

INSTITUTO FEDERAL DE SANTA CATARINA - IFSC
CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇOS
CST EM GESTÃO DA TECNOLOGIA DA INFORMAÇÃO

LUCAS BAGGIO REALE DOS REIS

PROJETO DE TRABALHO DE CONCLUSÃO DE CURSO:
DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA ZELADORIA URBANA

Florianópolis
2021

LUCAS BAGGIO REALE DOS REIS

DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA ZELADORIA URBANA

Trabalho de Conclusão de Curso apresentado Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, como parte dos requisitos necessários à obtenção do título de tecnólogo no curso superior de Tecnologia em Gestão de Tecnologia da Informação.

Orientador: Marcos André Pisching

Coorientador: Rayse Kiane

Florianópolis
2021

DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA ZELADORIA URBANA

LUCAS BAGGIO REALE DOS REIS

Este trabalho foi julgado adequado para obtenção do Título de Tecnólogo em Gestão da Tecnologia da Informação e aprovado na sua forma final pela banca examinadora do Curso Superior de Tecnologia em Gestão da Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis-SC, 12 de Agosto de 2021.

Banca Examinadora:

Prof. Marcos André Pisching
Orientador
Instituto Federal de Santa Catarina

Prof. Herval Daminelli, Mestre
Instituto Federal de Santa Catarina

Prof. Juliano Lucas Goncalves, Mestre
Instituto Federal de Santa Catarina

AGRADECIMENTOS

Agradeço primeiramente a Deus, por sempre estar presente em minha vida, por ter me dado força, proteção, sabedoria e saúde durante minha caminhada nesta vida.

Aos meus pais, Alessandro Reale e Zuniane Baggio, por todo apoio, conselho e amor ao longo destes anos.

Aos meus irmãos Gabriel Baggio e Leticia Baggio pela amizade, companheirismo e lealdade.

Aos meus amigos, colegas de serviço, colegas de curso e professores pelos momentos de auxílio, aprendizado e alegria, principalmente aos meus orientadores Marcos André Pisching e Rayse Kiane, por sua dedicação, disponibilidade e conhecimento, essenciais para a conclusão desta etapa.

A todos vocês, minha gratidão!

Quer você acredite que consiga fazer uma coisa ou não, você está certo.

Henry Ford

RESUMO

Uma das formas de melhorar a eficiência da gestão pública está em ampliar a participação dos cidadãos no processo de tomada de decisões públicas. A modernização da gestão pública está diretamente ligada a transformação digital, já que a transformação digital traz mais qualidade aos serviços, além de possibilitar integração, convergência, transparência, segurança e privacidade para os dados pessoais. Com base nisso, este trabalho apresenta o desenvolvimento de um aplicativo cujo objetivo principal é aproximar cidadãos de seus governos através de uma ferramenta de tecnologia. O aplicativo permite que usuários colaborem com questões do setor público através da criação de notificações referentes a problemas de infraestrutura urbana, na qual os serviços de infraestrutura são formados pelos sistemas de saneamento, transporte, energia e telecomunicação. Com o aplicativo se espera contribuir para um melhor desempenho da gestão pública e contribuir para a qualidade de vida dos cidadãos

PALAVRAS-CHAVE: Gestão Pública, Participação cidadã, Tecnologia, Aplicação.

ABSTRACT

One of the ways to improve the efficiency of public management is to increase citizen participation in the public decision-making process. The modernization of public management is directly linked to digital transformation, since digital transformation brings more quality to services, in addition to enabling integration, convergence, transparency, security and privacy for personal data. Based on this, this work presents the development of an application whose main objective is to bring citizens closer to their governments through a technology tool. The application allows users to collaborate with public sector issues by creating notifications regarding urban infrastructure problems, in which infrastructure services are made up of sanitation, transport, energy and telecommunication systems. The application is expected to contribute to a better performance of public administration and contribute to the quality of life of citizens

KEYWORDS: Public Management, Citizen Participation, Technology, Application

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 - Os desafios para a infraestrutura do Brasil | 15 |
| Figura 2 - Caracterização da Pesquisa | 27 |
| Figura 3 - Requisitos não funcionais | 32 |
| Figura 4 - Histórias de Usuário | 33 |
| Figura 5 - Diagrama de casos de uso do sistema | 33 |
| Figura 6 - Diagrama de classe..... | 34 |
| Figura 7 - Modelo Incremental..... | 35 |
| Figura 8 - Arquitetura da Aplicação..... | 35 |
| Figura 9 - Protótipo da tela Home | 36 |
| Figura 10 - Tela de Login | 37 |
| Figura 11 - Tela de cadastro..... | 38 |
| Figura 12 - Recuperação de senha | 38 |
| Figura 13 – Telas de introdução | 39 |
| Figura 14 - Tela Home..... | 40 |
| Figura 15 - Tela do Histórico | 40 |
| Figura 16 - Local da ocorrência..... | 41 |
| Figura 17 - Fotos da ocorrência | 42 |
| Figura 18 - Classificação da ocorrência..... | 43 |
| Figura 19 - Resumo da ocorrência | 43 |
| Figura 20 - Email Smart City | 44 |
| Figura 21 - Painel de Controle Goole Play | 45 |
| Figura 22 - Análise da versão publicada..... | 46 |
| Figura 23 - Testes fechados..... | 46 |

LISTA DE SIGLAS

| | |
|--------|--|
| FIA | Fundação Instituto de Administração |
| G1 | Grupo Globo |
| GPS | Global Position System (USA) |
| GSMA | Sistema Global para Comunicações Móveis |
| HD | Hard Disk |
| HTML | HyperText Markup Language |
| IDE | Integrated Development Environment |
| IFSC | Instituto Federal de Santa Catarina |
| MVC | Modew-View-Controller |
| RAM | Random Access Memory |
| SDK | Software Development Kit |
| SEBRAE | Serviço Brasileiro de Apoio às Micro e Pequenas Empresas |
| SSD | Solid State Drive |
| TCC | Trabalho de Conclusão de Curso |
| UI | User Interface |
| UML | Linguagem de Modelagem Unificada |
| USP | Universidade De São Paulo |
| UX | User Experience |
| WEB | World Wide Web |

SUMÁRIO

| | | |
|----------|---|----|
| 1 | INTRODUÇÃO | 10 |
| 2 | JUSTIFICATIVA | 12 |
| 3 | PROBLEMA | 15 |
| 4 | OBJETIVOS | 18 |
| 4.1 | OBJETIVO GERAL..... | 18 |
| 4.2 | OBJETIVOS ESPECÍFICOS | 18 |
| 5 | REFERENCIAL TEÓRICO | 19 |
| 5.1 | ENGENHARIA DE SOFTWARE | 19 |
| 5.1.1 | Requisitos do Sistema..... | 19 |
| 5.1.1.1 | Requisitos funcionais | 20 |
| 5.1.1.2 | Requisitos não funcionais..... | 20 |
| 5.2 | DESENVOLVIMENTO DE SOFTWARE PARA DISPOSITIVOS MÓVEIS..... | 20 |
| 5.3 | UML..... | 22 |
| 5.3.1 | Diagrama de Casos de Uso..... | 22 |
| 5.3.2 | Diagrama de Classes..... | 22 |
| 5.4 | ARQUITETURA MVC | 23 |
| 5.5 | UX DESIGN | 23 |
| 5.6 | UI DESIGN | 24 |
| 5.7 | GIT..... | 24 |
| 5.8 | GITHUB..... | 25 |
| 6 | CARACTERIZAÇÃO DA PESQUISA | 26 |
| 7 | EQUIPAMENTOS E TECNOLOGIAS UTILIZADAS | 28 |
| 7.1 | EQUIPAMENTOS | 28 |
| 7.2 | ANDROID STUDIO | 28 |
| 7.3 | DART | 28 |
| 7.4 | FLUTTER..... | 29 |
| 7.5 | FIREBASE..... | 29 |
| 8 | DESENVOLVIMENTO | 31 |
| 8.1 | FASE DE DIAGNÓSTICO..... | 31 |
| 8.2 | LEVANTAMENTO E ANÁLISE DE REQUISITOS..... | 32 |
| 8.3 | DIAGRAMA DE CLASSES | 34 |
| 8.4 | PROCEDIMENTO METODOLÓGICO | 35 |
| 8.5 | ARQUITETURA DA APLICAÇÃO. | 35 |

| | | |
|-----------|---|-----------|
| 8.6 | PROTOTIPAÇÃO..... | 36 |
| 9 | Apresentação dos resultados: Telas do aplicativo | 37 |
| 9.1 | Login..... | 37 |
| 9.2 | Cadastro..... | 37 |
| 9.3 | Resetar Senha | 38 |
| 9.4 | Introdução ao aplicativo | 39 |
| 9.5 | Home..... | 39 |
| 9.6 | Histórico de notificações criadas..... | 40 |
| 9.7 | Criação de notificação | 41 |
| 9.7.1 | Tela 1: Localização da ocorrência | 41 |
| 9.7.2 | Tela 2: Fotos da ocorrência | 41 |
| 9.7.3 | Tela 3: Classificação da ocorrência..... | 42 |
| 9.7.4 | Tela 4: Resumo da ocorrência..... | 43 |
| 9.8 | Disparo de e-mail automático..... | 44 |
| 9.9 | Etapa de implantação | 44 |
| 10 | CONCLUSÃO..... | 48 |
| 10.1 | TRABALHOS FUTUROS..... | 48 |
| | REFERÊNCIAS..... | 49 |

1 INTRODUÇÃO

A infraestrutura é uma área essencial para o desenvolvimento socioeconômico de um país. Ela é composta pelos serviços de **saneamento, transporte, energia e telecomunicação**, todos contribuindo para o progresso e evolução de uma determinada região (FUNDAÇÃO INSTITUTO DE ADMINISTRAÇÃO, 2019).

Essas quatro áreas macro influenciam no deslocamento de pessoas e mercadorias e também no processo produtivo do local, o que resulta no crescimento econômico.

Segundo FIA (2019), podem ser definidas como:

- Saneamento: é a atividade relacionada ao abastecimento de água potável, o manejo de água pluvial, a coleta e tratamento de esgoto, a limpeza urbana, o manejo de resíduos sólidos e o controle de pragas e qualquer tipo de agente patogênico, visando à saúde das comunidades.
- Transporte: investimento em mobilidade urbana e na construção de estradas, aeroportos, ferrovias, hidrovias e portos.
- Energia: esse é um dos principais serviços de infraestrutura, tanto a geração quanto a distribuição dela.
- Telecomunicação: é essencial para a troca de informações entre pessoas e empresas de diversas localizações.

