



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

**Tô de Olho para iOS: Um Software Para
Mapeamento Colaborativo de Desordens Sociais por
Meio de Sistemas Georreferenciados**

Paulo da Cunha Passos

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientadora
Prof.a Dr.a Maristela Terto de Holanda

Brasília
2019

Dedicatória

A minha amada esposa, Cybelle Oliveira Soares da Silva, pelo companheirismo, cumplicidade e compreensão que teve comigo nos muitos momentos que estive ausente do lar ou isolado no escritório para me dedicar aos estudos. Muitos foram os momentos que me ajudou, me deu força e incentivo. Por várias e várias vezes, teve a paciência de ler e reler meus trabalhos e emitiu opiniões que me ajudaram muito a melhorar os meus textos. Durante estes longos anos na Universidade de Brasília foi companheira em vários sentidos, já assistiu aulas comigo, já foi às apresentações minhas, já estudou comigo, me motivou em semestres difíceis e contribuiu de diversas maneiras para a conclusão deste curso.

Agradecimentos

Ao Nosso Senhor Jesus Cristo e sua mãe, Maria Imaculada, por terem me guiado e inspirado nessa longa jornada em busca de conhecimento e aprendizado.

Ao professor Jorge Luiz Pereira, amigo e mentor, que foi fonte de motivação e ensinamentos. E foi, acima de tudo, um guia durante uma dura jornada que culminou na mudança radical da minha vida, por meio da dedicação incansável aos estudos.

A professora Maristela Terto de Holanda, a quem muito admiro, minha orientadora e pessoa querida, por compartilhar comigo seu conhecimentos e suas vivências, pela paciência com que ensina e por ter me cativado a sair da inércia do dia-a-dia e embarcar em projetos acadêmicos que foram tão profícuos e aprazíveis nesses últimos anos da graduação.

Aos meus irmãos e amigos da Turma José Bonifácio de Andrada e Silva, pela camaradagem e apoio ao longo desses anos.

A Universidade de Brasília, seu corpo docente, servidores e demais pessoas, pelo privilégio do convívio, pelos conhecimentos que me passaram durante os longos anos que aqui estive e pela oportunidade de realizar o sonho de concluir a graduação em uma universidade federal.

A Fundação de Apoio a Pesquisa do Distrito Federal que financiou o projeto que deu origem a esta monografia.

Aos meus pais, Paulo e Lucia, meus primeiros incentivadores, pelo amor e apoio incondicional.

A minha irmã, Andreia, pelo apoio e pela força que me deu nos primeiros anos na Universidade de Brasília. Anos estes que foram os mais difíceis de minha vida acadêmica.

A Cleide, Paulo Célio e Rodrigo Oliveira, que como parte da família, sempre me ajudaram e me deram palavras de incentivo para chegar ao final desse curso.

E a todos que direta ou indiretamente fizeram parte de minha formação, deixo aqui meus agradecimentos.

Resumo

Este documento apresenta o Tô de Olho no DF, um sistema de mapeamento de desordens sociais, que inclui um aplicativo para iOS. Tal abordagem tem como inspiração a teoria das janelas quebradas e os processos de mapeamentos criminais, que demonstram que há uma relação de causalidade entre desordens sociais e eventos criminais. O objetivo do trabalho é propor uma arquitetura que utilize Sistema de Informação Geográfico com Participação Popular para realizar o mapeamento de desordens sociais e fornecer os dados coletados aos gestores públicos, com vistas a subsidiar a tomada de decisão. A arquitetura proposta é do estilo REST, utiliza o SGBD PostgreSQL com extensão PostGIS, possui API desenvolvida usando Node.js e seu aplicativo para iOS é desenvolvido em *Swift*. O aplicativo foi disponibilizado na Apple store. A validação dos usuários apresentou um bom resultado, com 100% dos avaliadores com interesse em utilizar o aplicativo.

Palavras-chave: Mapeamento de desordens, aplicativo, sistema georeferenciado

Abstract

This document presents T^o de Olho no DF, a social disorder mapping system which includes an iOS application. Such an approach is inspired by the theory of broken windows and the processes of criminal mappings, which demonstrate that there is a causal relationship between social disorder and criminal events. This document proposes an architecture that uses the Public Participation Geographic Information System to map social disorders. The collected data can support decision making for government authorities. The proposed architecture is REST style, uses the PostgreSQL with PostGIS extension, has API developed using Node.js and an iOS application developed in Swift. The details of the implementation, the technologies used, and the validation process are presented throughout the present study.

Keywords: Disorder mapping, application, georeferenced system

Sumário

1	Introdução	1
1.1	Objetivos	3
1.1.1	Objetivos Específicos	3
1.2	Estrutura do Trabalho	3
2	Referencial Teórico	5
2.1	Análise Criminal	5
2.1.1	Definição	5
2.1.2	Histórico da Análise Criminal	6
2.1.3	Teorias Criminais de Suporte à Análise Criminal	8
2.1.4	Tipos de Análise Criminal	11
2.1.5	Processo de Análise Criminal	13
2.2	Sistema de Informação Geográfico	14
2.2.1	Conceitos Básicos	15
2.2.2	Sistema de Gerenciamento de Bancos de Dados	16
2.2.3	Sistema de Informação Geográfico com Participação Popular	19
2.3	REST e RESTful	20
2.4	Sistema Operacional iOS	21
2.4.1	Arquitetura	21
2.4.2	A Linguagem de Programação <i>Swift</i>	25
3	O Estado da Arte	30
3.1	Ferramentas Analisadas	30
3.1.1	<i>WikiCrimes</i>	30
3.1.2	OndeFuiRoubado	31
3.1.3	UnB Alerta	31
3.1.4	QAP mobile	32
3.1.5	B.O. Coletivo	32
3.1.6	Eu fui Roubado	32

3.1.7 IMA – Denuncie	33
3.1.8 <i>Reporting Crimes</i>	33
3.2 Análise das Aplicações	33
3.3 Discussão Sobre as Ferramentas	37
4 Arquitetura Proposta	39
4.1 Cenário	39
4.2 Arquitetura do Sistema	41
4.2.1 Modelo em Três Camadas	42
4.3 Interface Web do Tô de Olho no DF	48
4.4 Ambiente Utilizado para Disponibilizar os Serviços do Tô de Olho no DF . .	50
4.4.1 Heroku	51
4.4.2 Amazon Web Services	52
5 O Aplicativo ToDeOlho para iOS	53
5.1 Engenharia de Software	53
5.2 Pré-planejamento	54
5.2.1 Análise de Domínio	54
5.2.2 Análise de Tarefas	55
5.2.3 Requisitos	55
5.3 Desenvolvimento	56
5.3.1 Tecnologias Utilizadas para o Desenvolvimento do Aplicativo	56
5.4 Resultados	57
5.4.1 Tela Inicial	58
5.4.2 <i>Login</i>	59
5.4.3 Cadastro de Usuário	60
5.4.4 Cadastro de Desordem	61
5.4.5 Mapa	63
5.4.6 Lista de Desordens	63
6 Prova de Conceito	65
6.1 Teste Funcional	65
6.2 Teste de Usabilidade	66
6.2.1 Testes Realizados pela Equipe da <i>App Store Connect</i>	67
6.2.2 Testes Realizados pelos Usuários Finais	69
7 Conclusão	84
Referências	86

Apêndice	93
A Formulário de Usabilidade do Aplicativo Tô de Olho no DF	94

Lista de Figuras

2.1	Definição de Análise Criminal (Adaptado)	7
2.2	Triângulo do Crime de Felson (Adaptado).	10
2.3	Triângulo do Evento Criminal(Adaptado).	10
2.4	Processo de Análise Criminal segundo Santos (Adaptado).	15
2.5	Arquitetura de um SIG (Adaptado).	17
2.6	Hierarquia de classes de geometrias do OGC (Adaptado).	18
2.7	Tipos de dados espaciais do PostGIS (Adaptado).	19
2.8	Participação no Mercado dos Sistemas Operacionais Móveis.	22
2.9	Interação entre o <i>hardware</i> , o iOS e aplicativos.	23
2.10	Arquitetura do iOS	23
4.1	Funcionamento do Sistema Tô de Olho.	41
4.2	Arquitetura do Sistema Tô de Olho.	42
4.3	Front-End e Back-End do Sistema Tô de Olho.	43
4.4	Modelo OMT-G do Tô de Olho.	47
4.5	Página inicial do sistema Tô de Olho.	49
4.6	Detalhes da denúncia.	49
4.7	Apresentação das denúncias em forma de mapa de calor.	50
4.8	Apresentação do número de denúncias por órgão responsáveis	50
4.9	Apresentação do número de denúncias por mês do ano.	51
4.10	Lista de denúncias cadastradas no sistema.	52
5.1	Análise de tarefas no mapeamento de desordens sociais feito pela SGI.	55
5.2	Visão geral das telas do aplicativo Tô de Olho no DF	58
5.3	Telas iniciais.	59
5.4	Telas de <i>login</i>	60
5.5	Telas do cadastro.	61
5.6	Registro de denúncia.	62
5.7	Telas de exibição do mapa.	63

5.8	Telas para listar desordens.	64
6.1	Mensagem sobre <i>bug</i> enviada pela equipe da <i>App Store Connect</i>	67
6.2	Mensagem sobre melhoria a ser realizada em relação a solicitação de acesso a localização do usuário.	67
6.3	Mensagem sobre melhoria a ser realizada em relação a solicitação de data de nascimento do usuário para cadastro no sistema.	67
6.4	Comparação entre telas.	68
6.5	Comparação entre telas.	68
6.6	Comparação entre telas.	69
6.7	Quantidade de <i>downloads</i> do aplicativo Tô de Olho no DF realizados na <i>Apple Store</i>	70
6.8	Dados do <i>App Analytics</i> sobre o aplicativo Tô de Olho no DF	70
6.9	Dados do <i>App Analytics</i> sobre o aplicativo Tô de Olho no DF	71
6.10	Grau de Escolaridade.	71
6.11	Familiaridade com <i>Smartphones</i>	72
6.12	Frequência com que usa <i>Smartphones</i>	72
6.13	Histórico de uso de aplicativos colaborativos.	73
6.14	Histórico de uso de aplicativos de mapeamento colaborativo.	73
6.15	Histórico de uso de aplicativos semelhantes ao Tô de Olho no DF.	74
6.16	Nível de participação em redes sociais ou aplicativos de colaboração popular.	74
6.17	Classificação da ordem/conservação no bairro onde reside.	75
6.18	Pré-disposição para relatar desordens em seu bairro por meio de aplicativo de colaboração.	75
6.19	Modelo de iPhone usado.	76
6.20	Cadastro no Aplicativo.	77
6.21	Cores e botões do Aplicativo.	77
6.22	Visualização de botões e informações.	78
6.23	Entendimento da nomenclatura do aplicativo.	78
6.24	Visualização do mapa com marcadores.	79
6.25	Compreensão do conteúdo do mapa.	79
6.26	Listar desordens.	80
6.27	Apresentação de detalhes de uma desordem.	80
6.28	Realização de <i>login</i>	81
6.29	Cadastro de denúncia.	81
6.30	Movimentação do mapa e carregamento de denúncias.	82
6.31	Opinião sobre potencial do Tô de Olho no DF para auxiliar a administração pública.	82

6.32 Primeira impressão sobre Tô de Olho no DF.	83
6.33 Considerações para melhoria do Tô de Olho no DF.	83

Lista de Tabelas

2.1 Principais métodos internos de solicitações HTTP e comandos semelhantes em SQL (Adptado)	21
3.1 Plataforma das Aplicações (adaptado).	34
3.2 Principais Funcionalidades das Aplicações (adaptado).	35
3.3 Controle de Credibilidade de Registro (adaptado).	36
3.4 Apresentação dos Crimes no Mapa (adaptado).	36
3.5 Outras Funcionalidades (adaptado).	36
4.1 Tabela de Relacionamento do Modelo de Dados.	48
5.1 Resultado da análise de domínio.	54
6.1 Descrição do <i>smartphone</i> utilizado para o teste funcional	65

Lista de Abreviaturas e Siglas

ABI *Application Binary Interface.*

Amazon S3 *Amazon Simple Storage Service.*

API *Application Programming Interface.*

AWS *Amazon Web Services.*

CIC Departamento de Ciência da Computação.

Consegs Conselhos Comunitários de Segurança Pública.

DF Distrito Federal.

FAP-DF Fundação de Apoio a Pesquisa do Distrito Federal.

GCD *Grand Central Dispatch.*

GiST *Generalized Search Tree.*

GPS Sistema de Posicionamento Global.

GSSAPI *Generic Security Services Application Program Interface.*

HTML *Hypertext Markup Language.*

HTTP *Hypertext Transfer Protocol.*

IaaS *Infrastructure as a Service.*

IACA *International Association of Crime Analysts.*

IALEIA *International Association of Law Enforcement Intelligence Analysts.*

IDE *Integrated Development Environment.*

IMA-AL Instituto do Meio Ambiente de Alagoas.

JSON *JavaScript Object Notation.*

LDAP *Lightweight Directory Access Protocol.*

MR Modelo Relacional.

MVC *Model-View-Controller.*

NCGIA *National Center for Geographic Information and Analysis.*

ODbL *Open Data Commons Open Database License.*

OGC *Open Geospatial Consortium.*

OSM OpenStreetMap.

OSMF *Foundation OpenStreetMap.*

PaaS *Platform as a Service.*

POP *Protocol Oriented Programming.*

PPGIS *Public Participation Geographic Information System.*

REST *Representational State Transfer.*

SaaS *Packaged Software as a Service.*

SBDE Sistemas de Bancos de Dados Espaciais.

SCRAM *Salted Challenge Response Authentication Mechanism.*

SENASP Secretaria Nacional de Segurança Pública.

