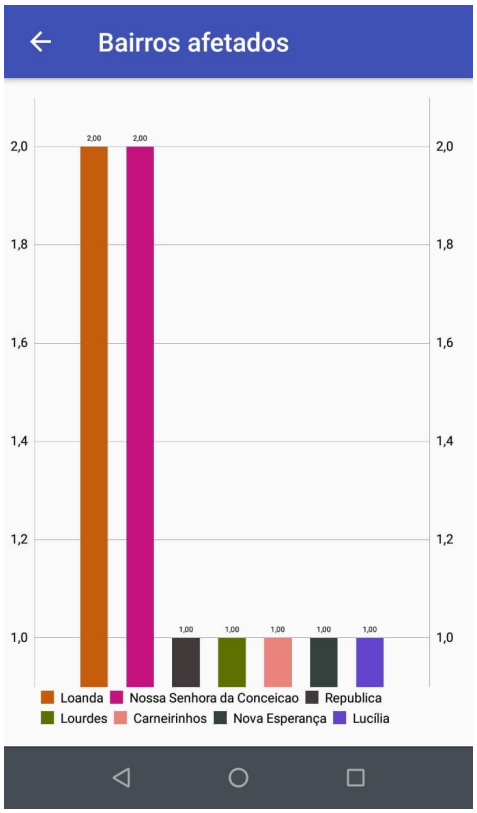


Figura 25 – Bairros afetados

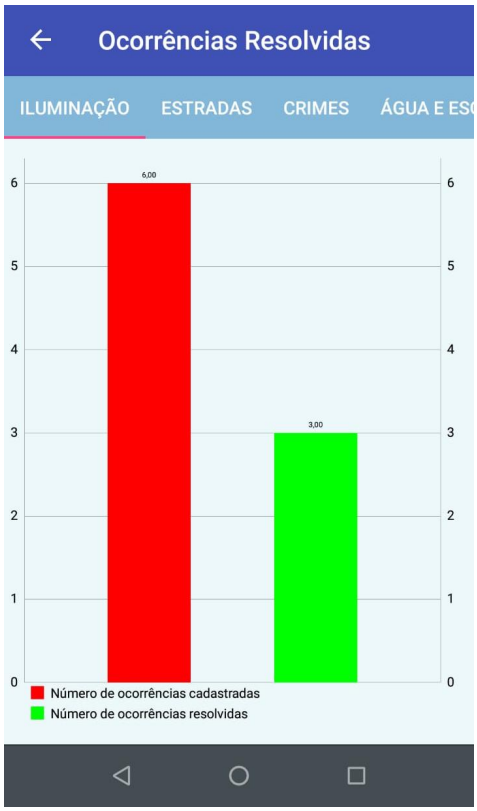


Figura 26 – Resolução de ocorrências

A outra funcionalidade é a de notificação. Um administrador será notificado quando alguma ocorrência for cadastrada e possuir categoria na qual o administrador é responsável, como também é descrito na Seção 3.7. O processo de notificações dos administradores é semelhante ao dos usuários, sendo o ícone o mesmo da Figura 19a. Porém, ao expandi-lá uma mensagem diferente aparecerá como mostrado na Figura 27. Ao clicar sobre a notificação o administrador é direcionado para a tela da Figura 22.