Entretanto, devido ao crescimento da população nas áreas urbanas, nas últimas décadas, a infraestrutura civil e econômica pública não está mais sendo capaz de atender a demanda necessária e conseqüentemente surgem problemas que acabam impactando na qualidade de vida dos habitantes (CARVALHO, 2018).

Os problemas são inúmeros, podendo ser vistos no cotidiano, como por exemplo, descarte irregular de lixo, esgoto a céu aberto, iluminação pública irregular, estradas esburacadas ou cheia de entulhos, pontos de água parada permitindo a proliferação da dengue, entre outros ocorrências.

A título de exemplo, a Organização Mundial da Saúde estima que anualmente 15 mil pessoas morram e 350 mil sejam internadas no Brasil devido a doenças ligadas à precariedade do saneamento básico. Cerca 35 milhões de brasileiros não têm

acesso à água tratada e metade da população não tem serviços de coleta de esgoto (JORNAL DA USP, 2020).

Devido a esses fatores, as cidades cresceram com vários problemas para a população, como déficit habitacional, serviços assistenciais urbanos de péssima qualidade, criminalidade, desigualdade social e entre outros (CARVALHO, 2018).

As tecnologias de informação e comunicação tem desempenhado um papel importante na comunicação coletiva, pois através dessa ferramenta a comunicação flui sem que haja barreira. Segundo Levy (1999), novas maneiras de pensar e de conviver estão sendo elaboradas no mundo da tecnologia.

Com a crescente evolução da internet e mídias sociais, o uso de dispositivos móveis, *smartphones* e *tablets* tem proporcionado uma revolução de maior impacto nos últimos tempos (SILVA; SANTOS, 2014). Para fins estatísticos, em todo o planeta, 5,1 bilhões de pessoas usam algum tipo de aparelho celular. O dado está no relatório a Economia Móvel 2019, da GSMA, empresa que representa o interesse de operadoras e avalia o ecossistema móvel. O número equivale a cerca de 67% da população mundial (VEJA, 2019).

Diante dos problemas apresentados, que serão melhor explicados no capítulo 3, esse trabalho tem como propósito a criação de um aplicativo para dispositivos móveis, uma ferramenta que possibilitará que usuários possam realizar notificações diretamente para a prefeitura local, diante dos problemas identificados pelo mesmo.

2 JUSTIFICATIVA

De acordo com uma pesquisa do Sebrae (2017, p.1), “Cidades inteligentes são aquelas que otimizam a utilização dos recursos para servir melhor os cidadãos. Isso vale para a mobilidade, a energia ou para qualquer serviço necessário à vida das pessoas”, ou seja, utilizam a tecnologia de modo estratégico para melhorar a infraestrutura.

Com o crescimento das cidades, torná-las inteligentes é essencial, seja por meio de tecnologias avançadas, por intermédio de projetos simples, mas que mudam completamente a vida da população.

A pesquisa reforça que a união entre universidades, empresas e instituições públicas é essencial para que as soluções sejam aplicadas e sirvam os cidadãos de forma cada vez mais eficaz (SEBRAE, 2017).

Portanto, o desenvolvimento deste aplicativo ajudará a mitigar problemas relacionados a infraestrutura de certa região, contribuindo para que estas cidades se tornem mais sustentáveis e inteligentes.

Estes problemas estão diretamente ligados as quatro áreas macros da infraestrutura urbana, sendo elas: Saneamento básico, Transporte, Energia e Telecomunicações.

Foi realizado um mapeamento desses problemas usando como base aplicativos com propostas semelhantes, principalmente Qzela e Colab, o capítulo 8 apresenta informações adicionais referente a estes aplicativos. A tabela 1 representa as categorias e suas subcategorias.

Tabela 1: Categorias e suas subcategorias

| CATEGORIAS | SUBCATEGORIAS |
|------------------------------|--|
| Água e Esgoto | Bueiro entupido Bueiro sem tampa Esgoto a céu aberto Ponto de alagamento Vazamento de água Falta de água |
| Iluminação e Energia | Fiação irregular, Falta de energia Lâmpada apagada à noite Iluminação pública irregular |
| Limpeza e conservação | Lixeira quebrada Entulho na calçada/via pública Descarte irregular de lixo Praia suja Mato alto |
| Meio Ambiente | Desmatamento irregular Aterro sanitário irregular Ponto de queimada irregular recorrente Poda de árvore Área com risco de deslizamento Retirada de árvore |
| Pedestres e Ciclistas | Faixa de pedestre apagada Calçada irregular Ponto de travessia irregular Calçada inexistente Rampa de acessibilidade irregular Ciclovia/ciclofaixa mal sinalizada |
| Saúde | Foco de mosquito da dengue/zika Ponto recorrente de poluição sonora Emissão de fumaça preta |
| Transporte Público | Ponto de ônibus danificado Ônibus/trem/metrô danificado Ônibus fora do horário/rota Má conduta de motorista ou cobrador" |
| Vias e Trânsito | Buraco nas via Ponto de infração de trânsito recorrente Placa de sinalização quebrada/inexistente Semáforo quebrado Veículo abandonado Bloqueio na via Via de terra com desnível |

Fonte: Elaborado pelo autor (2021)

Para melhor esclarecimento, as categorias servem como um filtro para o usuário se guiar, e as subcategorias indicam diretamente o tipo do problema, a Figura 20 mostra como isso foi estruturado no aplicativo.

Sendo assim, a população será capaz de colaborar com questões do setor público através de publicações de zeladoria urbana, auxiliando nas tomadas de decisões dos serviços públicos.

Publicações de zeladoria urbana são, em sua essência, problemas de infraestrutura identificados pelo usuário do aplicativo, onde ele poderá informar o local do problema identificado, tirar fotos e registrar comentários. Por fim, esta publicação será encaminhada para a prefeitura local.

Com a interação dos cidadãos e servidores públicos, as entidades públicas se tornarão mais eficientes e responsivas às demandas da sociedade. Essa interação torna os governos mais transparentes e abertos, já que mais acesso à informação o possibilita a trabalhar de forma mais responsiva.

Com base nisso, entende-se que este presente trabalho se mostra relevante em diferentes aspectos. Coincide com a missão do IFSC, que em sua essência tem o propósito de contribuir para o desenvolvimento socioeconômico e cultural, e visa também aperfeiçoar ainda mais os conhecimentos do autor em desenvolvimento de softwares e recursos para dispositivos móveis.

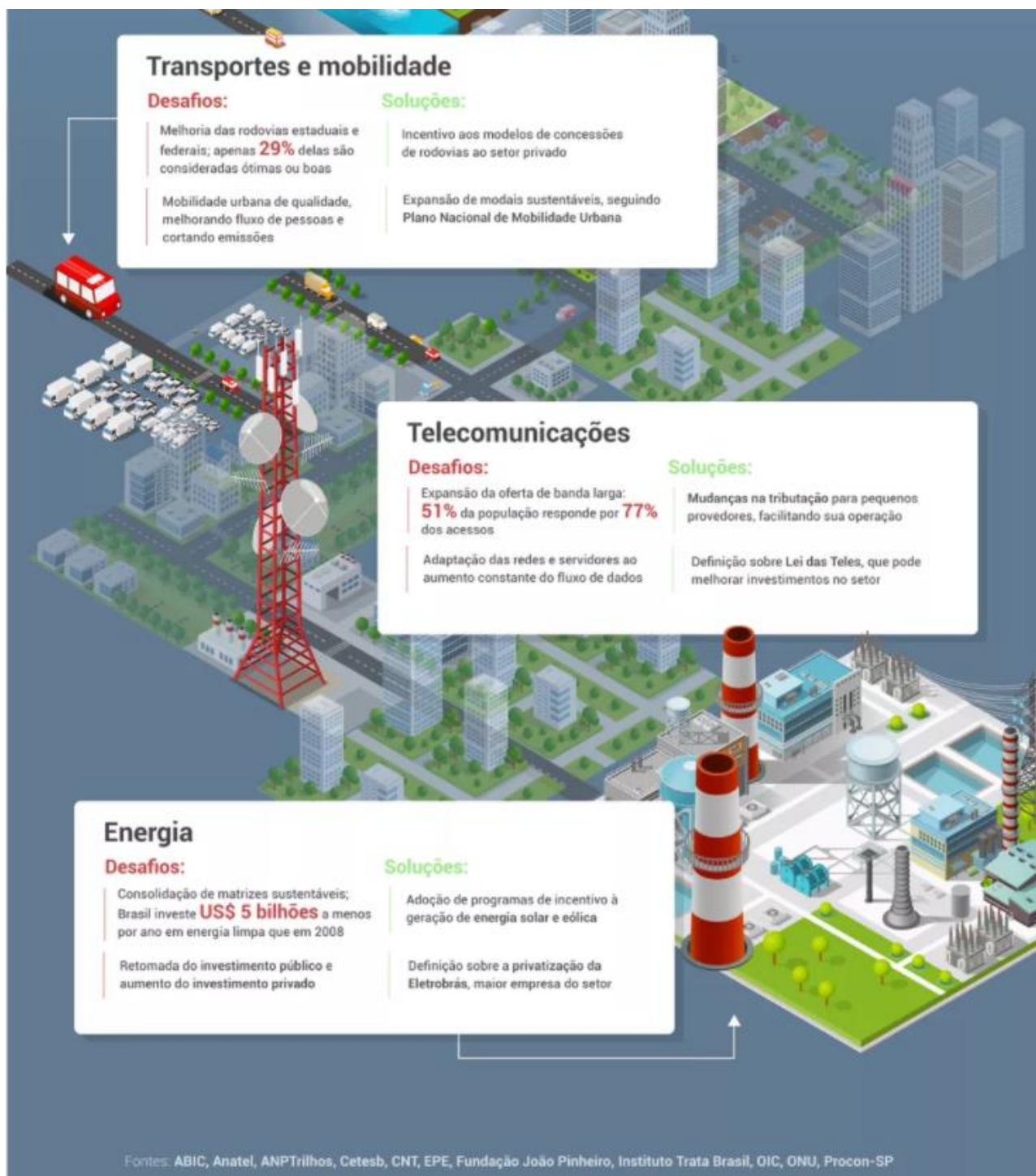
3 PROBLEMA

Conforme abordado no capítulo 1, a administração pública possui dificuldades para analisar e resolver todos estes problemas de infraestrutura, muitas vezes ocasionado pela falta de tecnologias que ajudem a localizar os pontos principais que devem ser tratados.