SGBD Sistemas de Gerenciamento de Bancos de Dados.

SGBD-OR Sistema de Gerenciamento de Banco de Dados Objeto-Relacional.

SGI Subsecretaria de Gestão da Informação.

SIG Sistema de Informação Geográfica.

SIGEO Sistema Integrado de Geoestatística.

SQL *Structured Query Language.*

SSP Secretaria de Estado da Segurança Pública e da Paz Social.

SSPI *Security Support Provider Interface.*

TI Tecnologia da Informação.

UnB Universidade de Brasília.

Unifor Universidade de Fortaleza.

URL *Universal Resource Locator.*

WKB *Well-Known Binary.*

WKT *Well-Known Text.*

WWDC *Apple Worldwide Developers Conference.*

Capítulo 1

Introdução

No ano de 1982, o criminologista George Kelling e o cientista político James Q. Wilson escreveram o artigo denominado "*Broken windows: the police and neighborhood safety*" (Janelas Quebradas: A Polícia e a Segurança do Bairro, tradução livre). O artigo foi publicado na revista *The Atlantic Monthly* e estabelecia uma relação de causalidade entre desordem e criminalidade e entre repressão a pequenos delitos e à criminalidade violenta. No estudo, os autores se valeram das imagens de janelas quebradas para explicar como aos poucos a desordem e a criminalidade poderiam se infiltrar em uma comunidade e causar deterioração. Eles relatavam que se a janela de uma fábrica ou prédio é quebrada e não é consertada de pronto, as pessoas que por ali transitam tendem a concluir que ninguém se importa com o local e que não há autoridade zelando pela manutenção da ordem. Dessa forma, sentem-se à vontade para quebrar as demais janelas e inicia-se a deterioração do local, de maneira que pequenas desordens passam a levar a grandes desordens [1].

À vista disso, se as desordens pequenas forem reprimidas, prevenir-se-á desordens maiores e até mesmo crimes. Portanto, o mapeamento de locais de desordem e suas frequências de ocorrência possibilita ao gestor público conhecer melhor o fenômeno da desordem e sua distribuição no espaço e no tempo, bem como pode tornar mais efetivo o planejamento público e a tomada de decisão. No entanto, reunir informações de desordem em diversos pontos das cidades e do estado é um desafio ao gestor público, pois podem ser muitos os locais de desordem, o que implica em um grande esforço do Estado para coleta dessas informações, se o fizer de forma centralizada e por meio de servidores públicos ou de pessoas contratadas.

Nesse contexto, o uso da inteligência coletiva surge como uma boa opção para a administração pública. A população é estimulada a participar do processo administrativo da sua cidade e se aproximar do gestor público. E isso torna-se possível, em grande escala, por causa do desenvolvimento progressivo do Sistema de Posicionamento Global (GPS), dos Sistema de Informação Geográfica (SIG) e das tecnologias contidas na Web

2.0, acompanhadas de sua acessibilidade ao cidadão por meio dos *smartphones*[2][3].

A popularização dos *smartphones* e a facilidade de conexão à internet, aumentou a capacidade do cidadão de criar conteúdo, interagir com sistemas de informática e com outras pessoas distantes fisicamente, praticamente, em tempo real. Possibilitando a qualquer cidadão, que tenha acesso a um dispositivo móvel, contribuir com aplicações SIG, uma vez que os *smartphones* permitem o georreferenciamento, associando alguma informação a uma localização no espaço, geralmente por meio de latitude e longitude [4] [2].

Visando fomentar a pesquisa na área de Segurança Pública no Distrito Federal e RIDE-DF, a FAP-DF, abriu o edital 07/2017, disponibilizando recursos para algumas linhas de pesquisa. Sendo uma delas o desenvolvimento de tecnologias de aproximação entre a população e os órgãos de segurança pública do Distrito Federal. Ressalta-se que nesta linha de pesquisa, o Departamento de Ciência da Computação (CIC) da Universidade de Brasília (UnB) teve o projeto “Arquitetura para Gerenciamento de Dados Colaborativos para Secretaria de Estado da Segurança Pública e da Paz Social (SSP) do Distrito Federal (DF) baseada em Sistema de Informação Geográfico com Participação Popular” selecionado no referido edital. E o presente trabalho é resultado de parte da pesquisa realizada neste projeto.

No ano de 2015, a SSP-DF criou o programa “Viva Brasília - Nosso Pacto pela Vida” voltado à prevenção da violência e combate da criminalidade, por meio de metodologias de resolução de problemas de criminalidade. Sendo, uma das frentes do programa, o mapeamento de desordens urbanas e de crimes. Para realizar esses mapeamentos, a Subsecretaria de Gestão da Informação (SGI) vale-se dos Conselhos Comunitários de Segurança Pública (Consegs) ou de articuladores territoriais contratados pelo Estado [5]. No entanto, apesar de boa parte do processo de mapeamento de desordens e crimes, feito pela SGI, ser informatizado, a parte coleta dos dados de desordem é realizada de forma manual.

Portanto, no presente projeto pretende-se reunir os conceitos de computação móvel, banco de dados geográficos e *Public Participation Geographic Information System* (PPGIS) [6] para propor o desenvolvimento de uma arquitetura, que inclua um aplicativo para iOS; que permita realizar o mapeamento de desordens sociais com participação da população; e que forneça os dados coletados ao gestor público para subsidiar a tomada de decisão. Obtendo-se assim, um processo de mapeamento de desordens todo informatizado, com maior alcance e que conte com a colaboração de qualquer cidadão que possua um *smartphone* e que esteja disposto a contribuir.

1.1 Objetivos

O presente trabalho propõe o desenvolvimento de uma arquitetura e de um aplicativo iOS que utilize Sistema de Informação Geográfico com Participação Popular para realizar o mapeamento de desordens sociais e que forneça dados coletados ao gestor público para subsidiar a tomada de decisão. A arquitetura é voltada para a coleta de dados de desordem social, que são fornecidos de maneira voluntária pelos usuários, por meio de *smartphones* e busca apresentar esses dados em tempo real aos gestores públicos, para que tenham a possibilidade de realizar análise do fenômeno da desordem ou de eventuais crimes resultantes destas desordens sociais.

1.1.1 Objetivos Específicos

Para alcançar o objetivo geral, este projeto tem os seguintes objetivos específicos:

- Criar um banco de dados central para os dados coletados;
- Implementar uma API RESTful que receba requisições de clientes, por meio do protocolo HTTP, e retorne os dados requisitados que estejam gravados no banco de dados;
- Especificar e implementar um aplicativo iOS que permita aos usuários cadastrarem informações de desordem social;
- Validar o aplicativo no contexto do Distrito Federal.

1.2 Estrutura do Trabalho

Este trabalho de conclusão de curso é composto pelos seguintes capítulos:

- Capítulo 2 - Referencial Teórico: apresentação dos conceitos fundamentais para o compreensão do estudo. Neste capítulo são abordados os seguintes assuntos: Análise Criminal, Sistema de Informação Geográfica, Sistemas de Bancos de Dados Espaciais, REST e RESTful e sistema operacional iOS.
- Capítulo 3 - O Estado da Arte: apresentação de estudo sobre aplicativos que realizam mapeamentos de crimes e sobre suas funcionalidades.
- Capítulo 4 - Arquitetura do Sistema Tô de Olho: apresentação da arquitetura proposta e as tecnologias necessárias para a implementação dessa arquitetura;
- Capítulo 5 - O aplicativo Tô de Olho: apresentação do aplicativo desenvolvido para a plataforma iOS que viabiliza a coleta e o mapeamento de dados de desordens. E

descrição das ferramentas utilizadas para o desenvolvimento do Tô de Olho, bem como as etapas do processo de desenvolvimento.

- Capítulo 6 - Prova de Conceito: descrição dos testes realizados na aplicação e de seus resultados, apontando possíveis problemas e pontos a melhorar.
- Capítulo 7 - Conclusão: apresentação das conclusões e dos trabalhos futuros.

Capítulo 2

Referencial Teórico

Neste capítulo são apresentados os principais conceitos utilizados no desenvolvimento deste trabalho. Esses conceitos estão subdivididos nas seguintes seções: Seção 2.1 apresenta os principais conceitos relacionados a Análise Criminal; Seção 2.2 aborda os conceitos relacionados a Sistema de Informação Geográfica (SIG), Sistemas de Gerenciamento de Bancos de Dados (SGBD), Sistemas de Bancos de Dados Espaciais (SBDE) e a *Public Participation Geographic Information System* (PPGIS); Seção 2.3 discorre brevemente sobre os conceitos de REST e RESTful; e Seção 2.4 trata do sistema Operacional iOS.

2.1 Análise Criminal

A análise criminal é uma ferramenta que dá suporte a tomada de decisões, em relação as estratégias de prevenção e repressão ao crime e as desordens sociais. Ela apoia as modernas estratégias de policiamento: orientadas a incidentes, orientadas para a comunidade, orientadas para a solução de problemas ou híbridas.

2.1.1 Definição

Existem diferentes definições de análise criminal, variando de país para país, de polícia para polícia ou de especialista para especialista, no entanto; Ranchel Boba [7] relata que apesar das diferenças, os especialistas concordam que a análise criminal serve para dar suporte à atividade policial, utilizando-se de métodos sistemáticos de coleta e análise de dados e produzindo informação e conhecimento para diferentes tipos de usuários. Nas palavras de Santos [7], a análise criminal é:

“O estudo sistemático de problemas de crime e desordem, bem como outros relacionados à segurança pública, incluindo fatores sociodemográficos, espaciais e temporais, para apoiar os órgãos de segurança pública na prevenção e redução

de crime e desordem, na prisão de criminosos e na avaliação da eficácia de medidas e ações de segurança pública.”

Entretanto, no ano de 2014, a *International Association of Crime Analysts* (IACA) publicou o artigo *Definition and Types of Crime Analysis* com a finalidade de lançar luz sobre as diferentes definições de análise criminal, nivelando-as e realizando a padronização de muitos termos utilizados na área [8]. Nesta esteira definiu análise criminal da seguinte forma:

“A análise criminal é uma profissão e um processo em que um conjunto de técnicas quantitativas e qualitativas são usadas para analisar dados valiosos para as agências policiais e suas comunidades. Inclui a análise de crimes e criminosos, vítimas de crimes, desordem, problemas de qualidade de vida, questões de trânsito e operações policiais, e seus resultados apoiam a investigação e acusação criminal, as atividades de patrulha, as estratégias de prevenção e redução do crime, a solução de problemas e a avaliação dos esforços policiais.”

Segunda a IACA, a análise criminal inclui todos os tipos de análises realizadas por uma agência policial, com exceção das análises de evidências de local de crime e de algumas análises relativas a rotinas administrativas da agência policial [8].

O foco da análise criminal é o estudo do crime e dos problemas de desordem, bem como todas as informações relacionadas à natureza de sua incidência, dos que a praticam e das vítimas ou alvos desses problemas [9]. Dessa forma, muitas são as características dos crimes e das desordens que são relevantes para a análise criminal, no entanto, existem três tipos de informações sobre esses fenômenos que se destacam: as informações sociodemográficas, que compreende as características das populações, tais como, sexo, idade, nível de escolaridade, altura, dentre outros; as informações espaciais, que se referem ao local de incidência do problema e suas características; e as informações temporais, que se referem ao momento na linha do tempo em que ocorreu o crime ou da desordem [7].

Dessa forma, verifica-se que por meio do estudo do crime e da desordem a análise criminal contribui com os órgãos de segurança para que realizem a prisão de criminosos, o combate e a prevenção de crimes e que possam realizar a avaliação das medidas e ações adotadas (Figura 2.1).

2.1.2 Histórico da Análise Criminal

A realização de análise de dados em agências policiais tem sua base na atividade de inteligência, que remonta às antigas estratégias chinesas e aos tempos bíblicos. Desde a antiguidade, as estratégias de guerra e a inteligência sempre dependeram da coleta e do tratamento de dados para se realizar a tomada de decisão [10].

Segundo Gottlieb [11] já no período feudal, se identificava, dentre as atividades análogas às exercidas pela polícia moderna, atividades que visavam identificar pessoas e seus hábitos,



Figura 2.1: Definição de Análise Criminal (Adaptado) (Fonte: [7]).

de forma que quando um crime fosse cometido já se tinha um ideia de quem poderia ser o criminoso.

No entanto, apesar da ideia de análise criminal ter uma longa história e ser antiga, a história da análise criminal como disciplina começou com a primeira força policial moderna, que foi criada em Londres, no século XIX. No ano de 1842, a força policial de Londres criou um departamento responsável por identificar padrões de crimes para ajudar a resolver crimes. O escritório desse departamento, já em 1844, coletava, agrupava e analisava informações policiais [7].

Nos Estados Unidos, as primeiras aplicações formais de análise criminal se deram no início dos anos mil e novecentos [7]. Nesse período, o Chefe do Departamento de Polícia de Berkeley, August Vollmer, além de implantar no departamento o uso de veículos para patrulhamento, o rádio como comunicação e o uso de impressão digital, estimulou o uso de mapa de alfinetes, a análise dos registros policiais e o patrulhamento policial baseado no volume de crimes da área. Vollmer introduziu a técnica inglesa de classificação sistemática de criminosos conhecidos e seus respectivos *modus operandi* e desenvolveu a técnica de exame das chamadas de emergência (*callsfor-service*) com a finalidade de analisar o serviço policial (rondas policiais) [12].

O primeiro que usou o termo "análise criminal" foi O. W. Wilson, que trabalhou com Vollmer e criou um avançado programa de treinamento de oficiais da polícia. Para Wilson a análise criminal era o processo de identificação de tendências e padrões de criminalidade, por meio do tratamento estatístico da informação e do exame de relatórios de investigação [9].