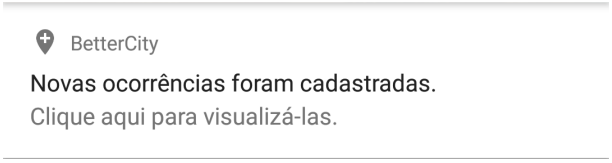


Figura 27 – Notificando administradores

4.4 Ausência de sinal de GPS ou internet

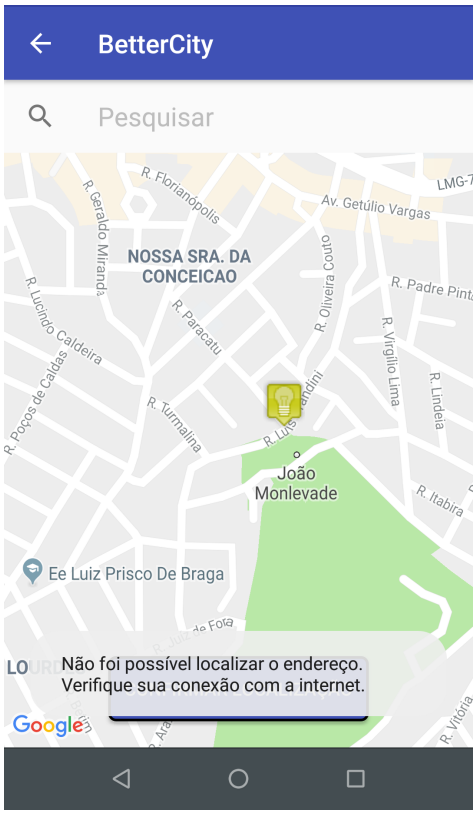
Para que a aplicação funcione de forma completa, possibilitando que os usuários e administradores utilizem todas as suas funcionalidades sem restrição são necessários sinal de GPS e internet. Caso não haja, algumas funcionalidades podem ficar limitadas. Como por exemplo, na ausência de sinal de internet não é possível utilizar o campo de busca de endereço e consequentemente a mensagem da Figura 28a aparecerá na tela do dispositivo.

Na funcionalidade de cadastrar uma nova ocorrência, no momento de selecionar a localização do problema também é necessário ter internet, pois é utilizado uma API da *Google* para localizar o endereço marcado no mapa. Caso não haja sinal de internet a mensagem da Figura 28b é exibida na tela e não é possível concluir o processo de cadastro.

As ocorrências cadastradas, bem como os dados de cada uma delas, são exibidos normalmente mesmo não havendo sinal de internet, pois como já dito anteriormente, o *Firebase* utiliza um banco de dados local no dispositivo e os sincroniza com os dados armazenados na nuvem quando houver sinal de internet.

O sinal de GPS também é importante durante várias etapas, servindo para localizar endereços ou a posição atual do usuário, por exemplo. Portanto, é requerido que a conexão GPS esteja habilitada no dispositivo do usuário logo após o processo de *login*. Caso não esteja a caixa de diálogo da Figura 29 é aberta. Ao clicar no botão "Abrir configurações de GPS" o usuário é direcionado para a tela de configuração de GPS do dispositivo, onde pode habilitar a conexão.

(a) Localização de endereços



(b) Confirmação de localização

Figura 28 – Ausência de sinal de internet

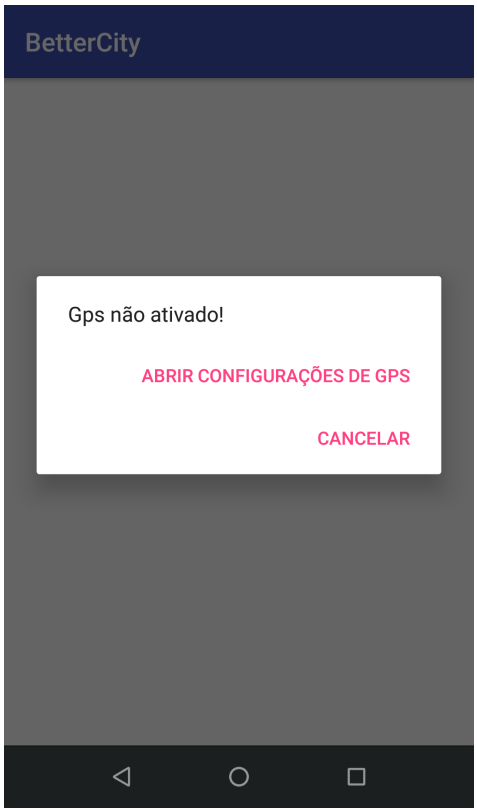


Figura 29 – Mensagem de GPS desabilitado

4.5 Inserção de problemas que não pertencem a João Monlevade-MG

Como a princípio o sistema proposto é focado apenas na cidade de João Monlevade-MG, é necessário que todos os problemas inseridos pertençam aos seus limites territoriais. Portanto, ao tentar inserir um problema que não respeite essa condição, a mensagem da Figura 30 é exibida ao usuário, não permitindo que o mesmo avance sem antes selecionar uma localização válida.