De acordo com a reportagem “Os desafios para a infraestrutura do Brasil”, o site G1(2018) apresenta as seguintes informações:

Figura 1 - Os desafios para a infraestrutura do Brasil





Fonte: G1(2018)

O Brasil possui déficits expressivos na sua infraestrutura. Esses déficits podem ser representados a partir de serviços precários de saneamento básico, pelo setor elétrico enfraquecido e uma infraestrutura de transportes problemática, que reduzem a competitividade e a produtividade da economia brasileira, sendo responsável também pela redução direta do bem-estar da população (G1, 2018).

De acordo com FIA (2019), um dos principais desafios de governo está no mapeamento dos problemas e definição de prioridades. Esse é um esforço que demanda planejamento e uso transparente dos recursos públicos.

Diante dos dados apresentados, neste trabalho é desenvolvido um aplicativo para dispositivos móveis para auxiliar a administração pública a equacionar problemas de infraestrutura, interligando servidores públicos e cidadãos.

4 OBJETIVOS

Os objetivos deste trabalho são divididos em duas sessões. A seguir são apresentadas cada uma delas.

4.1 OBJETIVO GERAL

O objetivo geral deste trabalho é desenvolver um aplicativo para dispositivos móveis que permita aos usuários auxiliarem na resolução de problemas de infraestrutura, através de publicações de zeladoria urbana.

4.2 OBJETIVOS ESPECÍFICOS

A partir do objetivo geral deste trabalho, formularam-se os seguintes objetivos específicos:

- Levantar um acervo de estudo a respeito de programação para dispositivos móveis.
- Realizar o mapeamento de problemas urbanos que envolvem a infraestrutura urbana.
- Realizar levantamento dos requisitos do sistema, necessários para desenvolvimento do aplicativo.
- Criar, testar e implementar protótipo desenvolvido.

5 REFERENCIAL TEÓRICO

Este capítulo apresenta conceitos e características importantes sobre todo o conteúdo a ser abordado, que serve de base para o desenvolvimento do trabalho.

5.1 ENGENHARIA DE SOFTWARE

De acordo com Pressman (2011, p. 48) “A engenharia de software engloba processos, métodos e ferramentas que possibilitam a construção de sistemas complexos baseados em computador dentro do prazo e com qualidade”. A prática da engenharia de software é uma atividade de resolução de problemas que segue um conjunto de princípios básicos.

Segundo Pressman (2011), o software distribui o produto mais importante da nossa era, a informação. Essas e muitas outras questões indicam a preocupação com o software e seus métodos de desenvolvimento, levando à adoção de práticas de engenharia de software.

A engenharia de software é guiada por uma série de princípios básicos que ajudam a aplicar processos de software significativo e na execução de métodos de engenharia de software efetivos (PRESSMAN, 2011). Estes princípios fundamentais estabelecem regras que servem como guias conforme se analisa um problema, projeta uma solução, implementa e testa uma solução e, por fim, estabelece o software na comunidade de usuários (PRESSMAN, 2011).

5.1.1 Requisitos do Sistema

A engenharia de requisitos fornece mecanismos apropriados para entender as necessidades do cliente, analisar requisitos, avaliar a viabilidade, negociar soluções razoáveis, especificar soluções sem ambiguidades e gerenciar requisitos à medida que são transformadas em um sistema (PRESSMAN, 2011).

Para Pressman (2011, p.127) “as tarefas abordadas pela engenharia de requisitos, são: concepção, levantamento, elaboração, negociação, especificação, validação e gestão”. É importante notar que algumas delas ocorrem em paralelo e todas são adaptadas às necessidades do projeto.

Do ponto de vista do processo de software, a engenharia de requisitos é uma operação importante da engenharia de software, que começa nas atividades de comunicação e continua nas atividades de modelagem. Deve se adaptar às necessidades do processo, do projeto, do produto e das pessoas envolvidas no trabalho (Pressman, 2011).

5.1.1.1 Requisitos funcionais

Os requisitos funcionais do sistema descrevem o que ele deve fazer. Eles dependem do tipo de software a ser desenvolvido, dos possíveis usuários e da abordagem geral adotada pela organização ao escrever os requisitos (SOMMERVILLE, 2011).

Em princípio, para Sommerville (2011), a especificação dos requisitos funcionais de um sistema deve ser íntegra e consistente. Integridade significa que todos os serviços exigidos pelo usuário devem ser definidos. Consistência significa que os requisitos não devem ter definições conflitantes.

5.1.1.2 Requisitos não funcionais

Sommerville (2011) define requisitos não funcionais da seguinte forma:

Os requisitos não funcionais são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários. Eles podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de área (SOMMERVILLE, 2011, p. 74).

Requisitos não funcionais, por exemplo: desempenho, proteção ou disponibilidade geralmente especificam ou limitam as características de todo o sistema. Requisitos não funcionais são geralmente mais críticos do que requisitos funcionais individuais (SOMMERVILLE, 2011).

5.2 DESENVOLVIMENTO DE SOFTWARE PARA DISPOSITIVOS MÓVEIS

Diante do crescimento exponencial do número de *smartphones* no mercado, o desenvolvimento de aplicativos móveis tornou-se uma nova oportunidade de negócios

para desenvolvedores e empresas. Cresce na mesma proporção a busca por aplicações que satisfazem as necessidades dos usuários. Segundo Ambros (2013), existem três tipos diferentes de aplicações para dispositivos móveis: híbridas, nativas e web.

- **Aplicativos Nativos:** são aplicativos instalados no dispositivo e adquiridos por meio de lojas, como Google Play, App Store e Amazon Underground¹. Seu desenvolvimento é específico para determinado sistema operacional, podendo aproveitar todos os recursos de hardware e de software presentes no dispositivo. Além do acesso fácil aos recursos do dispositivo, se o aplicativo for bem concebido, ele respeitará os padrões de design de cada plataforma, atendendo a padrões de usabilidade. Sua maior desvantagem é o fato de ser necessário desenvolver aplicativos específicos para cada sistema operacional.
- **Mobile Web App:** São aplicativos acessados por meio de um navegador, geralmente desenvolvidos em HTML 5, ou seja, não precisam ser instalados no *smartphone*. Seu acesso é realizado por meio de uma URL. Uma das principais desvantagens de um Web App é na dificuldade de utilizar todas as funcionalidades nativas de um dispositivo, como: câmera, *blueetooth*, GPS, *WiFi*, entre outros. São mais lentos que os aplicativos nativos por não serem integrados ao sistema operacional
- **Aplicações híbridas:** Não são desenvolvidas completamente na linguagem específica de cada sistema operacional. Nesse caso, ele utiliza várias linguagens e engloba dois formatos: metade nativo e metade *Web App*. Assim como os nativos, os aplicativos híbridos também podem ser baixados pelas lojas e utilizar as funcionalidades do dispositivo.

Neste projeto foi utilizado o *framework* Flutter. Um framework para desenvolvimento de aplicações híbridas, que utiliza a linguagem Dart como base de criação dos aplicativos. Apesar das características distintas em cada um dos modelos, a adoção do modelo híbrido se deu por consequência da escolha do framework Flutter. No capítulo 7 é apresentado em maiores detalhes as características dessas ferramentas tecnológicas citadas.

¹ Google Play, App Store e Amazon Underground são lojas virtuais de softwares para dispositivos móveis, onde usuários podem adquirir aplicativos para seus aparelhos.

5.3 UML

A UML (*Unified Modeling Language*) é uma linguagem padrão usada para desenvolver a estrutura de projetos de software. Ela define um número de diagramas que permite dirigir o foco para aspectos diferentes do sistema de maneira independente (BOOCH et al., 2012). Se bem usados, os diagramas facilitam a compreensão do sistema que está sendo desenvolvido.

De acordo com Silva et al. (2001), é uma linguagem utilizada para padronizar, construir, visualizar e registrar componentes de sistemas de software. A UML permite que a equipe de desenvolvedores de sistema visualize os resultados de seus trabalhos na forma de diagramas padronizados.

5.3.1 Diagrama de Casos de Uso

Na Linguagem de Modelagem Unificada (UML), o diagrama de caso de uso resume os detalhes dos usuários do sistema e as interações deles com o sistema, ou seja, em uma forma direta e concisa ela demonstra como o sistema é utilizado em passos, em suas respectivas ordens (LUCIDCHART, 2020).

De acordo com a empresa Lucidchart (2020), especialista em linguagem UML, o diagrama de caso de uso é ideal para:

- Representar as metas de interações entre sistemas e usuários;
- Definir e organizar requisitos funcionais no sistema;
- Especificar o contexto e os requisitos do sistema;
- Modelar o fluxo básico de eventos no caso de uso.

5.3.2 Diagrama de Classes

Diagramas de classes estão entre os tipos mais úteis de diagramas UML pois mapeiam de forma clara a estrutura de um determinado sistema ao modelar suas classes, seus atributos, operações e relações entre objetos (LUCIDCHART, 2020).

Entre os benefícios da utilização de diagramas de classe, segundo Lucidchart (2020), podemos destacar:

- Ilustração de modelos de dados para sistemas de informação;
- Melhor entendimento a visão geral dos esquemas de uma aplicação;
- Expressão visual das necessidades específicas de um sistema e a divulgação dessas informações por toda a empresa;
- Criação de gráficos detalhados que destacam qualquer código específico necessário para ser programado e implementado na estrutura descrita.

5.4 ARQUITETURA MVC

MVC é o acrônimo de *Model-View-Controller*. Trata-se de um padrão de projeto de software, ou padrão de arquitetura de software desenvolvido na década de 1970, com foco na reutilização de código e a separação de conceitos em três camadas interconectadas.

Este padrão é utilizado em muitos projetos devido à arquitetura que possui, o que possibilita a divisão do projeto em camadas muito bem definidas. Cada uma delas, o *Model*, o *Controller* e a *View*, executa o que lhe é definido e nada mais do que isso (DEV MEDIA, 2013).

O uso do padrão MVC traz o benefício de separar as regras de negócios da lógica de apresentação (interface do usuário). Isso permite múltiplas interfaces de usuário que podem ser modificadas sem alterar as regras de negócio, proporcionando assim maior flexibilidade e oportunidades de reutilização de classes (DEV MEDIA, 2013).

5.5 UX DESIGN

UX significa, literalmente, *User Experience*, isto é, Experiência do Usuário. Na prática, significa a relação que uma pessoa tem com um determinado produto ou serviço. A partir do momento em que o usuário se interessa, comenta, pesquisa e compra algo, isso tudo é *User Experience* (MEDIUM, 2019).