Nos anos 60, a análise criminal passou a se consolidar nos Estados Unidos e grandes

departamentos de polícia começaram a implementar unidades de análise criminal. Por volta de 1967, o Departamento de Justiça da Califórnia, seguido pelo Departamento Estadual de Polícia de New Jersey, passaram a desenvolver e usar técnicas analíticas. E logo, diversos órgãos passaram a realizar análise associativa, análise de chamadas telefônicas, gráficos de fluxo de eventos e análise investigativa visual [11].

Desde então, diversos países e departamentos de polícia no mundo vêm adotando a análise criminal como apoio às suas ações de prevenção e de repressão ao crime. Nas diferentes agências de polícia buscou-se estruturas de análise criminal e, como os Estados Unidos e o Canadá, a atividade de analista criminal passou a ser em uma carreira profissional específica. Nesse ínterim, foram criadas também diversas associações com a finalidade principal de difundir a análise criminal e padronizar a atividade[10]. Dentre essas associações destacam-se a *International Association of Law Enforcement Intelligence Analysts* (IALEIA) e a *International Association of Crime Analysts* (IACA).

No Brasil é recente o uso da análise criminal para planejamento de segurança pública e poucas são as polícias que de fato aplicam seu policiamento orientado a análise criminal. No ano de 2003, a Secretaria Nacional de Segurança Pública (SENASP), lançou o Sistema Nacional de Estatística de Segurança Pública e Justiça Criminal com a pretensão de consolidá-lo como a grande base nacional de dados estatísticos sobre o crime e a criminalidade [12]. Mas ainda nem todas as polícias possuem sistemas para análise de dados criminais nem tão pouco pessoal qualificado para tal. Espera-se que pouco a pouco, a análise criminal se torne tema de discussão nas universidades, nos órgãos da justiça e organizações policiais.

2.1.3 Teorias Criminais de Suporte à Análise Criminal

Ao longo da história, o homem realizou várias tentativas no sentido de explicar o comportamento de indivíduos desviantes, em especial dos criminosos. As diversas teorias elaboradas podem ser classificadas em cinco grupos, sendo elas: as teorias que focam a patologia individual; as teorias de base econômica; as teorias que defendem que o crime é fruto de uma sociedade perversa; as teorias que consideram que o crime é fruto da desorganização social; e as teorias que defendem que o crime está relacionado a fatores ambientais [13].

No contexto da análise criminal, as teorias ambientais (criminologia ambiental) recebem destaque, uma vez que as teorias tradicionais se focam em explicações relacionadas ao desenvolvimento do indivíduo (fatores psicológicos, sociais e biológicos) e fornecem poucas ou nenhuma solução que esteja ao alcance dos órgãos de segurança pública, seja pelo fato de encontrarem em níveis de decisão e ação que estão fora da alçada das polícias ou pelo

simples fato de não ter valor prático para estes órgãos, dependendo de outras esferas da sociedade [9] [13].

A criminologia ambiental não busca explicar as origens do crime, ou por qual motivo um indivíduo se transformou em um criminoso, ela busca entender as diversas variáveis envolvidas no fenômeno, procurando identificar padrões de comportamento e fatores ambientais que criam as oportunidades para o acontecimento do crime [14].

A perspectiva da criminologia ambiental é fundamentada em três premissas:

1. O comportamento criminal é influenciado pelo tipo de ambiente onde ele ocorre;
2. A distribuição do crime no tempo e no espaço não é aleatória, pois depende do ambiente criminal. Os padrões de crimes estão mais concentrados em ambientes que lhe sejam mais favoráveis;
3. O entendimento do papel do ambiente no crime e dos padrões criminais é uma poderosa ferramenta para a investigação, controle e prevenção do crime.

2.1.3.1 A Teoria da Escolha Racional

A Teoria da Escolha Racional defende que o comportamento criminoso não é aleatório, e sim a consequência de uma atitude racional e intencional, resultante da decisão que toma ao levar em conta os benefícios e custos de cometer um delito. Nesse sentido, o indivíduo pretendo a cometer um delito avalia e seleciona as formas ideais para alcançar seu propósito criminoso, minimizando os riscos e aumentando a possibilidade de sucesso [15].

Os teóricos dessa teoria creem que alguns fatores estão envolvidos no processo de decisão do criminoso, quais sejam [13]:

- O fácil acesso ao objeto do crime;
- A baixa probabilidade de ser descoberto;
- A utilidade do objeto do delito para o perpetrador;
- O sentimento de anonimato durante a prática do delito.

2.1.3.2 A Teoria da Atividade de Rotina

A Teoria da Atividade de Rotina é uma teoria complementar a Teoria da Escolha Racional, e procura explicar as formas que os indivíduos desviantes se utilizam para encontrar oportunidades, vítimas ou alvos durante o seu dia a dia. Essa teoria defende que o criminoso tem uma rotina própria ou um hábito que é costumeiro quando ele escolhe vítima ou quando pratica os delitos [13].

Sob o ponto de vista da criminologia ambiental, em especial no âmbito da teoria da atividade da rotina e da teoria da oportunidade, Cohen e Felson [16] explicam que três elementos devem convergir no tempo e no espaço para que um crime ocorra: um ofensor motivado a cometer um ato ilegal, alvo apropriado que tenha valor para o criminoso e a ausência de um guardião que possa impedir o delito (Figura 2.2). Essa abordagem posteriormente ficou conhecida como Triângulo do Crime [17](Figura 2.3). Em 2003, John Eck, depois de pesquisar sobre vitimização, acrescentou ao modelo um novo triângulo externo, que aponta elementos que devem ser agregados ao triângulo original para que a possibilidade de ocorrência do crime seja mitigada.



Figura 2.2: Triângulo do Crime de Felson (Adaptado) (Fonte: [17]).



Figura 2.3: Triângulo do Evento Criminal(Adaptado) (Fonte: [13]).

2.1.3.3 A Teoria do Padrão Criminal

A Teoria do Padrão Criminal sugere que o crime tem uma probabilidade maior de acontecer nas áreas geográficas onde há interseção das atividades rotineiras do ofensor (um local que o criminoso tem em sua consciência) com as atividades da vítima em potencial. Dessa forma, o perpetrador terá segurança em praticar o delito, pois conhece o ambiente, terá firmeza que sua atuação será oportuna, e dificilmente interrompida e que o alvo estará no local desejado em um momento também oportuno [15].

Essa teoria também está relacionada à teoria de atividade de rotina e se estabelece sobre três conceitos principais: nós, caminhos e fronteiras (bordas). Os nós se referem aos pontos de coincidência das pessoas, isso é, intersecção das direções de onde estão vindo e para onde estão indo; as bordas são as barreiras do ambiente que impedem a mobilidade do indivíduo; e os caminhos são os seguimentos que levam de um nó ao outro [13].

2.1.3.4 A Teoria das Janelas Quebradas

A Teoria das Janelas Quebradas, criada por Wilson e Kelling no ano de 1982, defende que há uma relação de causalidade entre desordens sociais e eventos criminais, de forma que as primeiras levam às segundas, em um tipo de desenvolvimento sequencial. Os autores relatam que pequenos desvios como jogar lixo nas ruas, ou beber bebida alcoólica em público, ou ter ambientes com pichações, se não forem reprimidos, podem levar a crimes [1].

Nesse contexto é conveniente ressaltar que o crime é considerado o evento mais danoso e ruidoso de um comportamento antissocial, enquanto a desordem é a incapacidade da comunidade de agregar valores comuns de seus membros e manter um efetivo controle social [18] [19].

Portanto, segundo essa teoria, se os pequenos desvios forem combatidos, manter-se-á a ordem e prevenir-se-á o crime. O exemplo clássico dado por essa teoria é que se uma janela de uma fábrica ou prédio é quebrada e não é imediatamente consertada, os indivíduos que passam pelo local tendem a concluir que o ambiente está largado, que não há ninguém zelando pela manutenção da ordem. Dessa forma, se sentem à vontade para quebrar as outras janelas, o que leva a decadência do local, isso é, com pequenas desordens levando a grandes desordens [1].

2.1.4 Tipos de Análise Criminal

A IACA relata que os tipos de análise criminal se organizam em torno de vários fatores, incluindo: a natureza e a fonte dos dados, as técnicas aplicadas, os resultados da análise, a regularidade e a frequência da análise, bem como o público-alvo e a finalidade. Ressaltam

que nenhuma tipologia poderá resultar em um conjunto de definições completamente exclusivas ou exaustivas. E que o objetivo da classificação que sugerem tem por finalidade fornecer uma estrutura para o foco e o trabalho dos analistas criminais, auxiliando na formação de novos analistas, e apoiando as atividades profissionais e facilitando a descrições de cargos, funções e responsabilidades [8].

Os tipos de análise criminal mais conhecidas são [7]: Análise Criminal Tática, Análise Criminal Estratégica; Análise Criminal de Operações e Análise Criminal Administrativa. Sendo que cada um desses tipos possui várias subcategorias e processos [8].

2.1.4.1 Análise Criminal Estratégica

A Análise Criminal Estratégica é o estudo de fenômenos criminais e suas influências direcionado ao desenvolvimento e avaliação de estratégias, políticas e técnicas de prevenção de longo prazo [8] [20]. Esta análise tem como principal objetivo o estudo e identificação de tendências criminais e de desordem. Dentre seus principais propósitos estão:

- Auxílio na formulação de políticas públicas;
- Planejamento e desenvolvimento de soluções;
- Auxílio no direcionamento de investimentos;
- Auxílio na elaboração de plano orçamentário;
- Formulação de indicadores de desempenho;
- Auxílio na identificação dos problemas de crime, violência e desordem, ao longo de período expressivo (no mínimo 6 meses); e
- Avaliação dos procedimentos organizacionais.

2.1.4.2 Análise Criminal Tática

A Análise Criminal Tática é o estudo de fenômenos criminais e suas influências, direcionado ao desenvolvimento do patrulhamento policial e ao emprego de recursos a médio prazo [8] [20]. Nesse tipo de análise estão incluídos a análise do espaço, do tempo, do ofensor, da vítima e *modus operandi*, dentre outros. Dentre e seus principais propósitos estão:

- Estudo das relações entre ocorrências criminais para identificar padrões criminais;
- Produção de informação que possa orientar o policiamento ostensivo nas ações preventivas e repressivas;

- Realização da análise dos padrões encontrados, com fins de identificar potenciais suspeitos; e
- Auxílio na identificação de estratégias de enfrentamento dos problemas de crime e desordem.

2.1.4.3 Análise Criminal de Operações

A Análise Criminal de Operações, segundo Gottlieb [11], é:

"O estudo analítico dos serviços prestados pela organização de segurança pública, realizado para prover aos comandantes e gestores policiais uma base científica para a tomada de decisão ou ação que possibilite a melhoria da alocação de recursos e da qualidade das operações".

Santos [7] afirma que a análise criminal de operações trata questões internas e operacionais dos órgão de segurança, tais como: o quantitativo e a tipificação de chamadas de emergência, o quantitativo de notificações de infrações de trânsito, o tempo de resposta no atendimento de chamados, a distribuição geográfica da viaturas policiais, entre outras.

2.1.4.4 Análise Criminal Administrativa

A Análise Criminal Administrativa, segundo Santos [7], é:

"A apresentação dos resultados de um estudo ou pesquisa sobre crime, baseados em preocupações legais, políticas e práticas, para informar diferentes públicos sobre o que lhes interessar sobre o crime e desordem".

Dentre os seus principais propósitos estão [20]:

- Elaboração de estatística descritiva;
- Elaboração de informações gerais sobre tendências criminais;
- Comparações de dados com períodos semelhantes do passado;
- Comparações de dados com outras cidades similares; e
- Oferecimento de informações para diversos públicos, tais como, cidadãos, gestores públicos, instituições públicas, etc;

2.1.5 Processo de Análise Criminal

O Processo de Análise Criminal pode ser entendido como um processo de produção de conhecimento continuado e cíclico, que realiza a coleta de dados de diferentes fontes, para identificar tendências, padrões e séries de delitos que já ocorreram ou que estão ocorrendo, ou para identificar características de pessoas suspeitas [21]. Percebe-se que análise criminal

pode fazer uso das ferramentas de estatística para alcançar seu propósito. Gottlieb [22] e Santos [7] estabelecem que a análise criminal pode ser realizada por meio de uma análise estatística, composta de cinco etapas, que são:

1. Coleta dos dados - consiste na obtenção ou busca de dados sobre os delitos desejados, sobre as vítimas, sobre os locais e horários de ocorrência, dentre outras informações relacionadas ao fenômeno desejado. As pesquisas são realizadas em bancos de dados de ocorrências policiais ou outros que estejam à disposição do analista;
2. Colagem e organização dos dados - consiste na compilação e organização dos dados coletados para fins de comparação, confronto, produção de mapas e tabelas, verificação de frequências e tratamentos estatísticos. As atividades realizadas nessa fase podem ser divididas em três grupos: limpeza dos dados, geocodificação dos dados e criação de novas variáveis ou recodificação das existentes;
3. Análise dos dados - consiste em se utilizar as diferentes técnicas estatísticas e de visualização de dados para sistematizar conclusões sobre o fenômeno pesquisado;
4. Disseminação dos dados - consiste na comunicação dos resultados obtidos após a análise;
5. *Feedback* e Avaliação - consiste no recebimento e crítica das avaliações feitas pelas pessoas que receberam a análise produzida.

Gottlieb [22] e Santos [7] não incluem o desenvolvimento e implementação de estratégias como uma fase do processo de análise criminal, no entanto a Figura 2.4 a apresenta como uma etapa, pelo fato de que ela é o objeto fim da análise criminal, como já mencionado anteriormente no presente trabalho.