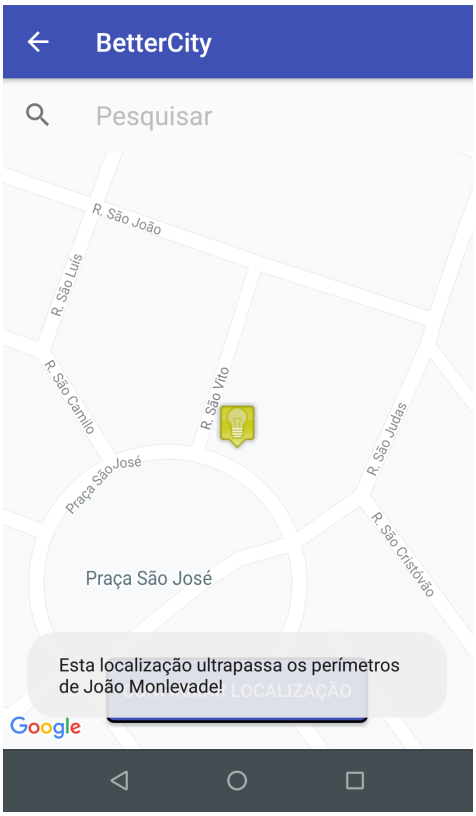


Figura 30 – Problema inserido ultrapassa os limites territoriais

4.6 Testes realizados

Para realizar os testes do sistema em questão foram utilizados dois dispositivos, um *smartphone* Motorola Moto G5S Plus com a versão Android 8.1.0, com tela de 5.5 polegadas e um tablet Samsung Galaxy Tab 2 com versão Android 4.1.2, com tela de 7 polegadas.

Esses dois dispositivos foram utilizados com o intuito de verificar como as telas criadas se comportariam em dispositivos com dimensões diferentes. Os resultados obtidos em ambos os dispositivos foram satisfatórios neste contexto, apresentando telas consistentes e intuitivas para os usuários e administradores.

Além disso, como o *tablet* possui uma versão bem mais antiga do sistema operacional, diferentemente do *smartphone*, esse teste permitiu identificar como a aplicação executaria em versões bastante distintas. O resultado alcançado foi o esperado neste contexto, não apresentando qualquer tipo de erro de execução durante os testes.

Contas de administradores e usuários comuns foram utilizadas em ambos os dispositivos para testar todas as funcionalidades, onde todas funcionaram conforme esperado. Além disso, ao observar o banco de dados do console do *Firebase* através do navegador *Google Chrome*, foi possível perceber que as alterações de dados causadas por alguma ação no aplicativo refletia praticamente em questão de milissegundos no banco de dados, confirmando assim a eficiência do *Firebase* na manipulação dos dados na nuvem. Essa eficiência também foi percebida ao executar a aplicação de forma simultânea em ambos os dispositivos. Ao alterar algum dado na aplicação em um dispositivo as alterações rapidamente apareciam na aplicação executando no outro dispositivo.

A parte das notificações também foi testada desta forma, sendo uma conta de administrador sendo executada em um dispositivo e uma conta de usuário comum em outra. Uma ocorrência foi adicionada na conta do usuário comum e a notificação prontamente apareceu no dispositivo executando a conta do administrador. No momento em que a ocorrência foi marcada como resolvida no dispositivo do administrador, uma notificação imediatamente apareceu no dispositivo do usuário.