O UX Design é isso: se preocupa com cada etapa em que o usuário interage com o produto/serviço. É a combinação de empatia, usabilidade, tecnologia.

Dentre os benefícios proporcionados pelo UX, podemos destacar:

- Melhora na experiência do usuário com o produto;

- Fidelização à marca;
- Aumento na lucratividade;
- Melhora da imagem empresarial.

Em resumo, a área de UX se responsabiliza por realizar diversas pesquisas para entender mais sobre os usuários, seus costumes e necessidades.

5.6 UI DESIGN

UI Design, ou *User Interface Design* (Design de Interface do Usuário), é o meio pelo qual uma pessoa interage e controla um dispositivo, software ou aplicativo. Esse controle pode ser feito por meio de botões, menus e qualquer elemento que forneça uma interação entre o dispositivo e o usuário.

Segundo empresa RD Station (2019), os benefícios de um UI design bem projeto, são:

- Aquisição de novos clientes;
- Maior engajamento e retenção de clientes;
- Menores custos de desenvolvimento;
- Menores custos no apoio ao cliente (Suporte);
- Aumento da produtividade.

5.7 GIT

O GIT é um sistema *open-source* de controle de versão utilizado pela grande maioria dos desenvolvedores atualmente (*StackOverFlow*, 2019). Com GIT é possível criar todo histórico de alterações no código do projeto e facilmente voltar para qualquer ponto para saber como o código estava naquela data. (ROCKETSEAT, 2019).

GIT foi utilizado durante o desenvolvimento, em conjunto com GitHub. O uso dessas ferramentas permite maior segurança e flexibilidade para o projeto em questão.

Dentre as variadas funções do GIT, foi utilizado majoritariamente o **Controle do histórico**: facilidade em desfazer e possibilidade de analisar o histórico do desenvolvimento, como também facilidade no resgate de versões mais antigas e estáveis. Dessa forma, foi possível ter maior organização, segurança e flexibilidade no desenvolvimento do código fonte.

5.8 GITHUB

O Github é um serviço online de hospedagem de repositórios Git. Dessa forma é possível manter todos *commits* e *branches* sincronizados entre os membros de um time.

O Github também é um dos maiores depósitos online de trabalho colaborativo do mundo (STACKOVERFLOW, 2019), servindo como uma boa fonte para estudos e para tirar dúvidas.

6 CARACTERIZAÇÃO DA PESQUISA

Esse capítulo esclarece métodos e técnicas de pesquisas adotados para o desenvolvimento do trabalho científico.

Segundo Lakatos e Marconi (2003, p.17) “Pode-se definir pesquisa como o procedimento racional e sistemático que tem como objetivo proporcionar respostas aos problemas que são propostos.”

Um estudo é considerado científico quando são adotados métodos em sua realização (RIBAS, 2018).

Sendo assim, 4 critérios de classificação foram avaliados, sendo eles:

- Natureza
- Abordagem
- Objetivos
- Procedimentos

Em relação à natureza, o presente trabalho classifica-se como uma **pesquisa científica aplicada**, pois “contém finalidades imediatas, gera produtos e/ou processos (VIANA, 2013, p.1).

Quanto à abordagem, trata-se de uma **pesquisa qualitativa**, pois tem o ambiente natural como fonte direta de dados e o pesquisador como instrumento fundamental (VIANNA, 2013). Não foram utilizadas ferramentas estatísticas de análise de dados para o desenvolvimento do trabalho.

A Figura 2, demonstra as principais diferentes características entre uma pesquisa Qualitativa x Quantitativa.

Figura 2 - Caracterização da Pesquisa

Caracterização da Pesquisa

Pesquisa Qualitativa x Pesquisa Quantitativa

| | Pesquisa Qualitativa | Pesquisa Quantitativa |
|------------------------|--|---|
| Cenário | Familiar, natural | Desconhecido, artificial |
| Amostra | Pequena, não aleatória, teórica | Grande, aleatória, representativa |
| Coleta de dados | O investigador é instrumento primário, observações | Instrumentos inanimados (escalas, questionários...) |
| Análise | Indutiva (pelo investigador) | Dedutiva (por métodos estatísticos) |
| Resultados | Compreensivos, holísticos, expositivos | Precisos, limitados, reducionistas |

INSTITUTO FEDERAL SANTA CATARINA

105 ANOS
INSTITUTO FEDERAL DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

Fonte: IFSC (2018).

Em relação aos objetivos, trata-se de uma **pesquisa descritiva**, VIANNA (2013, p.1) define uma pesquisa descritiva como:

Registra e descreve os fatos observados sem interferir neles. Descreve as características de determinada população ou fenômeno, estabelece relações entre variáveis. Envolve o uso de técnicas padrões de coleta de dados: questionário e observação sistemática

Por fim, os procedimentos técnicos utilizados durante o ciclo de vida da pesquisa foram através de pesquisas bibliográficas.

7 EQUIPAMENTOS E TECNOLOGIAS UTILIZADAS

Este capítulo esclarece as ferramentas computacionais e tecnologias que foram utilizadas para o desenvolvimento do aplicativo.

7.1 EQUIPAMENTOS

Os equipamentos utilizados no desenvolvimento deste trabalho foram um Desktop, com as seguintes especificações: Processador Intel® Core™ i3-9100F, 8GB de Memória RAM, placa gráfica Gigabyte NVIDIA GeForce GTX 1050 Ti 2GB, SSD Kingston A400, 240GB, SATA, HD Seagate BarraCuda 1TB e sistema operacional Windows 10 Home.

7.2 ANDROID STUDIO

Android Studio é um ambiente de desenvolvimento integrado (IDE) para desenvolver preferencialmente para a plataforma Android, disponível para os sistemas operacionais Windows, Linux e Mac OS.

Alguns motivos que justificam sua escolha são: diversidade de atalhos disponíveis para agilizar o processo na digitação e edição de texto recursos para auto completar comandos pré-definidos, e interface amigável, além de ser facilmente utilizado em conjunto com frameworks, além de oferecer emuladores mobile.

7.3 DART

Para o desenvolvimento do aplicativo, foi utilizada a linguagem Dart, uma linguagem de programação fortemente tipada. Inicialmente criada pela Google em 2011. Ela hoje pode ser considerada uma linguagem multi-paradigma, embora a linguagem apresente fortes estruturas típicas de linguagens orientadas a objeto.

Dart possui uma sintaxe com estilo baseado na linguagem C. Isso faz com que sua sintaxe seja muito similar à linguagens atualmente populares, como Java e C#.

O Dart compila em código nativo e não há necessidade de acessar *widgets* OEM. Em suma, isso significa uma comunicação menos mediada entre o aplicativo e a plataforma. Esse é o único SDK para dispositivos móveis que fornece visualizações

reativas sem exigir uma ponte JavaScript (BLENDIT, 2019).

7.4 FLUTTER

O Flutter, um *framework* para o desenvolvimento de aplicativos mobile para Android e iOS, desenvolvido pelo Google, tendo seu primeiro lançamento em maio de 2017.

Ao criar um aplicativo com o Flutter, o código é compilado para a linguagem base do dispositivo, ou seja, as aplicações são realmente nativas e por isso conseguem acessar recursos do dispositivo sem a “ajuda” de terceiros e com o desempenho maior (TREINAWEB, 2020).

Diversas são as características do Flutter. Dentre elas podemos citar:

- Multiplataforma – Possível desenvolver aplicações com Flutter em qualquer sistema operacional (Windows, Linux e MacOS);
- Criação de aplicações nativas a partir de um único código base – Com o Flutter é possível desenvolver aplicações nativas para Android e iOS;
- Acesso direto aos recursos nativos do sistema – Uma aplicação criada com Flutter possui acesso nativo aos recursos do dispositivo (câmera, wifi, memória, entre outros);
- Maior desempenho – As aplicações criadas com Flutter possuem um maior desempenho quando comparadas ao React Native, por exemplo, pois todo código-fonte é transformado em código nativo.

7.5 FIREBASE

Firebase é uma plataforma de desenvolvimento mobile (e web) adquirida pela Google em 2014. Com finalidade em promover serviços para *backend*, essa ferramenta disponibiliza diversos serviços que auxiliam no desenvolvimento e gerenciamento de aplicativos (Firebase, 2019).

Estes serviços são gerenciados através de um *console web*, criado para facilitar a implementação. Neste, o desenvolvedor adiciona um projeto e inclui os serviços que desejar, cada um com uma explicação de como proceder. Nem todos os serviços são grátis, em sua maioria são limitados, é possível criar um plano conforme as necessidades do desenvolvedor.

Entretanto, os serviços grátis, são inúmeros, com diversas funcionalidades, sendo dignos de uma seção inteira para descrevê-los, porém como este não é o foco do trabalho, os mesmos podem ser acessados no link a seguir: <https://firebase.google.com/products-build>

De forma resumida, os serviços utilizados para a construção do aplicativo, foram:

- Cloud Firestore: Banco de dados (NoSQL) na nuvem;
- Cloud Storage: Banco de dados para armazenamento de imagens e vídeos;
- Firebase Authentication: Serviço para permitir autenticação de usuários através de diversos provedores de login, como: Google, Facebook, Twitter, Github, Telefone, Email/Senha. Para o projeto, foi utilizado o modo de autenticação E-mail/Senha.

8 DESENVOLVIMENTO

Este capítulo tem por objetivo apresentar as etapas do desenvolvimento do aplicativo.

O mercado de tecnologia faz com que cada vez mais novos métodos de desenvolvimento de software se destaquem. Mas, independentemente do número e da diversidade, todos seguem os mesmos **pilares** e **princípios** (DEV MEDIA, 2007), sendo eles:

1. Fase de Diagnóstico
2. Levantamento e Análise de requisitos
3. Fase de Desenvolvimento
4. Etapa de Implantação

8.1 FASE DE DIAGNÓSTICO

Nesta fase foi realizado um estudo detalhado do problema, os possíveis atores do aplicativo, soluções já existentes no mercado.

Para a criação de uma solução eficiente, é necessário que o problema seja analisado de forma profunda, pois quanto mais claro e nítidos estiverem as exigências do problema, mais completa e relevante será a solução (MOREIRA, 2018).

Com base nisso, foram analisadas as seguintes informações:

- Origem do problema
- Particularidades da situação atual do problema
- Prazo estipulado para a solução deste problema
- Exigências geradas por este problema
- Abrangência deste problema
- Urgência de uma solução para este problema

Após análise dos aspectos citados acima, foram definidos os atores do aplicativo.