2.2 Sistema de Informação Geográfico

O processo de análise criminal moderno, na realização de seu estudo sistemático de dados de crimes e de desordem, tem lidado com um volume muito grande de dados e de informações. Dessa forma, a análise manual desses dados tem se tornado inviável e os analistas criminais buscam, cada vez mais, suporte no uso de Tecnologia da Informação (TI), para relacionar as diversas características de crimes e desordem, identificando padrões e tendências que dificilmente seriam identificadas manualmente.

Com isso, tornou-se comum o armazenamento de dados referentes aos boletins de ocorrência e de registros de chamados de emergência em Sistemas de Gerenciamento de Bancos de Dados (SGBD). Esses sistemas, quando bem projetados, possibilitam a coleta e

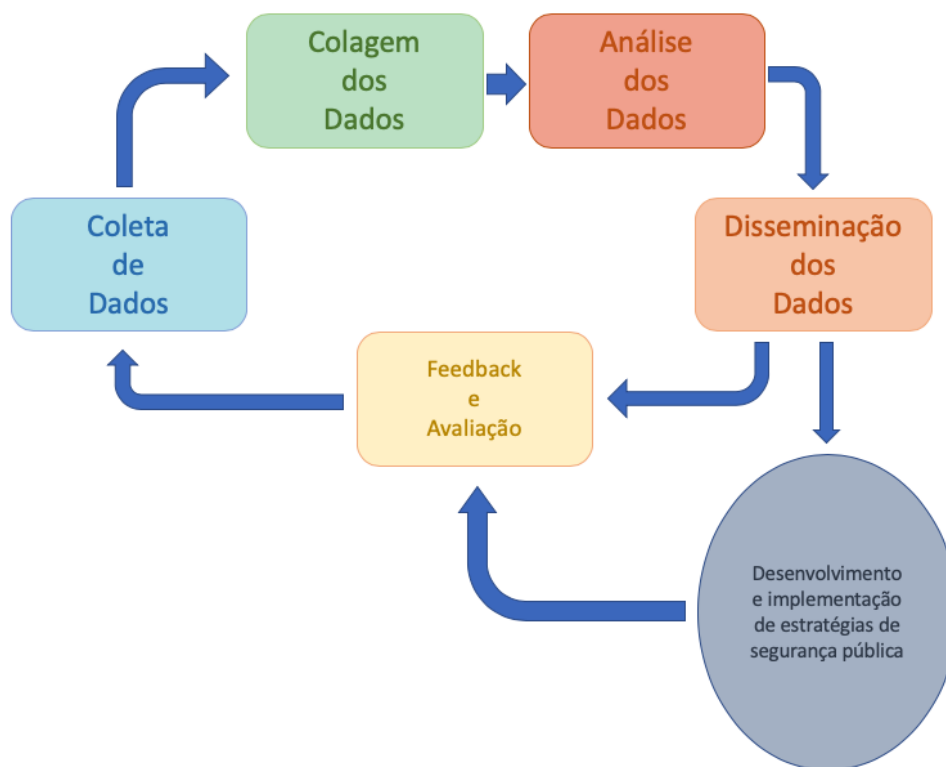


Figura 2.4: Processo de Análise Criminal segundo Santos (Adaptado) (Fonte: [7]).

o armazenamento de dados de crimes e de desordem de forma categorizada e em diversos campos, que posteriormente viabilizam ao analista criminal a seleção daquilo que de fato é relevante em sua análise [7].

Outra ferramenta de TI que tornou-se igualmente comum no processo de análise criminal foi o SIG, que é utilizado tanto na fase de colagem e organização dos dados quanto na fase de análise desses dados. O SIG potencializa o mapeamento criminal, pois permite: a análise espacial do fenômeno criminogênico, a manipulação de seus dados e a visualização de concentrações e padrões de crimes, provenientes de modelos estatísticos [23].

2.2.1 Conceitos Básicos

Segundo Worboys e Duckham [24], SIG são sistemas computacionais capazes de capturar, armazenar, manipular, recuperar, analisar e apresentar dados referenciados geograficamente. E estes, por sua vez, são dados que representam qualquer evento geográfico cuja localização está associada a uma posição sobre a superfície da terra [25].

Os SIGs podem ser vistos como um conjunto de elementos composto por seis partes [26]:

- As pessoas: são os indivíduos que interagem com o sistema, ou seja, usuários, projetistas, programadores, mantenedores, dentre outros;
- A rede: é o meio pelo qual se dá a comunicação e a troca de informações digitais;
- A base de dados: é o repositório dos dados do sistema, que possui informações relacionadas à representação digital de uma série de características da superfície da terra;
- O *hardware*: é o equipamento com o qual as pessoas interagem diretamente, por meio do qual realizam as operações;
- O *software*: são os programas executados sobre o *hardware* e viabilizam o uso do sistema;
- O sistema de gerenciamento: são as regras, políticas e procedimentos usados para garantir o funcionamento do sistema dentro do planejado e conforme as necessidades de implantação do sistema.

O SIG possui a capacidade de manipular dados geo-referenciados e não geo-referenciados de modo integrado [27], o que torna possível o mapeamento, a análise de fenômenos criminogênicos e o relacionamento destes com vários outros fenômenos sociais. Isso fica claro pela forma que Lisboa Filho [28] caracteriza o dado geo-referenciado; segundo o autor um dado geográfico deve possuir uma descrição do fenômeno, uma localização, o relacionamento espacial com outros fenômenos e o instante ou intervalo de tempo em que o fenômeno existe ou é válido.

Câmara e Davis [29], apresentam a arquitetura de um SIG como sendo a composição de: interface; entrada e integração dos dados; consulta e análise espacial; visualização e plotagem; e gerência dos dados espaciais (Figura 2.5).

2.2.2 Sistema de Gerenciamento de Bancos de Dados

Ramakrishnan e Gehrke [30] definem um Sistemas de Gerenciamento de Bancos de Dados (SGBD) como sendo “um *software* projetado para auxiliar a manutenção e a utilização de vastos conjuntos de dados”. Neste mesmo sentido, pode-se definir os Sistemas de Bancos de Dados Espaciais (SBDE), como sendo sistemas gerenciadores de bancos de dados que gerenciam dados com representação geométrica. Esses sistemas são capazes de armazenar coleções de dados referentes aos objetos espaciais, que são representações de

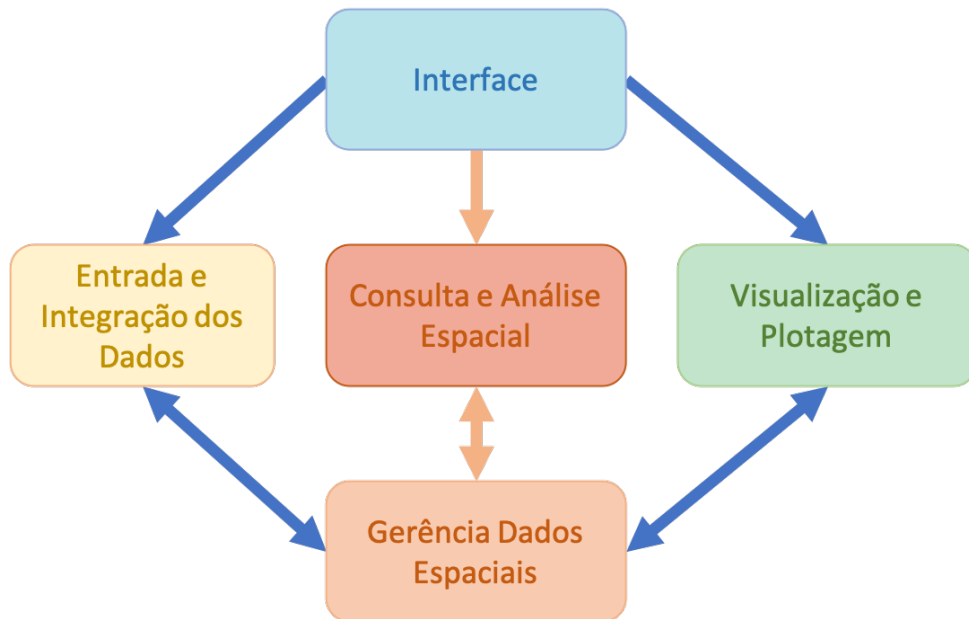


Figura 2.5: Arquitetura de um SIG (Adaptado) (Fonte: [29]).

objetos do mundo real [31]. Como principais exemplos de SBDE disponíveis no mercado tem-se: MySQL Spatial, Oracle Spatial, PostGIS e SpatialLite.

Para o presente projeto, adotou-se o SGBD PostgreSQL [32] com a extensão espacial PostGIS [33], conforme previsto no projeto “Arquitetura para Gerenciamento de Dados Colaborativos para Secretaria de Estado da Segurança Pública e da Paz Social baseada em Sistema de Informação Geográfico com Participação Popular” apresentado a Fundação de Apoio a Pesquisa do Distrito Federal (FAP-DF) (protocolo no 19781.82.31496.17082017).

2.2.2.1 PostgreSQL

O PostgreSQL é um Sistema de Gerenciamento de Banco de Dados Objeto-Relacional (SGBD-OR)¹ de código aberto, com mais de 30 anos de desenvolvimento, e possui uma arquitetura com características de confiabilidade, integridade de dados, extensibilidade e com um conjunto robusto de recursos [32]. Foi desenvolvido pelo Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley com base no Postgres versão 4.2 [34].

Pode ser executado nos principais sistemas operacionais, incluindo UNIX, Mac OS X, Linux, Solaris, Windows, dentre outros. É compatível com ACID², desde 2011, possui suporte

¹Um banco de dados objeto-relacional (DOR), ou SGBD-OR, é um sistema de gerenciamento de banco de dados relacional que permite aos desenvolvedores integrar ao banco de dados seus próprios tipos de dado e métodos.

²ACID é a definição relacionada às quatro propriedades de transação de um sistema de banco de dados: Atomicidade, Consistência, Isolamento e Durabilidade.

a diversos tipos de dados: Integer, Numeric, String, Boolean, Date/Time, Array, Range, UUID, JSON/JSONB, XML, Key-value, Point, Line, Circle, Polygon, dentre outros. Dá suporte à chave primária e à chave estrangeira, ao controle de concorrência multiversão (Multi-Version Concurrency Control - MVCC), à autenticação: Generic Security Services Application Program Interface (GSSAPI), Security Support Provider Interface (SSPI), Lightweight Directory Access Protocol (LDAP), Salted Challenge Response Authentication Mechanism (SCRAM) SHA-256 [32]. E além dos citados, possui diversos outros recursos e funcionalidades, mas, em especial no nosso estudo, é importante destacar seu potencial de extensibilidade, que permitiu a criação do PostGIS, uma extensão para dados geográficos [35].

2.2.2.2 PostGIS

O PostGIS foi desenvolvido pela Refractions Research Inc como um projeto de pesquisa na área de banco de dados espacial de código aberto [36]. O PostGIS é disponibilizado sob a Licença Pública Geral (GNU GPL) [35]. Dá suporte a objetos geográficos, de forma que as consultas de localização possam ser executadas via *Structured Query Language* (SQL)³ e possui as seguintes características [36]:

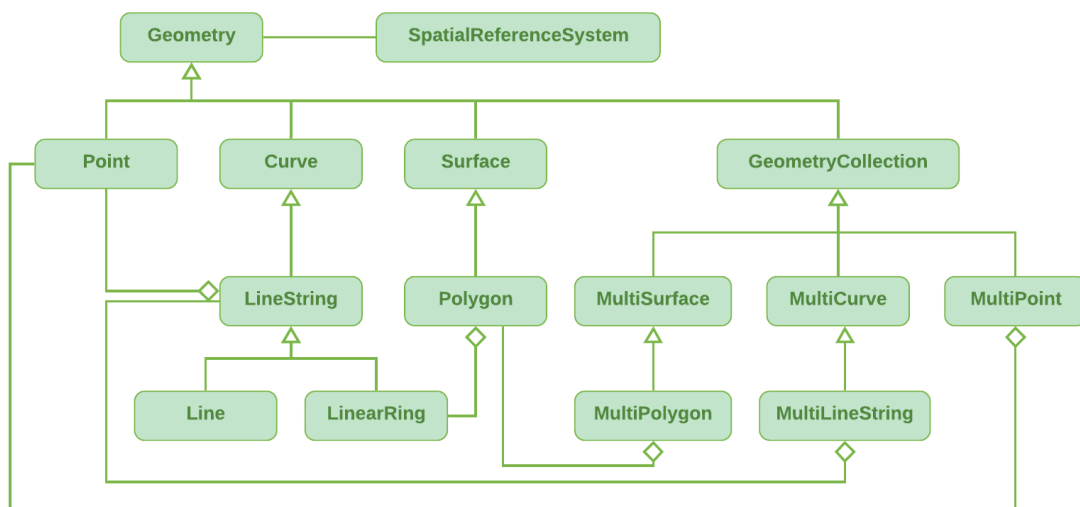


Figura 2.6: Hierarquia de classes de geometrias do OGC (Adaptado) (Fonte: [37]).

³SQL significa Linguagem de Consulta Estruturada, e é uma linguagem de pesquisa declarativa padrão para banco de dados relacional

- Dá suporte a objetos e funções do padrão de especificação SFS-SQL⁴ do *Open Geospatial Consortium* (OGC)(Figura 2.6);
- Dá suporte a duas formas de manipulação de objetos geográficos: *Well-Known Text* (WKT)⁵ e *Well-Known Binary* (WKB)⁶;
- Usa GiST⁷ para realizar uma indexação espacial mais rápida;
- Realiza análise geoespacial usando funções da biblioteca GEOS.

O PostGIS implementa os tipos de dados espaciais: Point, LineString, Polygon, Multi-Point, MultiPolygon, MultiLineString e GeometryCollection (Figura 2.7).