5 Conclusão e trabalhos futuros

Com o trabalho desenvolvido foi possível concluir que o sistema proposto pode operar de forma eficiente e satisfatória na cidade de João Monlevade-MG. O sistema pode então fornecer um apoio às autoridades públicas na localização e resolução dos problemas que assolam a cidade, visto que assim como grande parte das cidades brasileiras, João Monlevade carece de ajuda, de suporte, para enfrentar essa realidade.

Como um dos objetivos durante toda a etapa de desenvolvimento foi a criação de um sistema simples e intuitivo, o mesmo possui então baixo custo operacional e exige poucos recursos dos dispositivos, permitindo deste modo que habitantes com uma menor condição financeira possam utilizar dispositivos mais baratos, com recursos limitados. Além disso, com pouco tempo de uso já é possível aprender a utilizar todas as funcionalidades.

Foi possível notar também que as estratégias tomadas durante o desenvolvimento do sistema, bem como as tecnologias e ferramentas utilizadas, foram acertadas, visto que agilizaram o processo de construção e forneceram o resultado esperado. Embora um teste utilizando uma carga real não pôde ser realizado, é esperado que o sistema se comporte bem, pois possui um banco de dados em nuvem e comunicação cliente/servidor utilizando uma tecnologia já consolidada no mercado.

Sendo assim, o sistema pode ser adotado pela cidade imediatamente, sendo necessário apenas contar com o apoio dos órgãos responsáveis, como a prefeitura por exemplo, para selecionar os administradores responsáveis.

Em relação aos trabalhos futuros pode-se citar:

- Melhorias das funcionalidades da aplicação. Na parte das estatísticas por exemplo, adição de mais dados como ranking de administradores, ocorrências cadastradas por tempo, usuários que mais contribuíram, entre outras.
- Adição de novas funcionalidades que podem melhorar a qualidade da aplicação. Como exemplo, poderia ser criado um filtro de exibição dos problemas no mapa. Apenas problemas de determinada categoria exibidos, ou apenas os problemas que já foram resolvidos, ao invés dos pendentes, são alguns exemplos de filtro que poderiam ser adicionados. Além disso, poderia haver uma avaliação dos usuários, aumentando a credibilidade daqueles que cadastram problemas que de fato são encontrados pelos órgãos responsáveis e desqualificando aqueles que por ventura criam problemas inexistentes, e dessa forma, tentar bloquear o cadastro de ocorrências falsas.
- Novas categorias de problemas podem ser adicionadas para uma melhor divisão,

ou mesmo para não sobrecarregar a categoria "Outros". Um exemplo disso seria a criação da categoria "Perda de documentos", destinada a resolver casos em que pessoas acabam perdendo alguma documentação como RG ou CPF, por exemplo.

- Criação de uma versão *web* para administradores acessarem dados e manipulá-los através do *browser*, contendo geração de relatórios, controle de usuários, entre outros serviços.
- Criação de uma versão multiplataforma com *React Native* ou *Flutter* com o intuito de permitir a execução da aplicação em outros sistemas operacionais além do *android*.
- Expansão do serviço para outras cidades.

Em suma, pode-se concluir que os objetivos propostos nesse trabalho foram alcançados com êxito e as sugestões de futuros trabalhos podem gerar melhorias ao sistema de forma a prover mais benefícios aos seus usuários.

Referências

4RSISTEMAS. *Cidadão Online 4R*. 2017. <https://play.google.com/store/apps/details?id=br.com.sistemas4r.CidadaoOnline>. Acesso em: 18 Mar. 2019. Citado na página 17.

ANDROID. *Conheça o Android Studio*. Disponível em: <https://developer.android.com/studio/intro?hl=pt-br>. Acesso em: 23 Mar. 2019. Nenhuma citação no texto.

ANDROID. *Developer Guides*. Disponível em: <https://developer.android.com/guide>. Acesso em: 25 Mar. 2019. Nenhuma citação no texto.