Atores: Apenas cidadãos.

Por fim, foi realizado uma pesquisa de mercado, referente a soluções semelhantes já existentes, as quais serviram como fontes de informações e inspirações para o desenvolvimento do aplicativo.

Empresas com soluções semelhantes: **Qzela e Colab**.

Tais aplicativos permitem a criação de notificações relacionados a problemas de infraestrutura. Entretanto, abaixo é apresentado diferenças entre estes aplicativos e o aplicativo Smart City.

- Categorias e subcategorias de problemas apresentados em cada um dos aplicativos são diferentes.
- Validação do atendimento das notificações criadas ficam por responsabilidade do usuário, enquanto Qzela e Colab, pelo próprio aplicativo.
- Qzela não permite informar o endereço da ocorrência através de um campo de texto, ou seja, o endereço é obtido através da localização atual do usuário, isso pode ser um problema, caso o usuário queira criar a notificação posteriormente, não estando no local da ocorrência.
- Qzela e Colab não apresentam um histórico detalhado das notificações criadas, apenas um contador que informa a quantidade de “Aberturas” e “Fechamentos”, o histórico de notificações do aplicativo Smart City é apresentado no capítulo 9.6

Os aplicativos Qzela e Colab podem ser encontradas na Google Play com seus respectivos nomes.

8.2 LEVANTAMENTO E ANÁLISE DE REQUISITOS

Esta seção apresenta os requisitos funcionais e não funcionais do aplicativo. A Figura 3 representa os requisitos não funcionais do sistema.

Figura 3 - Requisitos não funcionais

| Requisitos Não Funcionais | |
|--|--|
| O aplicativo deve estar disponível 24h por dias, 7 dias por semana | |
| O aplicativo deve ter boa performance | |
| O aplicativo deve ser responsivo com diferentes tipos de dispositivos móveis | |
| O aplicativo será corrigido caso o usuário encontre algum erro de execução | |

Fonte: Elaborado pelo autor (2021)

Os requisitos funcionais, foram definidos a partir de histórias de Usuário, ou seja, cada história de usuário representa uma necessidade do usuário (Figura 4).

Figura 4 - Histórias de Usuário

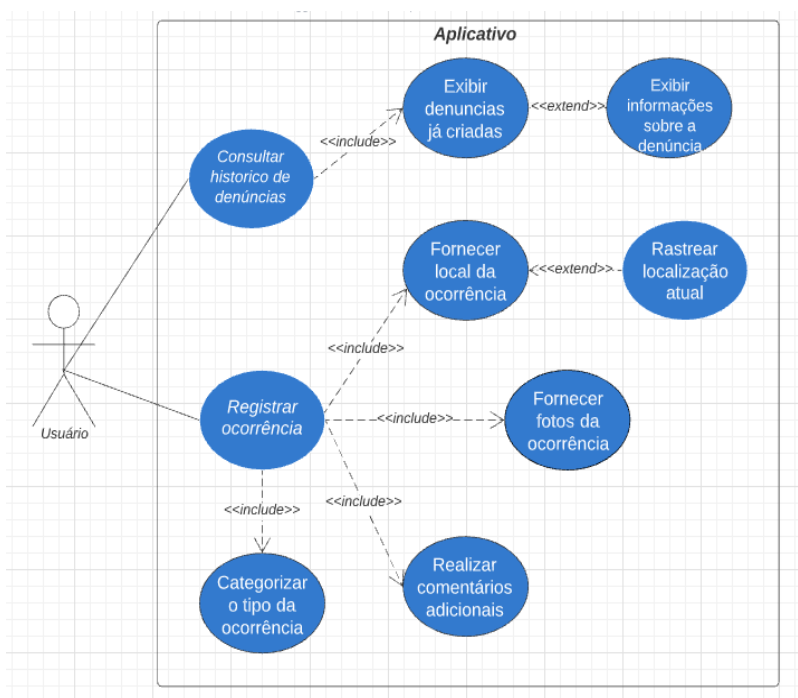
| Histórias de Usuário | |
|---|--|
| Como usuário, o aplicativo deve permitir que eu forneça a localização da ocorrência | |
| Como usuário, o aplicativo deve permitir que eu registre uma notificação | |
| Como usuário, o aplicativo deve permitir que eu forneça fotos da ocorrência | |
| Como usuário, o aplicativo deve permitir que possa fazer comentarios sobre notificação na sua criação | |
| Como usuário, o aplicativo deve permitir que eu tenha um historico das notificações já criadas. | |
| Como usuário, o aplicativo deve permitir que eu classifique a categoria da ocorrência | |
| Como usuário, o aplicativo deve permitir que eu acompanhe as atualizações da minhas notificações | |
| Como usuário, o aplicativo deve fornecer a geolocalização atual | |

Fonte: Elaborado pelo autor (2021)

Essas histórias de usuário servem como base para a criação de casos de uso, que tem como objetivo descrever o comportamento do sistema, para suprir as necessidades do usuário.

A Figura 5 representa como os casos de uso interagem entre si no sistema e com o usuário, ou seja, como as funcionalidades se relacionarão umas com as outras e como serão utilizadas pelo usuário, durante o uso do sistema.

Figura 5 - Diagrama de casos de uso do sistema

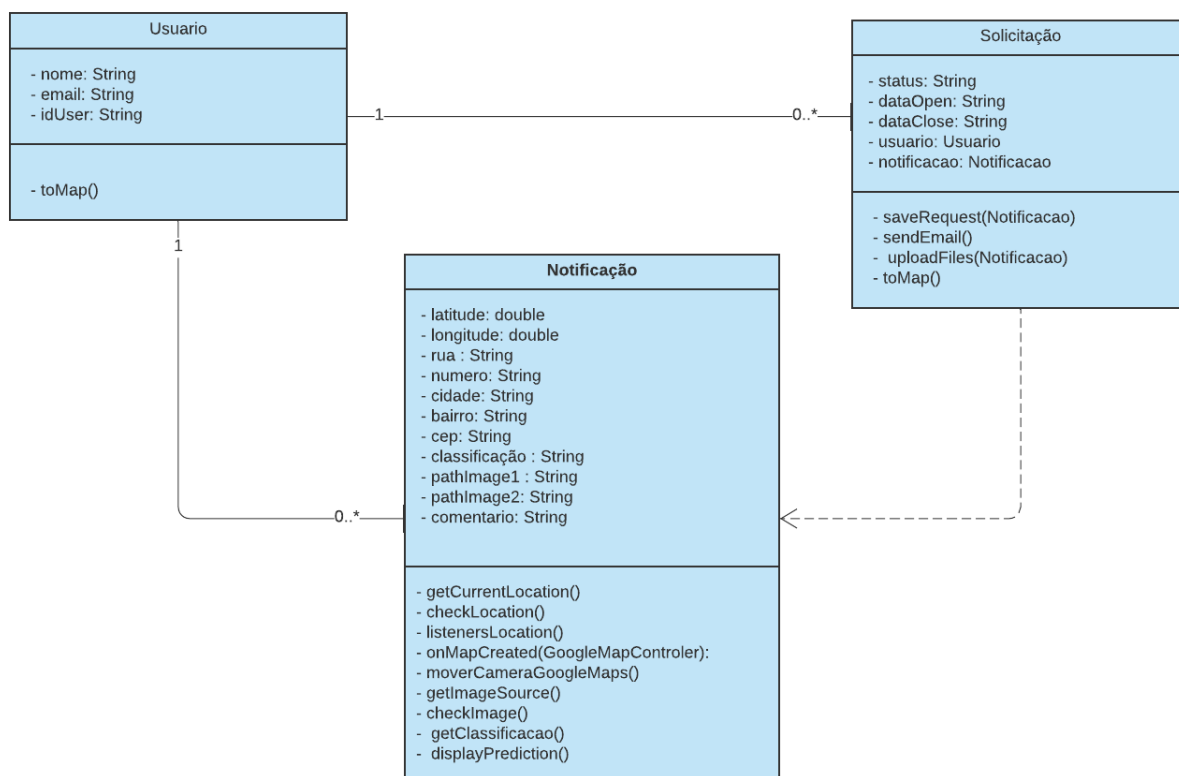


Fonte: Elaborado pelo autor (2021)

8.3 DIAGRAMA DE CLASSES

A Figura 6 representa como as classes se interagem entre si.

Figura 6 - Diagrama de classe



Fonte: Elaborado pelo autor (2021)

Um usuário, pode criar quantas notificações desejar. Toda solicitação criada está relacionada a um usuário.

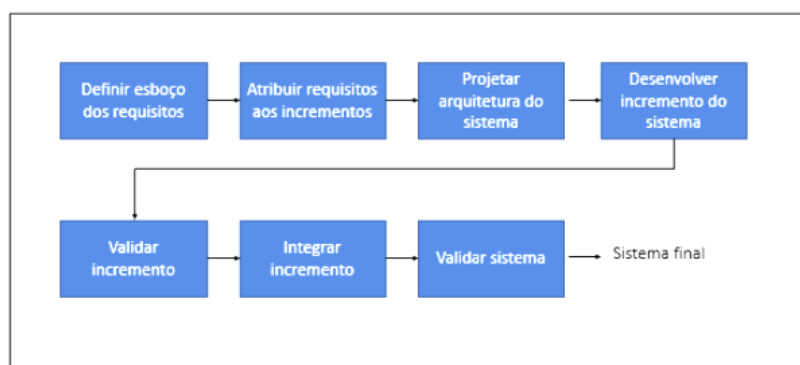
A dependência é um relacionamento no qual a mudança de uma classe pode afetar o comportamento ou estado de outra classe. Uma dependência é representada graficamente por uma linha tracejada, ligando duas classes, com uma seta apontando para a classe que a outra depende (PRESSMAN, 2006).

No caso acima, podemos dizer que **Solicitação** depende de **Notificação**. A operação `saveRequest()` e `uploadFiles()` de **Solicitação** recebe uma **Notificação** como argumento.

8.4 PROCEDIMENTO METODOLÓGICO

Como procedimento metodológico para o desenvolvimento do aplicativo, foi utilizado a o Modelo Incremental. Este modelo combina elementos do Modelo em Cascata aplicados de maneira iterativa, ou seja, de forma que o progresso aconteça através de sucessivos refinamentos, melhorados a cada iteração. (PRESSMAN, 2006).