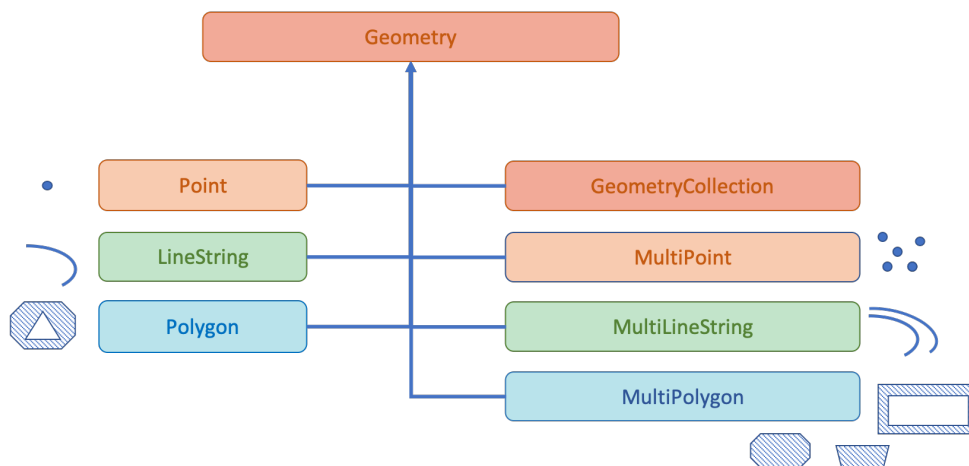


Figura 2.7: Tipos de dados espaciais do PostGIS (Adaptado) (Fonte: [35]).

2.2.3 Sistema de Informação Geográfico com Participação Popular

Segundo Schlossberg e Shuford [38], apesar de existirem diversos estudos sobre *Public Participation Geographic Information System* (PPGIS), não existe um consenso entre os estudiosos sobre a definição exata de PPGIS. Esse termo foi criado em 1996, nos Estados

⁴SFS é um documento emitido pela OGC que contém a descrição de predicados e operadores espaciais. Sendo os predicados espaciais aquelas funções que comparam dois objetos espaciais e retornam um resultado booleano indicando a existência (ou ausência) de um relacionamento espacial particular. E os operadores espaciais as funções que recebem dois objetos geométricos e retornam um novo objeto derivado dos primeiros.

⁵ WKT é uma linguagem de marcação de texto para representar objetos vetoriais geométricos em um mapa.

⁶ Equivalente binário de WKT

⁷ *Generalized Search Tree*, ou Árvore de busca genérica, é uma estrutura de dados que facilita a busca de dados.

Unidos, em um encontro do *National Center for Geographic Information and Analysis* (NCGIA), e é comumente usado para se referir a SIGs que produzem dados geográficos por meio da participação de populares [6].

A finalidade dos PPGIS é facilitar a participação do usuário de serviços públicos, para que estes, de forma voluntária, possam fornecer informações relacionadas ao seu cotidiano e ao ambiente em que vivem, melhorando a qualidade de informações que os gestores possuem e participando do planejamento urbano [39]. Nas palavras de Bugs [40], os PPGIS são “uma alternativa através da qual o público participa produzindo mapas e/ou dados espaciais que representam a sua percepção do espaço urbano em questão”.

Nos últimos anos, os PPGIS foram potencializados, tendo em vista, a popularização dos *smartphones* e o crescente acesso à internet. Essas inovações permitiram que o usuário pudesse criar conteúdo e interagir com outros usuários em tempo real [29]. O que tornou possível o uso de georreferenciamento, por qualquer indivíduo que possua um aparelho *smartphones*, de forma que este pudesse associar informações a quaisquer locais geográficos [41]. Alguns exemplos de PPGIS, como o WikiCrimes, o UnB Alerta, o B.O. Coletivo e o OndeFuiRoubado, são analisados no Capítulo 3 do presente trabalho.

2.3 REST e RESTful

O *Representational State Transfer* (REST) ou Transferência de Estado Representativo, é um estilo de arquitetura para sistemas hipermídia distribuídos [42], e tem sua origem na tese de doutorado de Roy Fielding, um dos responsáveis pela especificação do *Hypertext Transfer Protocol* (HTTP). Os serviços web baseados em REST são guiados pelas boas práticas de uso do protocolo HTTP: uso correto de cabeçalhos *Hypertext Transfer Protocol* (HTTP), uso correto de *Universal Resource Locator* (URL), uso adequado dos métodos *Hypertext Transfer Protocol* (HTTP), uso de códigos de *status* padronizados para representação de sucessos ou falhas e interligações entre vários recursos diferentes [43].

Os sistemas desenvolvidos usando o estilo de arquitetura REST são conhecidos como RESTful [44] e segundo Fielding [42]:

- Devem seguir o modelo de comunicação cliente-servidor;
- A comunicação entre o cliente e o servidor deve ser *stateless*, isso é, não é armazenado no servidor qualquer estado da sessão;
- As respostas do servidor são sinalizadas com *cacheable* ou *non-cacheable*, para indicar se o cliente pode ou não utilizar novamente os dados já recebidos em outras requisições, o que aumenta a eficiência da rede; e

- Todos os componentes têm interface uniforme, isso é, deve satisfazer as seguintes: mensagens auto-descritivas, identificação de recursos, manipulação de recursos por meio de representações e hipermídia como o motor do estado da aplicação.

Para manter a restrição de interface uniforme entre componentes, propiciando uma arquitetura mais simples e maior visibilidade das interações, a forma mais habitual é a de padronizar a interface por meio dos métodos do protocolo HTTP (Tabela 2.1). Este protocolo viabiliza a transferência de mensagens entre clientes e servidores, por meio da *World Wide Web* (WWW), e é muito utilizado na *web* para troca de dados. Permite que várias operações sejam realizadas por meio de seus métodos, que podem executar operações de consulta, inclusão, exclusão e atualização de dados ou recursos, de forma semelhante a alguns comandos SQL [45].

Tabela 2.1: Principais métodos internos de solicitações HTTP e comandos semelhantes em SQL (Adptado) (Fonte: [45]).

Descrição	Equivalente SQL	Método HTTP
Obtém a representação de um recurso	SELECT	GET
Cria ou atualiza um recurso	Create/ Update	PUT
Cria recurso	Create	POST
Remove um recurso	Delete	DELETE

2.4 Sistema Operacional iOS

O iOS é o sistema operacional criado pela Apple Inc. para rodar somente em seus dispositivos móveis: iPhone, iPad, iPod touch e, mais recentemente, Apple Watch. Quando foi lançado, em 2007, era chamado de iPhone OS, pois foi originalmente desenvolvido para o primeiro iPhone [46]. Foi criado com base nas mesmas tecnologias usadas pelo Mac OS X Tiger e MAC OS X Leopard [47], isso é, o kernel e as interfaces BSD, tornando-o um sistema operacional baseado no UNIX [48]. É o segundo sistema operacional para dispositivos móveis mais popular no mundo, ficando atrás apenas do Android [46] (Figura 2.8).

2.4.1 Arquitetura

O iOS foi construído com base no MAC OS X, no entanto, existem diferenças tecnológicas entre os dois sistemas operacionais, como é o caso da interface Multi-Touch, que está presente somente no iOS [50]. A plataforma iOS foi concebida sobre uma arquitetura em camadas, onde cada camada oferece um conjunto de *frameworks* que podem ser utilizados para o desenvolvimento de aplicativos para a própria plataforma [51]. Este

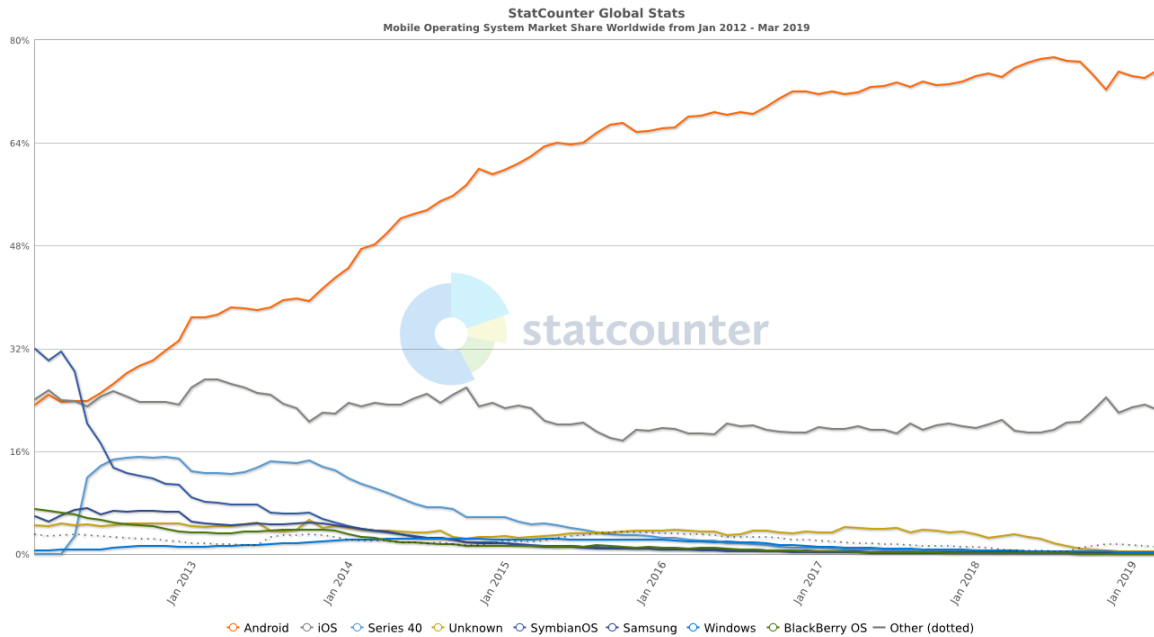


Figura 2.8: Participação no Mercado dos Sistemas Operacionais Móveis (Fonte: [49]).

sistema operacional funciona como um intermediário entre os aplicativos e o *hardware* [52](Figura 2.9).

A arquitetura do iOS é composta de quatro camadas: *Core OS*, *Core Services*, *Media* e *Cocoa Touch*, uma variação da camada Cocoa do OS X com a inclusão da funcionalidade multi-touch (Figura 2.10). Nas camadas mais inferiores estão presentes os serviços essenciais dos quais todos os aplicativos dependem. Já nas camadas mais superiores estão os serviços mais sofisticados [53].

2.4.1.1 Camada *Cocoa Touch*

A camada *Cocoa Touch* é a de mais alto nível, nela são encontrados os principais *frameworks* para os aplicativos. Ela disponibiliza a infra-estrutura para as tecnologias essenciais do iOS, oferecendo funcionalidades como: multitarefa, reconhecimento gestual (*touch*), serviço de notificação Apple por *push* e interface gráfica [54].

Os *frameworks* oferecidos por esta camada são [53]:

- iAd Framework;
- UIKit Framework;
- Twitter Framework;
- Map Kit Framework;

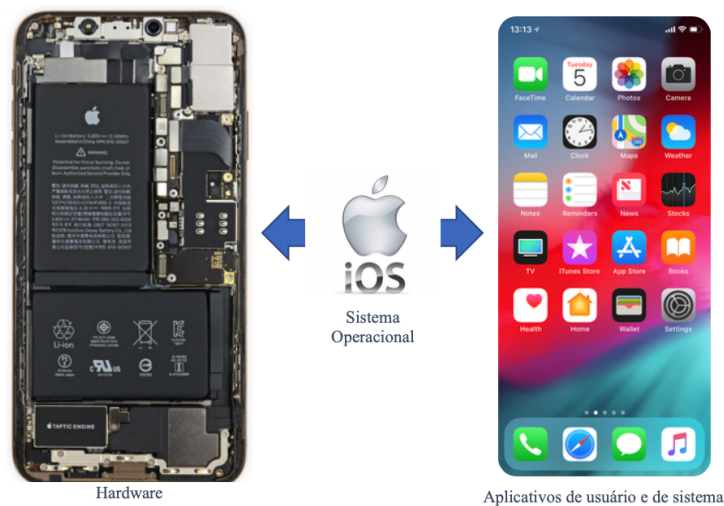


Figura 2.9: Interação entre o *hardware*, o iOS e aplicativos.

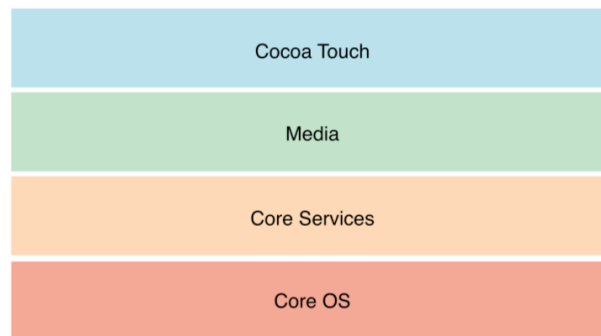


Figura 2.10: Arquitetura do iOS (Fonte: [53]).

- Game Kit Framework;
- Event Kit UI Framework; e
- Address Book UI Framework.

2.4.1.2 Camada *Media*

A camada *Media* é a camada responsável pelas tecnologias de áudio, vídeo e gráficos. A Apple esclarece que a camada *Media* foi projetada para melhorar a experiência multimídia em um dispositivo móvel, permitindo que aplicativos sejam desenvolvidos com excelente aparência e qualidade de som [53].

As tecnologias gráficas oferecidas pela camada de *Media* permite que o desenvolvedor de aplicativos iOS faça desde animações simples até animações mais complexas usando as tecnologias [53]:

- Core Graphics Framework: faz renderização de vetores a partir de imagens 2D;
- OpenGL ES Framework: fornece ferramentas para desenhar conteúdo 2D e 3D;
- Core Text Framework: contém um conjunto de interfaces de alto desempenho para *layout* de texto e manipulação de fontes;
- Image I/O Framework: fornece interfaces para importar e exportar dados de imagem e metadados de imagem;
- Assets Library Framework: fornece acesso a fotos e vídeos na biblioteca de imagens e vídeos do usuário.