DEITEL, H. M.; DEITEL, P. *Java como programar. 10ª edição. Ed.* [S.l.]: Editora Bookman, 2016. Citado na página 20.

DELLOITE. *Global Mobile Consumer Survey 2018 - A mobilidade no dia a dia do brasileiro*. [S.l.], 2018. Disponível em: <http://images.e-mail.deloittecomunicacao.com.br/Web/DeloitteToucheTohmatsuAuditoresIndepende/%7B72ce37ea-28e5-4756-9669-5e63800f0d31%7D_Global-Mobile-Consumer-Survey-2018-Deloitte-Brasil.pdf?utm_campaign=GMCS-download-OK&utm_medium=email&utm_source=Eloqua>. Acesso em: 14 mar. 2019. Citado na página 14.

FIREBASE. *Firebase helps mobile app teams succeed*. 2019. Disponível em: <https://firebase.google.com/?hl=pt-br>. Acesso em: 02 Abr. 2019. Citado na página 21.

JAHODA, P. *MPAndroidChart*. 2019. Disponível em: <https://github.com/PhilJay/MPAndroidChart>. Acesso em: 24 Jun. 2019. Citado na página 47.

MA, F. *What's new at Firebase Summit 2018*. 2018. Disponível em: <https://firebase.googleblog.com/2018/10/whats-new-at-firebase-summit-2018.html>. Acesso em: 02 Abr. 2019. Citado na página 21.

MAQHIN. *Vigilante App*. 2014. Disponível em: <https://play.google.com/store/apps/details?id=com.maqhin.vigilante>. Acesso em: 15 mar. 2019. Citado na página 17.

MEIRELLES, F. S. *29ª Pesquisa Anual do Uso de TI*. [S.l.], 2018. Disponível em: <https://eaesp.fgv.br/ensinoeconhecimento/centros/cia/pesquisa>. Acesso em: 14 mar. 2019. Citado na página 14.

PEGN, R. *Aplicativos usam tecnologia a favor de causas socias*. [S.l.], 2016. Disponível em: <https://revistapegn.globo.com/Startups/noticia/2016/08/aplicativos-usam-tecnologia-favor-de-causas-sociais.html>. Acesso em: 15 mar. 2019. Citado na página 17.

PRODABEL. *PBH APP*. 2018. Disponível em: <https://play.google.com/store/apps/details?id=br.pbh.appservicos>. Acesso em: 20 Mar. 2019. Citado na página 17.

RBS, G. *Pelas Ruas*. 2017. Disponível em: <https://play.google.com/store/apps/details?id=br.com.gruporbs.pelasruas>. Acesso em: 15 mar. 2019. Citado na página 17.

RIBEIRO, H.; VARGAS, H. C. Urbanização, globalização e saúde. *Revista USP*, n. 107, p. 13–26, 2015. Citado na página 14.

STATCOUNTER. *Android overtakes Windows for first time*. [S.l.], 2017. Disponível em: <http://gs.statcounter.com/press/android-overtakes-windows-for-first-time>. Acesso em: 14 mar. 2019. Citado na página 14.

TERMO DE RESPONSABILIDADE

Eu, **Jean Kenned Teodoro Jeronimo** declaro que o texto do trabalho de conclusão de curso intitulado "***Um aplicativo para gerenciamento de aspectos da qualidade do ambiente urbano.***" é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 3 de julho de 2019

Jean Kenned Teodoro Jeronimo

DECLARAÇÃO DE CONFORMIDADE

Certifico que o(a) aluno(a) **Jean Kenned Teodoro Jeronimo**, autor do trabalho de conclusão de curso intitulado "***Um aplicativo para gerenciamento de aspectos da qualidade do ambiente urbano.***" efetuou as correções sugeridas pela banca examinadora e que estou de acordo com a versão final do trabalho.

João Monlevade, _____ de _____ de _____.

_____

Prof. Dr. George Henrique Godim da Fonseca