Figura 7 - Modelo Incremental



Fonte: Treinaweb (2018)

8.5 ARQUITETURA DA APLICAÇÃO.

A parte do *backend* está integrado com a plataforma Firebase, a Figura 8 representa os serviços utilizados, disponibilizados pela plataforma.

A sincronização entre o aplicativo e *backend* é feita em tempo real, por qualquer modificação que altere o estado do *Database* ou *Storage*.

Figura 8 - Arquitetura da Aplicação



Fonte: Stackoverflow (2017) Adaptado pelo autor.

8.6 PROTOTIPAÇÃO

Após a definição dos requisitos funcionais e não funcionais, criação dos diagramas, foi realizado a criação de um protótipo do aplicativo.

De acordo com Noletto (2020, p. 1) “prototipagem de software é um processo iterativo de geração de modelos de software que faz parte da análise do ciclo de vida do desenvolvimento de sistemas”. Esta atividade busca desenvolver uma versão inicial do sistema com base no atendimento dos requisitos ainda pouco definidos.

Existem diversas ferramentas para prototipagem de telas, para este trabalho foi utilizado o site www.marvelapp.com, na qual o mesmo disponibiliza um plano grátis. A Figura 9 ilustra um protótipo da interface “Home” do aplicativo, a qual tem bastante semelhança com a interface original desenvolvida posteriormente.

Figura 9 - Protótipo da tela Home



Fonte: Elaborado pelo autor (2021)

Após a prototipação das principais telas do aplicativo, deu-se o início da codificação, sendo desenvolvido um incremento do sistema por vez, este incremento se refere a: Telas do aplicativo, funções, componentes, etc.

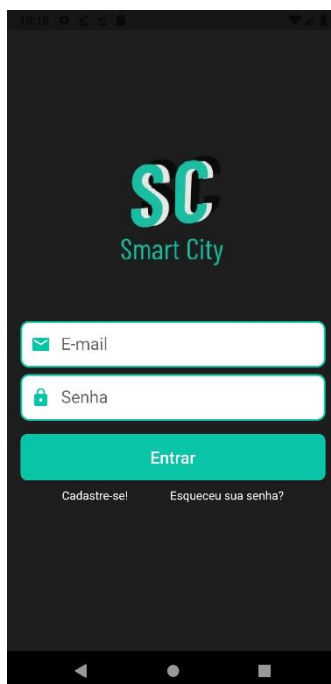
9 APRESENTAÇÃO DOS RESULTADOS: TELAS DO APLICATIVO

Uma das formas de exibir resultados, é apresentar as telas do sistema. Para uma melhor organização dos resultados, as telas principais são apresentadas nesta seção.

9.1 Login

Tela de *login*, onde a autenticação é realizada através de um e-mail e senha (Figura 10).

Figura 10 - Tela de Login

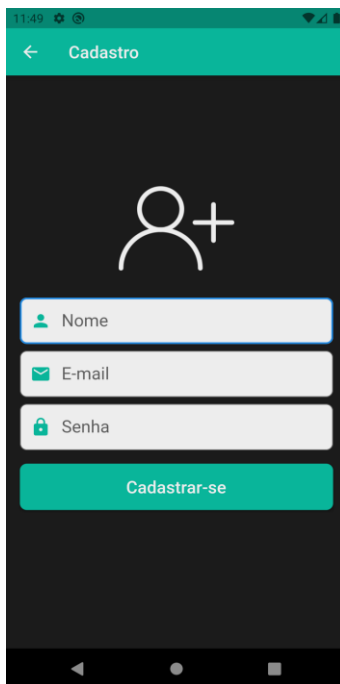


Fonte: Elaborado pelo autor (2021)

9.2 Cadastro

A Figura 11 demonstra a tela de cadastro, onde o usuário deve fornecer seu nome, e-mail e senha.

Figura 11 - Tela de cadastro

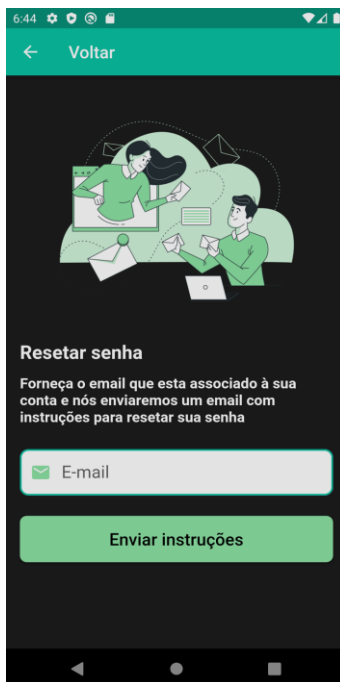
A imagem mostra a tela de cadastro de um aplicativo. No topo, há uma barra de navegação com o ícone de seta para trás e o texto "Cadastro". Abaixo, centralizado, há um ícone de perfil humano com um sinal de mais. Seguem-se três campos de entrada de texto: "Nome" (com ícone de pessoa), "E-mail" (com ícone de envelope) e "Senha" (com ícone de cadeado). Abaixo dos campos, há um botão verde com o texto "Cadastrar-se".

Fonte: Elaborado pelo autor (2021)

9.3 Resetar Senha

Permite que o usuário troque de senha, caso a esqueça, é enviado um e-mail para o usuário, caso exista uma conta registrada com o e-mail informado (Figura 12).

Figura 12 - Recuperação de senha

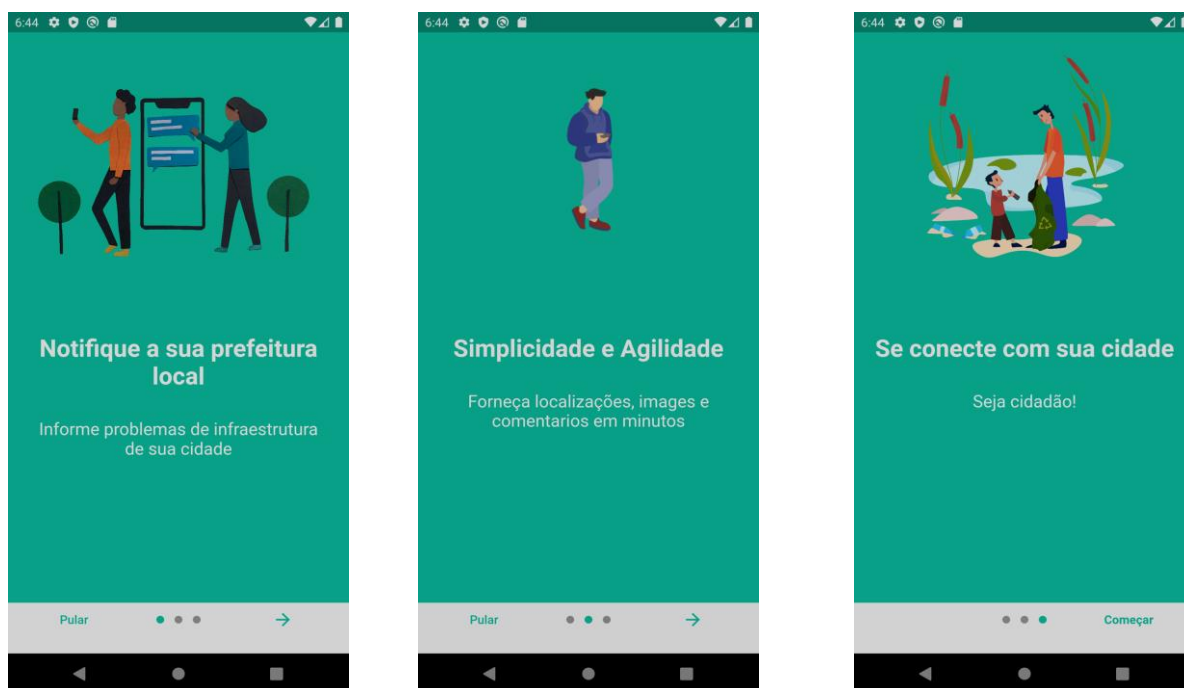
A imagem mostra a tela de recuperação de senha. No topo, há uma barra de navegação com o ícone de seta para trás e o texto "Voltar". Abaixo, há uma ilustração de uma mulher e um homem interagindo com dispositivos eletrônicos. Segue-se o título "Resetar senha" e o texto: "Forneça o email que esta associado à sua conta e nós enviaremos um email com instruções para resetar sua senha". Abaixo, há um campo de entrada de texto "E-mail" (com ícone de envelope) e um botão verde com o texto "Enviar instruções".

Fonte: Elaborado pelo autor (2021)

9.4 Introdução ao aplicativo

Caso seja a primeira vez que o usuário esteja acessando o aplicativo, é apresentado uma pequena introdução do aplicativo, reforçando os conceitos de *User Experience*, apresentado no referencial teórico.

Figura 13 – Telas de introdução

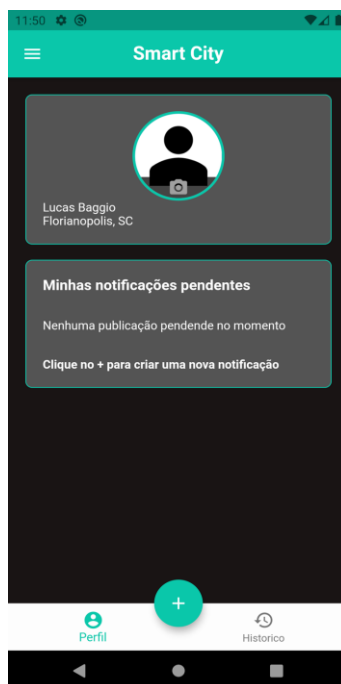


Fonte: Elaborado pelo autor (2021)

9.5 Home

Tela inicial do aplicativo após efetuar o login. O usuário possui um painel onde é exibido suas informações pessoais, notificações pendentes e um botão para criar notificações (Figura 14).

Figura 14 - Tela Home



Fonte: Elaborado pelo autor (2021)

9.6 Histórico de notificações criadas.

Na tela “Histórico” é apresentado o histórico de todas as notificações criadas pelo usuário, com todas as informações contidas em sua criação (Figura 15).