Além disso, oferece várias maneiras para reproduzir e gravar conteúdo de áudio usando os *frameworks*[53]:

- Media Player Framework: fornece suporte de alto nível para reproduzir conteúdo de áudio e vídeo do seu aplicativo e fornece acesso à biblioteca do iTunes;
- AV Foundation Framework: contém classes Objective-C para reproduzir conteúdo de áudio;
- Core Audio: é uma interface baseada em C que suporta a manipulação de áudio estéreo.

Também são oferecidas, por esta camada, tecnologias para reprodução e gravação de vídeo que dão suporte às extensões de arquivo mov, mp4, m4v e .3gp [52].

2.4.1.3 Camada *Core Services*

A camada *Core Services* abarca os serviços fundamentais às aplicações e possui acesso às funcionalidades de mais baixo nível como localização, telefonia, *threads*, dentre outras. Segundo a Apple [53], ainda que o desenvolvedor não use diretamente os serviços dessa camada, várias partes do sistema as usarão indiretamente. Ela oferece diversos *frameworks*, dentre eles pode-se destacar os seguintes:

- CFNetwork Framework: é um conjunto de interfaces que utiliza a abstração orientada a objetos para comunicar com protocolos de rede;
- Core Location Framework: fornece informações de localização e direção para os aplicativos, usando GPS ou Wi-Fi para encontrar a longitude e a latitude atual do usuário;
- Event Kit Framework: fornece uma interface para acessar eventos da agenda no dispositivo de um usuário;

- Store Kit Framework: fornece suporte para a compra de conteúdo e serviços de dentro dos seus aplicativos iOS;
- Core Foundation Framework: é um conjunto de interfaces baseadas em C que fornecem recursos básicos de gerenciamento e serviço de dados para aplicativos iOS. Ele inclui suporte para: tipos de dados como *array*, gerenciamento de *string*, gerenciamento de data e hora, gerenciamento de preferências, dentre outros.

2.4.1.4 Camada *Core OS*

A camada *Core OS* é a mais inferior das quatro camadas e é considerada o núcleo do sistema operacional [51]. É usada pelas camadas superiores por meio de seus *frameworks*. E normalmente o desenvolvedor usa os *frameworks* dessa camada quando precisa lidar diretamente com segurança ou comunicação com acessório de *hardware* externo [52]. A camada possui as seguintes tecnologias:

- Accelerate Framework: contém interfaces para executar sinais digitais, álgebra Linear e fazer cálculos de processamento de imagem;
- Core Bluetooth Framework: permite que os desenvolvedores interajam especificamente com acessórios *Bluetooth*;
- External Accessory Framework: fornece suporte para comunicação com acessórios de *hardware* conectados a um dispositivo iOS;
- Generic Security Services Framework: fornece um conjunto de serviços relacionados à segurança de aplicativos iOS;
- Security Framework: estrutura de Segurança explícita que pode ser usada para garantir a segurança dos dados gerenciados pelos aplicativos;
- System: O nível do sistema engloba o ambiente do *kernel, drivers* e interfaces de baixo nível UNIX do sistema operacional.

2.4.2 A Linguagem de Programação *Swift*

A linguagem de programação *Swift* foi lançada em 2014 pela Apple para o desenvolvimento de aplicativos para sistemas operacionais macOS, iOS, watchOS, tvOS e Linux [55]. A linguagem *Swift* é uma alternativa à linguagem Objective-C e contém uma sintaxe mais simples e intuitiva, mais rápida e fácil de programar, com fins de oferecer desempenho semelhante ao das linguagens baseadas em C, mas ao mesmo tempo, tornando mais fácil a escrita na manutenção do código [54]. Tem boa capacidade de escrita, sua sintaxe é

simples, limpa e de fácil leitura, ou seja, possui alta legibilidade. Tem características confiáveis, faz verificação de tipos e tratamento de exceções. E suporta portabilidade para os sistemas operacionais macOS, watchOS, tvOS e Linux.

O *Swift* foi inspirado em linguagens de programação como *Scala*, *C++11*, *Python*, *F#*, *C#*, *Haskell*, *Java*, *JavaScript* e *Ruby*. Possui uma sintaxe mais simplificada, não usa ponteiros e inclui melhorias em suas estruturas de dados e em sua sintaxe em relação ao *Objective-C*. *Swift* é uma linguagem orientada a objetos e imperativa como *Objective-C*, mas incorporou também o paradigma funcional [56]. Possui todos os tipos fundamentais de C e de *Objective-C*, incluindo *Int*, *Double* e *Float*, *Bool* e *String*. Fornece versões poderosas dos três principais tipos de coleções: *Array*, *Set* e *Dictionary*. E além dos tipos familiares, contém tipos avançados não encontrados no *Objective-C*, como tuplas [57].

2.4.2.1 Evolução da Linguagem

Desde que a primeira versão para desenvolvedores foi lançada em 2014, a linguagem evoluiu passando por diferentes versões, com mudanças na sintaxe e em funcionalidades. Sendo que a versão mais atual é a *Swift* 5.0, lançada em 25 de março de 2019 [55].

A primeira versão da linguagem se chamava *Swift 1.0 GM* e foi apresentada em 6 de junho de 2014. Era uma versão *Golden Master* (GM), pois a Apple pretendia continuar adicionando alterações e melhorias na linguagem. Essa versão apresentou muitas mudanças em sua sintaxe, nas bibliotecas nativas, nos tipos de algumas funções, possibilitou o uso de “opcional” com variáveis, dentre outras[56].

A segunda versão da linguagem, a *Swift 1.1*, foi lançada em 22 de outubro de 2014, e teve poucas alterações em relação a versão anterior. Adicionou à linguagem “*failable initializer*”, alterou alguns “protocolos” e algumas funcionalidades internas.

A linguagem *Swift 1.2* foi lançada em 8 de abril de 2015 e é considerada um novo marco para a linguagem, pois trouxe diversas novidades. Introduziu diferentes melhorias no compilador, tais como: compilações incrementais; melhoria na velocidade de compilação; melhoria no diagnóstico do compilador; e melhoria na estabilidade para evitar “*SourceKit warning*”. Inseriu novos recursos na linguagem, tais como: a palavra reservada “*as!*”; a possibilidade de exportar “*enumerations*” da linguagem *Swift* para a linguagem *Objective-C* com o atributo “*@objc*”; mudanças nas constantes (uso de *let*); e uma nova estrutura de dados *Set* [58].

A linguagem *Swift 2.0* foi anunciada em 8 de junho de 2015, na *Apple Worldwide Developers Conference* (WWDC), como parte do *Xcode7* e passou a ser de código aberto em 03 de dezembro de 2015. As principais atualizações da linguagem foram: a programação orientada a protocolos (*Protocol Oriented Programming*); novos mecanismos de *error handling*; foram feitas melhorias na sintaxe, criando novas palavras chaves para que a

escrita do código ficasse mais natural (`do`, `guard`, `defer` e `andrepeat`); *Availability checking*; *Pattern matching* [59]. Em seguida lançou-se a *Swift 2.1* e a *Swift 2.2*, que são atualizações menores que trazem a interpolação de `strings`, a eliminação dos operadores `++` e `-`, estilo de *loops* do C, comparação de tuplas, dentre outros [55].

A linguagem *Swift 3.0* foi um importante lançamento e como tradicionalmente é feito, foi lançada na WWDC16. Trouxe grandes mudanças e passou a ter portabilidade para macOS, iOS e Linux. Como as principais atualizações da *Swift 3.0*, pode-se citar: a retirada do prefixo `NS` do início do nome das classe que compunham a biblioteca *Foundation*; novos guias de nomeação de *Application Programming Interface* (API); o *framework Grand Central Dispatch* (GCD) [60]. As versões 3.1 e 3.2 são atualizações menores e aprimoram a linguagem adicionando o protocolo `Sequence`, múltiplos retornos de função; permite tipos aninhados com *generics*; extensão de `@availability` para usar na versão da *Swift*; e outros [55].

A linguagem *Swift 4.0* foi anunciada na WWDC17 e foi lançada em 19 de setembro de 2017 juntamente com o *Xcode 9*. Foi apresentada com as seguintes atualizações: implementa literais de cadeia múltipla; aperfeiçoou a forma de criação, uso e manipulação de tipos variados de *collections*; suporte para arquivamento e serialização; todos os tipos de *collections* passam a suportar *generic subscripts*; modo de compatibilidade do compilador com a versão 3.2 [61].

A linguagem *Swift 5.0* apresenta novos recursos que são blocos de construção para versões futuras, incluindo a reimplementação de `String`, acesso exclusivo à memória durante o tempo de execução, novos tipos de dados e suporte à tipagem dinâmica [62]. Além disso, nessa versão alcançou-se a estabilidade da interface binária de aplicação (ABI) que propiciou a redução dos aplicativos, uma vez que não precisarão mais incluir as bibliotecas das versões anteriores da *Swift*.

2.4.2.2 Algumas Características da Linguagem

A linguagem *Swift* introduziu várias mudanças em relação ao *Objective-C* e adicionou novas características, tais como: mudanças nas variáveis, modificação nas funções e métodos para incorporar múltiplos retornos e incorporou diversas características de programação funcional.

Constantes e Variáveis Simples

A linguagem *Swift* é fortemente tipada, faz inferência de tipos e obriga que todas as variáveis sejam inicializadas antes do primeiro uso. Além disso, a declaração de uma variável deve ser precedida por `var` e de uma constante por `let`. Ressalta-se que uma vez atribuído um valor a uma constante este não pode mais ser alterado [56].

Permite ao desenvolvedor especificar explicitamente o tipo de valor da variável ou deixar o compilador inferir o tipo, no entanto por ser fortemente tipada, quando o desenvolvedor define uma variável pela primeira vez, o compilador atribui o tipo e a partir disso, não se pode atribuir um novo valor com tipos diferentes àquela variável [56].

Funções

Todas as funções da linguagem *Swift* possuem um tipo, assim como os parâmetros de funções e os retornos de funções. O que facilita a passagem de funções como parâmetro para outras funções e o retorno de funções (objetos de primeira classe e funções de alta ordem). Essas funções podem retornar múltiplos valores ou nenhum valor, e podem ser escritas dentro de outras funções, encapsulando funcionalidades dentro do escopo da função aninhada. Além disso, a linguagem implementa as chamadas *Closures*, que é uma espécie de função que não precisa ser declarada, e é passada imediatamente como uma expressão [63].

Classes e Structs

Na linguagem *Swift* a sintaxe para criar uma classe ou uma *structs* é semelhante a de outras linguagens de programação como o C++, C# e Java. No entanto, não exige que crie-se arquivos separados de interface e implementação, pode-se definir uma *structs* ou classe no mesmo arquivo, e a interface externa dessa classe ou *structs* será automaticamente disponibilizada para outro código usar. As classes e as *Structs* podem: definir propriedades para guardar valores; Definir funções para comportamento; definir *subscripts* para ter acesso à valores dentro de elementos como *collection* e outros tipos de *sequences*; definir construtores para inicializar estados de variáveis; ser estendidas com a palavra reservada *extension*; e podem implementar *protocols* para ter suporte à programação orientada a *protocols*. As classes são ainda utilizadas para aplicar o conceito de herança e além do construtor(*init*) podem definir um destrutor (*deinit*) [64].

Operadores

A *Swift* suporta a maioria dos operadores de C e melhora vários recursos para eliminar erros comuns de codificação. De acordo com o tipo, seus operadores podem ser classificados em: unários (-, !), binários ((+, -, *, /) e ternários (a ? b : c). De acordo com as operações que fazem, seus operadores podem ser classificados em: operadores de atribuição(=, +=, -=, *=, /=, %=, <<=, >>=, &=, ^=, |=), operadores aritméticos (+, -, *, /, %), operadores de comparação (==, !=, >, <, <=, >=) e operadores lógicos (!, ||, &&). Além desses operadores, a linguagem *Swift* oferece o operador de intervalo fechado (a..b), que define um intervalo fechado de a a b, e os operador de intervalo

semi-aberto ($a..<b$), que é semelhante ao operador de intervalo fechado, mas não inclui b no intervalo [63].

A linguagem *Swift* permite ainda a definição de novos operadores usando os sinais aritméticos, sendo que para criar um novo operador, deve-se declarar seu cabeçalho e fazer sua implementação. São três as maneiras de se fazer um “Operador Personalizado”: prefixado, infixado e posfixado [56].

Generics

Generics é outra característica disponível na linguagem, também conhecida como polimorfismo paramétrico, ou como “*Templates*” em outras linguagens de programação como C++ e Java. É um dos recursos mais poderosos da linguagem.

O uso de *generics* pode evitar as conversões entre tipos, pois permite que o compilador execute checagem de tipo. Permite ao desenvolvedor escrever funções reusáveis e tipos que podem funcionar com qualquer tipo, sujeitos a requisitos definidos pelo desenvolvedor. Nas funções *generics*, o nome da função é seguida de um marcador de tipo ($<T>$) [57].

Tipos

Na *Swift* os tipos de objetos são divididos em dois grupos: tipos nomeados e tipos compostos. Tipos nomeados são tipos de objetos em que se permite a escolha de uma própria nomenclatura, eles incluem classes, estruturas, *enums* e protocolos. Já os tipos compostos são tipos sem nome, definidos na própria linguagem. São aqueles tipos de objetos que podem ter mais de um tipo associado, incluem matrizes, dicionários e opcionais. Existem dois tipos compostos: tipos de função e tipos de tupla [57].