Figura 15 - Tela do Histórico



Fonte: Elaborado pelo autor (2021)

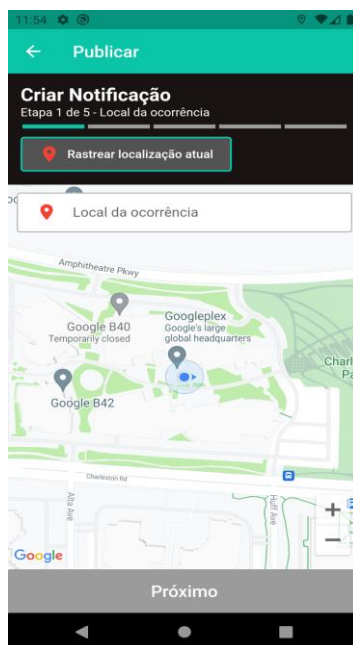
9.7 Criação de notificação

A seguir serão apresentadas as 4 telas responsáveis pela criação de uma notificação.

9.7.1 Tela 1: Localização da ocorrência

O usuário deve informar o endereço da ocorrência que pretende informar. Foram utilizados plugins para a utilização do Google Maps (Figura 16).

Figura 16 - Local da ocorrência

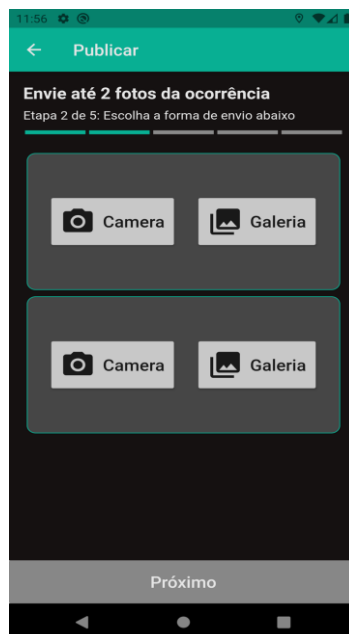


Fonte: Elaborado pelo autor (2021)

9.7.2 Tela 2: Fotos da ocorrência

Após informar a localização da ocorrência, o usuário deve enviar até 2 fotos da ocorrência (Figura 17).

Figura 17 - Fotos da ocorrência



Fonte: Elaborado pelo autor (2021)

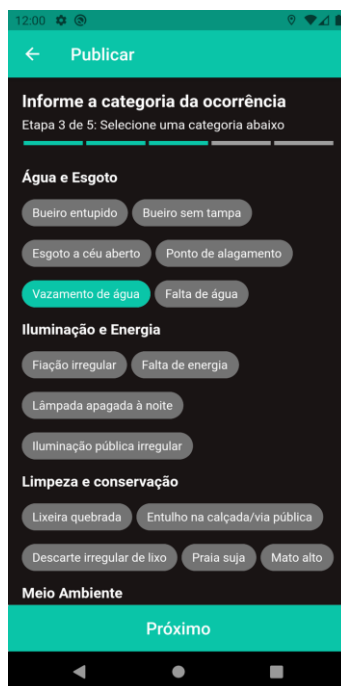
9.7.3 Tela 3: Classificação da ocorrência

Nesta etapa, o usuário deve classificar o tipo da ocorrência. São apresentadas categorias e o usuário deve escolher uma subcategoria que se encaixa corretamente com sua ocorrência.

As categorias servem como um filtro para o usuário se guiar, e as subcategorias indicam diretamente o tipo do problema.

A Figura 18 demonstra como que estão estruturadas as categorias e subcategorias no aplicativo.

Figura 18 - Classificação da ocorrência

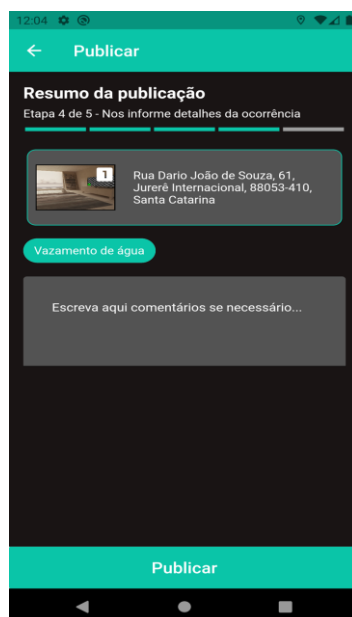


Fonte: Elaborado pelo autor (2021)

9.7.4 Tela 4: Resumo da ocorrência

Nesta tela é apresentado um resumo de todas as informações fornecidas anteriormente pelo usuário, para que ele possa validá-las e também um campo opcional onde ele pode colocar outras informações (Figura 19).

Figura 19 - Resumo da ocorrência



Fonte: Elaborado pelo autor (2021)

9.8 Disparo de e-mail automático

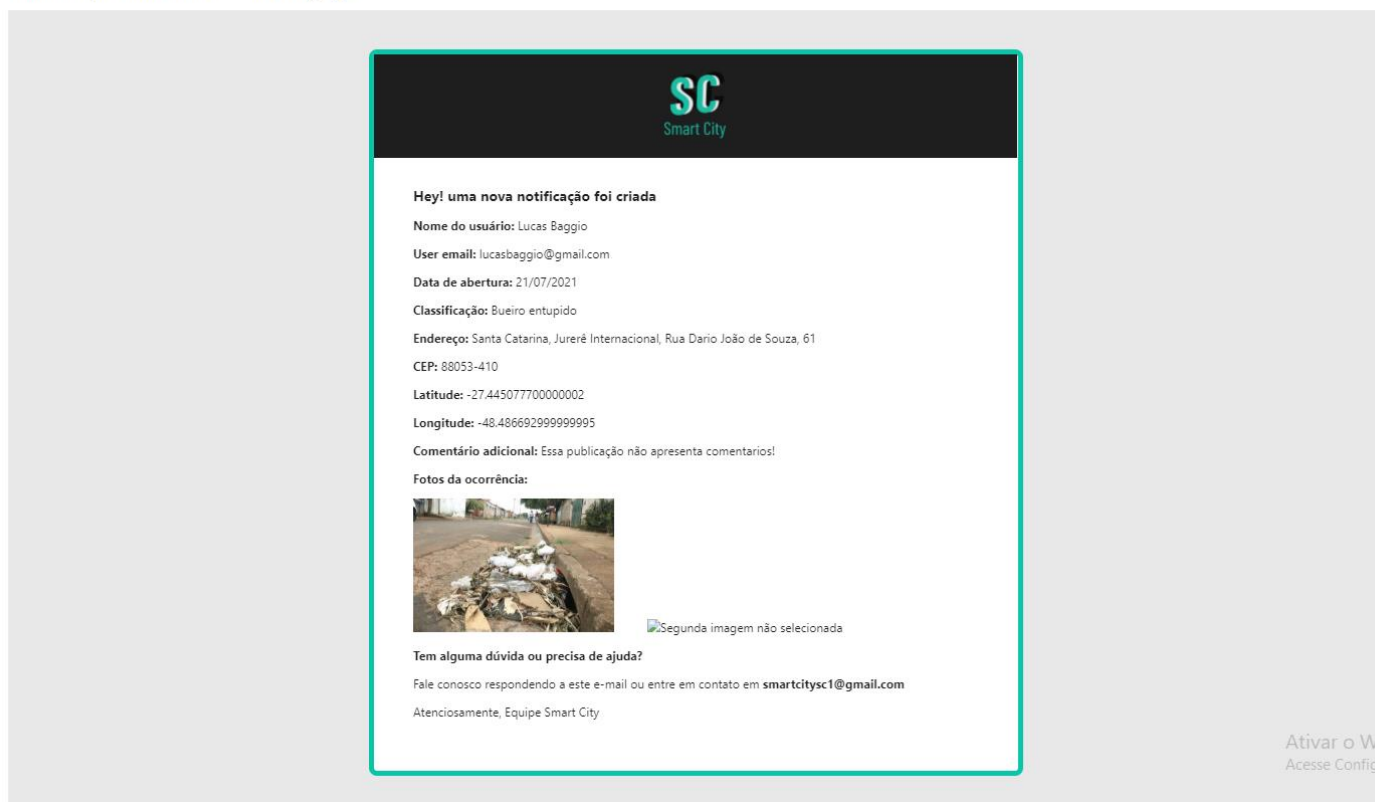
Até o momento, o aplicativo foi desenvolvido apenas para ser utilizado por parte do usuário. A seguir será demonstrado como a prefeitura receberá as notificações desses problemas.

Após a notificação de um problema, um e-mail é disparado diretamente para a prefeitura, com todas as informações coletadas e fornecidas. O *template* do e-mail foi configurado através de um documento HTML, permitindo assim a personalização do mesmo.

Exemplo de um e-mail criado a partir de uma nova notificação (Figura 20).

Figura 20 - Email Smart City

← Nova notificação criada - Número: 35 - Smart City - 😊



Fonte: Elaborado pelo autor (2021)

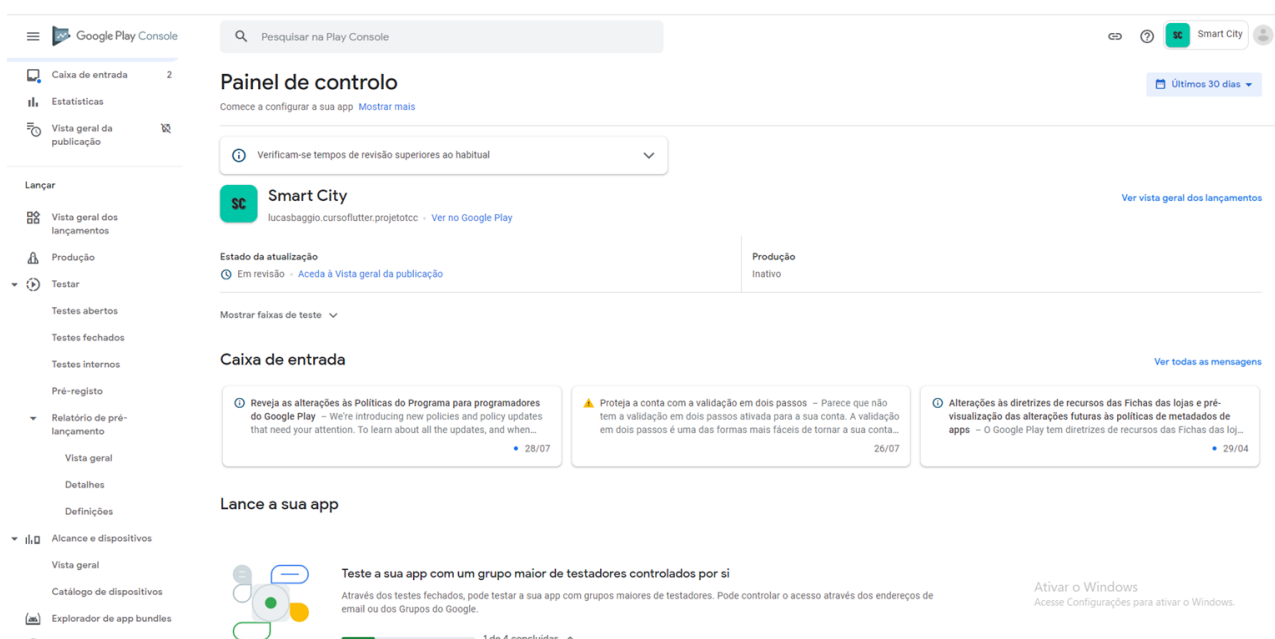
Para melhor visualização, acesse este link: <https://prnt.sc/1fsau1q>

9.9 Etapa de implantação

Finalizada a criação do protótipo do aplicativo, deu-se início ao processo de implementação do aplicativo no Google Play.