Os tipos de dados que normalmente são considerados básicos em outras linguagens, na *Swift* são tipos denominados, definidos e implementados na biblioteca padrão *Swift* usando estruturas. E como são tipos nomeados, o desenvolvedor pode estender seu comportamento de acordo com as necessidades de seu programa, usando uma declaração de extensão [57].

Capítulo 3

O Estado da Arte

Neste capítulo é apresentada uma análise de algumas aplicações disponíveis no Brasil para a realização de mapeamento criminal ou mapeamento de desordem social com a participação popular. A análise apresentada nesse capítulo teve como fonte o estudo feito Passos e Holanda [41] no artigo intitulado “Mapeamento Colaborativo de Desordens Sociais por meio de Sistema de Informação Geográfico com Participação Popular no Brasil”. No estudo, os autores selecionaram e analisaram algumas aplicações disponibilizadas de forma gratuita e que estavam disponíveis aos brasileiros para realização de mapeamento criminal ou de desordem urbana. O capítulo está dividido nas seguintes seções: Seção 3.1 Ferramentas Analisadas; Seção 3.2 Análise das Aplicações; e Seção 3.3 Discussão Sobre as Ferramentas;

3.1 Ferramentas Analisadas

Nesta seção 3.1 são apresentadas as principais ferramentas que viabilizam o mapeamento de crimes ou de desordens sociais por meio de participação popular de forma gratuita e que estão disponíveis no Brasil.

3.1.1 *WikiCrimes*

O *Wikicrimes* é um sistema que permite ao usuário realizar registro de ocorrências criminais realizando marcação em uma determinada localização geográfica de um mapa digital [65]. Os criadores, na sua concepção, seguiram a mesma linha da Wikipédia, buscando a participação individual com vistas a gerar um conhecimento coletivo. O *Wikicrimes* foi criado com a filosofia de dar mais transparência e publicidade às informações de crimes, para que o cidadão possa se precaver quanto áreas de riscos e para minimizar a chamada cifra negra de delitos [66].

O sistema foi estruturado seguindo o padrão de arquitetura *Model-View-Controller* (MVC) e possui três camadas; a camada de apresentação que é a responsável por viabilizar a interação do usuário com o sistema por meio dos mapas, utilizando-se da API do *Google Maps*. A camada de Serviços que contém as regras de negócio, permitindo a inserção dos registros de crime, a identificação de usuários, dentre outros. É a camada de integração e acesso aos dados contidos no banco de dados que é o MySQL [67].

Os criadores do *Wikicrimes* desenvolveram algumas estratégias para garantir maior credibilidade às informações fornecidas pelos usuários. Tais estratégias visam evitar ou minimizar notificações realizadas com má fé, que sejam falsas ou que visem formar tendências criminais no mapa. Na tentativa de evitar isso, o sistema busca atribuir uma reputação a cada usuário, e busca permitir que cada notificação possa ser confirmada por outros usuários [66], além disso criou-se um algoritmo de mineração que permite identificar a formação de tendências maliciosas, com base na densidade de *Kernel*, identificando *hot spots* pela estimação do grau de concentração dos eventos criminosos notificados em determinada área [68].

3.1.2 OndeFuiRoubado

O “Onde Fui Roubado” é uma plataforma colaborativa que visa realizar o mapeamento e o monitoramento de criminalidade em diversas cidades brasileiras. As notificações de crimes são apresentadas em um mapa onde são inseridas de forma anônima. A plataforma permite a geração de dados públicos de maneira que o cidadão possa conhecer as áreas com os índices mais altos de criminalidade, permite ainda gerar estatística e possibilita a prevenção de vitimização em zonas críticas.

A plataforma pode ser acessada no site <http://www.ondefuiroubado.com.br> ou por meio do aplicativo que está disponível para plataforma iOS e Android [69].

3.1.3 UnB Alerta

A plataforma UnB Alerta foi desenvolvida na Universidade de Brasília por alunos da disciplina de Linguagem de Programação sob orientação do professor Marcelo Ladeira. O objetivo da ferramenta é reforçar a segurança no campus da universidade e busca facilitar o contato entre os alunos e os responsáveis pela segurança da universidade. A aplicação realiza o mapeamento de delitos noticiados pela comunidade universitária e facilita a identificação dos locais mais suscetíveis a ocorrência de crimes.

O UnB Alerta permite ver as ocorrências inseridas no sistema em um mapa, e gerar relatórios e gráficos para acompanhar as estatísticas sobre os denúncias de crimes. A

ferramenta está disponível por meio do site <https://unbalerta.unb.br> ou do aplicativo para Android [70].

3.1.4 QAP mobile

O *QAP Mobile* é um aplicativo de coleta de dados sobre ocorrências policiais e que está disponível para as plataformas *Android* e *iOS*. O aplicativo permite a exibição do mapa de ocorrências em uma dada região e suas estatísticas, a inserção de registros de ocorrência, e possibilita a emissão de alertas ao usuário quando este está se aproximando de áreas de risco ou quando o usuário delimitou um região para ser notificado sobre ocorrências naquele local. Os alertas são ativados por meio da função “Proteja-me” e são disparados automaticamente ao usuário, toda vez que o usuário se aproxima de um local onde é frequente a prática de crimes [71]. Além disso, possui filtro de registros e de estatísticas, que facilitam a visualização do mapa em regiões com muitas ocorrências. É disponibilizado nos idiomas inglês, português e espanhol.

3.1.5 B.O. Coletivo

O B.O. Coletivo é um aplicativo distribuído somente para a plataforma *Android*. Teve sua origem no evento *OuiShare Fest Porto Alegre*, no ano de 2013. A ideia inicial do grupo de ativismo urbano que criou o aplicativo era que por meio de cartazes contendo a frase “Aqui fui assaltado”, pessoas divulgassem pontos onde foram vítimas de violência. A partir dessa ideia evoluiu-se para divulgação de delitos por meio de uma página do *Facebook* e logo depois criou-se o aplicativo que tem como objetivo realizar o mapeamento colaborativo de locais onde ocorreram furtos, roubos e sequestros.

Segundo sua descrição na *Google Play* [72][73]:

“O B.O.Coletivo é um grupo de ativismo urbano que tem como tema central a segurança pública. Seu primeiro projeto, o “Aqui fui Assaltado”, consiste em um modelo de cartaz que pode ser baixado através da página do projeto <https://www.facebook.com/B.O.Coletivo>. Estes cartazes são colados de forma colaborativa em diferentes localizações de várias cidades brasileiras, permitindo que qualquer pessoa escreva o seu nome caso já tenha sofrido algum tipo de incidência naquele determinado local.”

3.1.6 Eu fui Roubado

O Eu fui Roubado é uma aplicação para a *WEB* com suporte a dispositivos móveis, disponíveis nas plataformas *iOS* e *Android*. O “Eu fui Roubado” viabiliza o mapeamento colaborativo de crimes e disponibiliza suas notificações em mapas onde os usuários podem escolher a cidade que desejam visualizar, a categoria de ocorrências e o período. As

categorias de ocorrências disponibilizadas no sistema são: lei seca, roubo de caminhão, roubo de carro, roubo de moto, saidinha de banco, sequestro e a categoria “outros” [41][74].

3.1.7 IMA – Denuncie

O IMA – Denuncie é um aplicativo desenvolvido pelo Instituto do Meio Ambiente de Alagoas (IMA-AL), com a finalidade de ajudar tal órgão na fiscalização de crimes ambientais [75]. Foi criado para permitir que o cidadão possa realizar denúncias de crimes ambientais de forma fácil, ágil e totalmente anônima [76]. Entretanto, é um aplicativo que não apresenta ao usuário o mapa dos crimes ambientais notificados, é apenas uma ferramenta de coleta colaborativa de dados para o IMA-AL. As categorias de crimes estão divididas em: poluição ou degradação do ambiente, infrações contra a flora e infrações contra a fauna.

3.1.8 Reporting Crimes

O *Reporting Crimes* é um site que possibilita aos usuários denunciar crimes ou violência que testemunharam ou que sofreram, com o objetivo de ajudar outras pessoas a se prevenirem e não serem vítimas na região da denúncia. A aplicação busca mapear as regiões perigosas e apresentar o resultado no “mapa da violência”, visando divulgar o índice de criminalidade de cada região. O site auxilia a suprir a cifra oculta de crimes que não são registrados perante as autoridades policiais por motivos diversos [77].

A aplicação permite o registro e a visualização dos seguintes tipos de delitos: comércio e/ou uso de drogas; roubo; roubo tentado; arrastão; arrombamentos diversos (veicular, de residência, a estabelecimentos comerciais); sequestro; lesão corporal; furto; furto tentado; depredações; e outros. E permite a apresentação das estatística relacionadas a cada tipo de delito por meio de gráficos ou zonas quentes na região escolhida [41].

3.2 Análise das Aplicações

Passos e Holanda [41] selecionaram para sua análise somente aplicações gratuitas e que estavam disponíveis para uso no Brasil. A análise consistiu em verificar para quais plataformas as aplicações estavam disponíveis, quais as funcionalidades mais comuns, se realizavam alguma verificação de confiabilidade das denúncias, se apresentavam estatísticas e o mapa de desordens/crimes, se possuía algum tipo de tutorial que ensinasse a usar a ferramenta, se havia a presença de dicas de segurança e alertas sobre áreas de perigo, dentre outros.

O estudo verificou inicialmente para quais plataformas as aplicações estavam disponíveis e apresentou o resultado conforme pode ser visto na Tabela 3.1. Neste ponto os autores destacam que a aplicação B.O. Coletivo, apesar de estar disponível na plataforma *Android* e de ser instalada corretamente, apresentou falhas na conexão com seu servidor, inviabilizando a análise do aplicativo quanto as demais fases do estudo.

Tabela 3.1: Plataforma das Aplicações (adaptado) (Fonte: [41]).

Aplicação	Plataforma		
	WEB	iOS	Android
WikeCrimes	✓		✓
QAP mobile		✓	✓
Unb Alerta	✓		✓
IMA Denuncie		✓	✓
B.O Coletivo		✓	
Eu fui roubado	✓	✓	✓
OndeFuiRoubado	✓	✓	✓
Reporting Crimes	✓		

Quanto a análise das funcionalidades mais comuns a cada uma das aplicações, verificou-se que todas as aplicações apresentavam uma interface com usuário amigável e de fácil operação. Sendo que a maioria delas possuíam algum tipo de filtro que permitia a seleção da região, ou o período de tempo, ou o tipo de delito/ocorrência que se desejava visualizar no mapa. No entanto, observou-se que nenhuma das aplicações permitia a delimitação de uma área específica, selecionada manualmente pelo usuário, para a análise de ocorrências. Observou-se ainda que as opções de seleções de regiões eram disponibilizadas apenas por cidades.

Para realizar a classificação do registro de denúncia de cada aplicação como simplificado, o estudo observou os seguintes quesitos: se a quantidade de passos necessários para se realizar o registro era menor ou igual a cinco; se os tipos de delitos disponíveis para registro eram intuitivos e não exigia do usuários grandes conhecimentos da legislação penal para classificar corretamente o tipo de delito; se a inserção do endereço do local da denuncia era possibilitada por meio de indicação no mapa da aplicação, ou por meio de coordenadas GPS ou por meio de código postal.

As principais funcionalidades encontradas foram listadas conforme apresentado na Tabela 3.2.

Quanto ao controle de qualidade ou de credibilidade dos registros inseridos nas aplicações, os autores analisaram: primeiro, se as aplicações exigiam algum tipo de cadastro do usuário para realizar um registro; e segundo, se possuíam alguma política para evitar a inserção de dados falsos ou denúncias de delitos que não ocorreram. No que diz respeito

Tabela 3.2: Principais Funcionalidades das Aplicações (adaptado) (Fonte: [41]).

Aplicação	Apresenta Filtros de Seleção	Estatísticas	Área de alerta	O registro simplificado
WikeCrimes	✓	✓		✓
QAP mobile	✓	✓	✓	✓
Unb Alerta		✓		✓
IMA Denuncie				✓
Eu fui roubado	✓	✓	✓	✓
OndeFuiRoubado	✓	✓		✓
Reporting Crimes	✓	✓		✓

ao primeiro quesito, verificaram que a maioria das aplicações analisadas exigiam um cadastro mínimo para que cada usuário pudesse fazer um registro no sistema, tendo como exceções os aplicativos IMA Denuncie e do *QAP mobile*, pois o foco destes aplicativos era justamente fomentar o registro de denúncias anônimas. Para responder ao segundo quesito, os autores do artigo verificaram se as aplicações estabeleciam algum nível de confiança ou credibilidade aos usuários ou as suas denúncias, se o usuário podia contestar denúncias feitas por outro usuário que seja sabidamente falsa, se o usuário podia confirmar uma denúncia feita por outro usuário, ou se havia algum outro meio de validação das denúncias. Nesse sentido, verificaram que apenas o “*WikeCrimes*” e o “Eu fui roubado” possuíam uma política para aprovação de registro ou de controle de credibilidade do usuário e do registro. Em especial, verificaram que o “*WikeCrimes*” estabelecia níveis de confiança para cada usuário, de forma que, se o registro fosse feito ou fornecido por um órgão oficial de segurança pública, ele tinha maior nível de confiança, e se o registro fosse feito por um usuário do povo e confirmado por outros cidadãos, ele também ganhava maior nível de confiança. Além disso, observou-se que o *WikeCrimes* tinha uma política interna para identificar formações de tendências maliciosas ou fraudulentas feita por grupo de pessoas.

Com exceção do IMA Denuncie, que é uma aplicação apenas para coleta de dados e não apresenta ao usuário os crimes ambientais em um mapa, as demais aplicações apresentavam um mapa para que o usuário pudesse visualizar os crimes registrados e que pudesse verificar quais as regiões com maior ou menor concentração de ocorrências. Nesse mapa, o estudo observou que os crimes eram apresentados de duas formas, ou na forma de pontos/alfinetes, ou por meio de manchas de calor, conforme exposto na Tabela 3.4.