Para a publicação do aplicativo no Google Play, é necessário criar uma conta de desenvolvedor, na qual é cobrado uma taxa de \$25,00, esta taxa é cobrada apenas na criação da conta, a publicação de qualquer outro aplicativo posteriormente não contém custos. A Figura 21 demonstra o painel principal da plataforma.

Figura 21 - Painel de Controle Goole Play



Fonte: Google Play Console (2021).

Como ainda o aplicativo se trata de um protótipo, foi lançado uma versão Beta, para realização dos primeiros testes e verificações.

Utilizando a plataforma Google Play Console, é possível identificar o estado da publicação, erros encontrados, sugestões de melhora, em suma, o mesmo proporciona uma visão completa dos dados do aplicativo durante e após o lançamento.

Para exemplificação, durante o pré-lançamento do aplicativo, é realizado análises da versão, em relação a: estabilidade, desempenho, acessibilidade, segurança e fidedignidade (Figura 22).

Figura 22 - Análise da versão publicada

Vista geral do relatório de pré-lançamento Versão da app: 1.aab

Resumo

App bundle testado em 4 dispositivos diferentes [Saiba mais](#)

⚠ Foram encontrados avisos. Recomendamos que os corrija antes de lançar para produção.

| | | | |
|---|---|---|--|
| Estabilidade 4 problemas únicos | Desempenho 0 problemas únicos | Acessibilidade 2 problemas únicos | Segurança e fidedignidade 0 problemas únicos |
|---|---|---|--|

Estabilidade ⓘ [Ver detalhes](#) →

⚠ 4 problemas únicos (4 avisos)

[Mostrar resumo](#) ▾

Desempenho ⓘ [Ver detalhes](#) →

✅ Nenhum problema encontrado

Acessibilidade ⓘ [Ver detalhes](#) →

⚠ 2 problemas únicos (1 aviso, 1 problema menor)

[Mostrar resumo](#) ▾

Segurança e fidedignidade ⓘ [Ver detalhes](#) →

✅ Nenhum problema encontrado

Ativar o Windows
Acesse Configurações para ativar o Windows

Fonte: Google Play Console (2021).

Para realizar uma publicação na Play Store, é preciso que o lançamento do aplicativo esteja de acordo com as políticas e diretrizes da plataforma. Minha publicação foi revisada durante 7 dias e posteriormente aceita pela equipe do Google.

A versão Beta está disponível apenas para testes fechados, ou seja, apenas usuários convidados conseguem baixar o aplicativo (Figura 23).

Figura 23 - Testes fechados

The screenshot shows the Google Play Console interface for an app named 'Alpha'. The 'Testar' (Test) section is selected in the sidebar, and the 'Testadores' (Testers) sub-section is active. The main content area shows the 'Testadores' configuration page, which includes a progress bar at the top indicating '1 de 4 concluídas'. The page is divided into two main sections: 'SET UP YOUR TRACK' and 'CRIE E IMPLEMENTE UM LANÇAMENTO'. Under 'SET UP YOUR TRACK', there are options for 'Select countries' and 'Selecionar testadores'. Under 'CRIE E IMPLEMENTE UM LANÇAMENTO', there are options for 'Crie um novo lançamento' and 'Review and roll out release'. The 'Testadores' section is currently selected, showing options for 'Listas de correio' (Email lists) and 'Grupos Google' (Google Groups). The 'Listas de correio' option is selected, and there is a button to 'Criar lista de emails'. The interface also includes a sidebar with navigation options like 'Lançar', 'Testar', and 'Relatório de pré-lançamento', and a top navigation bar with a search bar and user profile.

Fonte: Google Play Console (2021).

Finalizado a criação e publicação do protótipo, foi agendada uma reunião dia 26/07/2021 com a prefeitura, para a apresentação da proposta.

Simpatizaram com a ideia proposta e deram sugestões para uma possível implementação, porém o aplicativo teria que ser adaptado de acordo com suas necessidades.

Tais adaptações se referem principalmente a redução/alteração do mapeamento de problemas (Tabela 1) providos pelo aplicativo, porque a prefeitura municipal tem um organograma próprio de serviços oferecidos para a solução de problemas urbanos, sendo assim, inicialmente o aplicativo deve dar suporte apenas aos serviços contidos neste organograma, para permitir uma implementação mais simples, sem a necessidade da participação de terceiros para a soluções destes problemas urbanos, que não estão no escopo de serviços oferecidos diretamente pela a prefeitura.

Foi sugerido uma possibilidade de estágio na prefeitura, durante o desenvolvimento e adaptação do aplicativo. A ideia central é que, após o desenvolvimento do aplicativo, o mesmo seja implementado como um serviço.

Em virtude de não atender as necessidades informadas pela prefeitura, não foi possível realizar a implantação do serviço/app dentro do prazo solicitado para a defesa deste TCC.

Porém, o projeto seguirá em andamento, não apenas como uma ideia que nasceu durante um trabalho de conclusão de curso, e sim uma ideia que poderá se materializar futuramente como um negócio.

10 CONCLUSÃO

Este capítulo visa apresentar as considerações finais com a experiência deste trabalho e sugerir possíveis melhorias para o futuro.

Este trabalho teve como objetivo propor uma solução informatizada para problemas urbanos que envolvem infraestrutura, proporcionando a comunicação entre a gestão pública x população, com o intuito que essa interação seja simples, ágil e satisfatória.

A metodologia escolhida para o desenvolvimento do aplicativo foi o Modelo Incremental. Esta metodologia se mostrou eficaz nas etapas aplicadas, contribuindo para a organização e agilidade das tarefas.

Com o protótipo do aplicativo finalizado, foi satisfatório perceber o interesse dos servidores públicos da prefeitura, na qual propuseram trabalhos futuros para a adaptação do aplicativo.

Por fim, o desenvolvimento deste trabalho contribuiu para uma maior aprendizagem no planejamento de desenvolvimento de software, aliando teoria a prática, seguindo uma metodologia adequada ao processo.

10.1 TRABALHOS FUTUROS

Abaixo são apresentados abaixo alguns pontos que poderão ser trabalhados no futuro:

- Adaptação do aplicativo de acordo com as solicitações da prefeitura
- Validação dos requisitos para identificar novas funcionalidades ou possíveis correções
- Realização de testes e correção de bugs.
- Disponibilização do aplicativo para a sistema operacional iOS, pois no momento apenas dá suporte ao Android.
- Coleta de feedbacks dos usuários, através da Google Play.
- Implementação do aplicativo como um serviço.
- Possibilidade de integração do aplicativo com redes sociais.
- Possibilidade de sistemas de recompensa / descontos para usuários que colaborarem com a criação de notificações.

REFERÊNCIAS

ALMEIDA, M. S. **Elaboração de Projeto, TCC, Dissertação e Tese: uma abordagem simples, prática e objetiva**. São Paulo: Atlas, 2011.

ANGELO CARVALHO. **Problemas Urbanos**. 2018. Disponível em: <https://querobolsa.com.br/enem/geografia/problemas-urbanos>. Acesso em: 21 dez. 2020.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: Guia do usuário**. 2. ed. Rio de Janeiro: Elsevier Editora Ltda, 2012. 745 p. Disponível em: <https://docero.com.br/doc/vee1vv>. Acesso em: 20 jan. 2021.

DEVMEDIA. **Introdução ao Padrão MVC**. 2013. Disponível em: <https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>. Acesso em: 02 jan. 2020.

FUNDAÇÃO INSTITUTO DE ADMINISTRAÇÃO. **Infraestrutura: O que é, Tipos, Importância e Problemas**. 2019. Disponível em: <https://fia.com.br/blog/infraestrutura/>. Acesso em: 12 dez. 2020.

G1. **Infraestrutura do Brasil tem desafios a enfrentar**. 2018. Disponível em: <https://g1.globo.com/especial-publicitario/em-movimento/noticia/infraestrutura-do-brasil-tem-desafios-a-enfrentar.ghtml>. Acesso em: 23 dez. 2020.

LUCIDCHART. **O que é um diagrama de classe UML?** 2020. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml#:~:text=Diagramas%20de%20classes%20est%C3%A3o%20entre,opera%C3%A7%C3%B5es%20e%20rela%C3%A7%C3%B5es%20entre%20objetos..> Acesso em: 07 jan. 2021.

LUCIDCHART. **O que é um diagrama UML?** 2020. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-uml#:~:text=A%20Linguagem%20de%20modelagem%20unificada,tanto%20estruturalmente%20quanto%20para%20comportamentos..> Acesso em: 07 jan. 2021.

PRESSMAN, Roger S. **Engenharia de Software: Uma abordagem profissional**. 7. ed. New York: AMGH, 2011. 780 p. Disponível em: https://www.academia.edu/40370740/Engenharia_de_Software_Uma_Abordagem_Profissional. Acesso em: 22 dez. 2020.

ROCKETSEAT. **Git & Github: O que é? Por que? Como iniciar?** 2019. Disponível em: <https://blog.rocketseat.com.br/iniciando-com-git-github/>. Acesso em: 13 jan. 2021.

SEBRAE. **CIDADES INTELIGENTES: O QUE SÃO?** 2017. Disponível em: <https://inovacaosebraeminas.com.br/cidades-inteligentes-o-que-sao/>. Acesso em: 22 dez. 2020.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011. 529 p. Disponível em: <http://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/engenhariaSoftwareSommerville.pdf>. Acesso em: 25 dez. 2020.

VEJA. 5,1 bilhões de pessoas têm celular no planeta, sendo 204 milhões no Brasil. 2019. Disponível em: <https://veja.abril.com.br/economia/51-bilhao-de-pessoas-tem-celular-no-planeta-sendo-204-milhoes-no-brasil/>. Acesso em: 22 dez. 2020.