Outros pontos analisados pelos autores nas aplicações foram: a presença de dicas de segurança para os usuários, a presença de alertas de áreas de risco e a existência de tutorial ou algum tipo de passo a passo ensinando como usar a ferramenta (Tabela 3.5). Como resultado, constataram que as aplicações “Eu fui roubado”, “OndeFuiRoubado” e

Tabela 3.3: Controle de Credibilidade de Registro (adaptado) (Fonte: [41]).

Aplicação	Controle de confiabilidade do registro	Identificação do usuário p/ inserir registro
WikeCrimes		✓
QAP mobile		
Unb Alerta		✓
IMA Denuncie	✓	
Eu fui roubado	✓	✓
OndeFuiRoubado		✓
Reporting Crimes		✓

Tabela 3.4: Apresentação dos Crimes no Mapa (adaptado) (Fonte: [41]).

Aplicação	Manchas de Calor	Pontos/Alfinetes
WikeCrimes	✓	
QAP mobile		✓
Unb Alerta	✓	✓
IMA Denuncie		
Eu fui roubado		✓
OndeFuiRoubado		✓
Reporting Crimes	✓	✓

“UnB Alerta” possuem dicas de segurança que podem ser consultadas dentro do próprio aplicativo, enquanto as demais não apresentavam essa funcionalidade. Que as aplicações “Eu fui roubado”, “OndeFuiRoubado” e “QAP mobile” permitiam a emissão de alerta de segurança ao usuário quando este se aproximava de áreas de risco. E que apenas o “Eu Fui Roubado” e o “QAP Mobile” possuíam tutorial presente na própria aplicação ou em sua página na internet.

Tabela 3.5: Outras Funcionalidades (adaptado) (Fonte: [41]).

Aplicação	Possui tutorial	Dicas de Segurança	Alertas em áreas de risco
WikeCrimes			
QAP mobile	✓		✓
Unb Alerta		✓	
IMA Denuncie			
Eu fui roubado	✓	✓	✓
OndeFuiRoubado		✓	✓
Reporting Crimes			

3.3 Discussão Sobre as Ferramentas

Passos e Holanda [41] apontam que todas as aplicações analisadas são de fácil manuseio, que a realização de uma denúncia nessas ferramentas são relativamente simples e que a interface das aplicações no geral são amigáveis. Relatam que:

“Algumas aplicações apresentaram funcionalidades muito interessantes, tais como: as “Dicas de Segurança”, que orientam o usuário como se prevenir contra delitos; os “Alertas em Áreas de Risco”, que avisam o usuário quando ele se aproxima ou entra em uma região como alto índice de criminalidade; e o estabelecimento de níveis de confiança de uma denúncia, que dá maior credibilidade ao sistema”.

No entanto, os autores expõem que em todas as aplicações analisadas, a lista de infrações disponibilizada para a realização do registro de uma denúncia era limitada, pois continha apenas uma pequena parcela das infrações contempladas pelo regramento penal brasileiro. Além disso, relatam que as infrações listadas, no geral, não possuíam uma explicação mínima que orientasse o usuário. O que é apontado como um risco, pois se um usuário não conhece a tipificação penal de cada delito, pode registrá-lo em uma tipificação errada.

Outra consideração que os autores julgam importante fazer é que nenhuma das aplicações analisadas, foram criadas ou patrocinadas pelos órgãos de segurança pública brasileiros. O que gera certa limitação destas aplicações estudadas, uma vez que apresentam poucos ou nenhum registro oficial de crime e desordem.

Não obstante, o estudo aponta que apesar da tímida participação dos órgãos de segurança pública na produção de ferramentas que realizem mapeamento colaborativo de crimes e desordens, existem algumas universidades públicas que têm desenvolvido pesquisas na área. A exemplo disso, indicam as aplicações “*WikeCrimes*” e “UnB Alerta” que foram desenvolvidas e mantidas pelas universidades públicas Universidade de Fortaleza (Unifor) e Universidade de Brasília (UnB), respectivamente.

Os autores concluem o estudo relatando que no geral as aplicações analisadas realizam o mapeamento de crime ou contravenções penais, mas que há pouca ênfase no mapeamento de desordens sociais. Que praticamente não há a participação dos órgãos de segurança pública brasileiros nesse processo, seja como idealizadores, colaboradores ou patrocinadores dessas aplicações. E que no Brasil ainda há a carência de uma aplicação que reúna as melhores funcionalidades de cada uma das aplicações analisadas, que consiga estimular a participação dos órgãos públicos brasileiros e que além de realizar o mapeamento criminal, também realize o mapeamento de desordens sociais.

Nesse contexto, a implementação do sistema Tô de Olho no DF pode ser de grande ajuda para aproximar o cidadão do Estado, permitindo que participe ativamente da

melhoria da sua comunidade, aumentando a capilaridade da coleta de dados de desordens e melhorando a planejamento de segurança pública com vistas a prevenir o crime.

Capítulo 4

Arquitetura Proposta

Neste capítulo é apresentada a arquitetura do “Sistema Tô de Olho no DF” e está subdividido nas seguintes seções: Seção 4.1 Cenário; Seção 4.2 Arquitetura do Sistema; Seção 4.3 Interface Web do Tô de Olho no DF; e Seção 4.4 Ambiente Utilizado para Disponibilizar os Serviços do Tô de Olho no DF.

4.1 Cenário

A Secretaria de Estado da Segurança Pública e da Paz Social do Distrito Federal (DF) criou, no ano de 2015, o programa “Viva Brasília - Nosso Pacto pela Vida” que é uma estratégia para potencializar a capacidade de combate e prevenção da violência, por meio da integração entre os órgãos de segurança pública do DF e demais atores sociais, utilizando-se de metodologias de resolução de problemas de criminalidade e realizando ações preventivas e de promoção da cultura da paz [78].

O documento orientador do “Viva Brasília – Nosso Pacto pela Vida O Pacto pela Vida” [79] expõe que:

“O Pacto pela Vida representa um modelo de gestão em segurança pública amparado na perspectiva de que a situação de vulnerabilidades a violências, desordens, incivildades e crimes requer do Estado a consolidação de estratégias de intervenção articuladas entre o setor policial e outros diferentes setores.”

Então, com vista a atender esse modelo de gestão, a Subsecretaria de Gestão da Informação (SGI) adotou, como uma das frentes do programa, o mapeamento de desordens urbana, onde os relatos de desordens são obtidos por meio de participação social dos Conselhos Comunitários de Segurança Pública (Consegs) ou por meio dos articuladores territoriais, agentes contratados pelo Estado [5]. Essas informações de desordem são repassadas aos órgãos responsáveis para providências cabíveis ou, a depender do caso, é

encaminhado ao Comitê Gestor do Pacto pela Vida, que é presidido pelo governador do DF.

Parte desse processo de mapeamento é realizado por meio da ferramenta de TI chamada Sistema Integrado de Geoestatística (SIGEO), que tem por finalidade viabilizar a identificação de condições que impactam na sensação de segurança e no crime, de forma que a partir da visualização dos fenômenos apresentados, possam buscar a solução para os problemas [80].

No entanto, existe uma parte inicial desse processo, a coleta dos dados de desordem, que é realizada de forma manual, por meio do preenchimento de formulários em papel, de forma que somente em um segundo momento, tais dados são inseridos no SIGEO.

Nesse contexto, a FAP-DF, abriu o Edital 07/2017, onde disponibilizou recursos para algumas linhas de pesquisa na área de Segurança Pública no Distrito Federal e RIDE-DF, sendo que uma dessas linhas era a pesquisa e desenvolvimento de tecnologias de aproximação entre a população e os órgãos de segurança pública do Distrito Federal.

Portanto, a arquitetura proposta, no presente trabalho, busca viabilizar a coleta de dados fornecidos de maneira voluntária por qualquer cidadão que tenha acesso a um *smartphone*, assim como também, apresentar esses dados, em tempo real, aos gestores públicos, facilitando a realização de análise criminal ou de desordem urbana. A arquitetura tem a finalidade de oferecer uma interface amigável aos usuários, permitir o armazenamento dos dados fornecidos pelos articuladores territoriais e pelos cidadãos, além de permitir aos gestores públicos a visualização e o acompanhamento dos relatos de desordem.

O sistema foi desenvolvido para ser utilizado por três tipos de perfis de usuários:

- Colaborador: é o articulador territorial da SGI ou qualquer pessoa do povo que seja voluntária. Eles podem cadastrar informações sobre desordem urbana, por meio de seu *smartphone*. Essas informações prestadas são associadas a uma coordenada geográfica e são enviadas por meio *web service* para armazenamento em um banco de dados central;
- Gestor: é o agente público responsável pela tomada de decisão. Ele visualiza as informações de desordem armazenadas no banco de dados, por meio da interface do sistema de informação geográfico *web*, de forma que possa identificar locais e eventos geradores de crimes e de desordens maiores. A interface fornecida ao gestor apresenta um mapa contendo a plotagem das pontos de desordens salvos no banco, o *status* da desordem e seus indicadores;
- Técnico: é o agente público responsável pela manutenção e administração do sistema. Ele interage com o sistema por meio da interface *web* e é o responsável por cadastrar órgãos públicos e seus usuários no sistema, por atribuir as desordens que

são responsabilidade de cada órgão e por verificar, acompanhar e alterar o *status* de cada desordem (com problema, em resolução, resolvido) conforme necessário.

O ambiente de funcionamento do Sistema Tô de Olho no DF é apresentado conforme a Figura 4.1. Cabe ressaltar, que apesar de não estar expresso na figura, todos os usuários do sistema realizam requisições e recebem respostas do servidor via protocolo HTTP, conforme é explicado mais adiante.

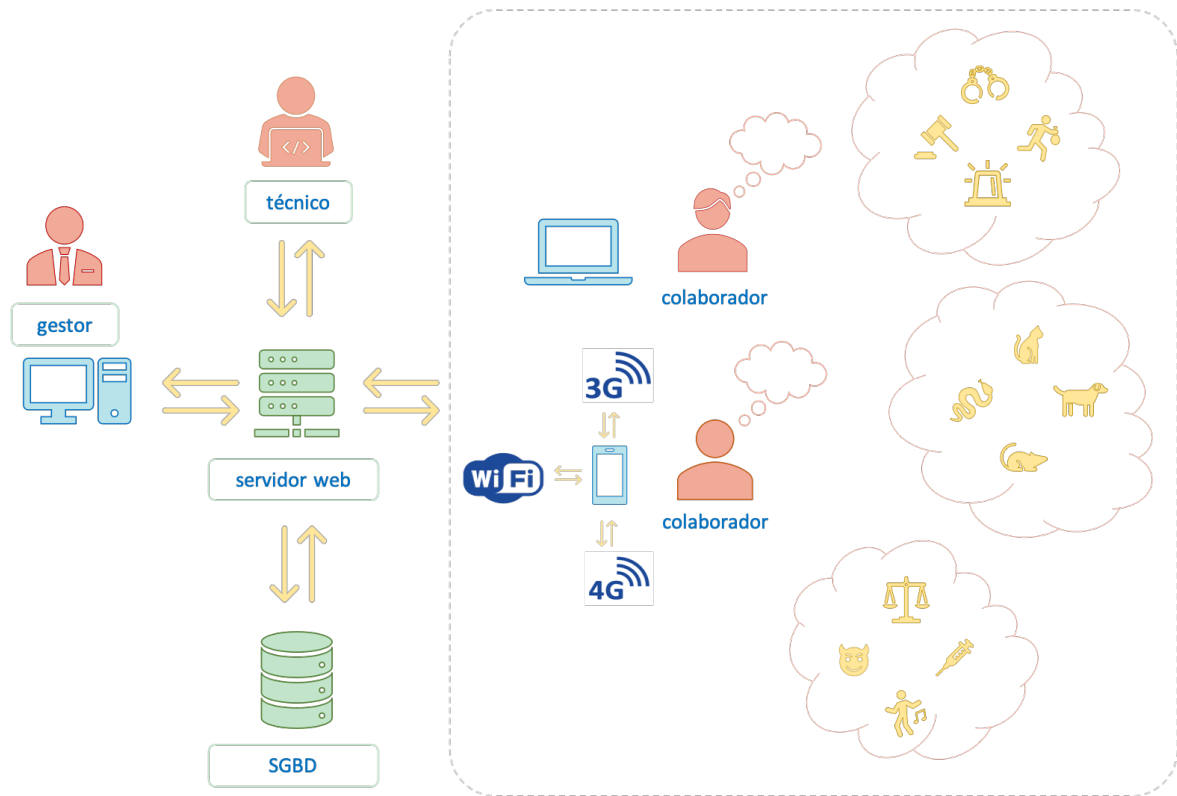


Figura 4.1: Funcionamento do Sistema Tô de Olho.

4.2 Arquitetura do Sistema

A arquitetura proposta para o Sistema Tô de Olho se baseia no estilo de arquitetura REST, é uma arquitetura híbrida para sistemas de hipermídia distribuídos, se utiliza do modelo de comunicação cliente-servidor, seus componentes possuem interface uniforme, a comunicação é *stateless*, as respostas do servidor são sinalizadas com *cacheable* ou *non-cacheable* e é implementada em três camadas: Camada de Apresentação, Camada de Aplicação e Camada de Dados (Figura 4.2). Ressalta-se que a comunicação entre as camadas é viabilizada por meio de uma API REST, onde as requisições e respostas se dão por meio do protocolo HTTP, conforme apresentado na Figura 4.2.

4.2.1 Modelo em Três Camadas

O modelo em três camadas é um modelo do tipo cliente-servidor, na qual a lógica, o acesso aos dados e a interface com o usuário (apresentação) são desenvolvidos e mantidos como módulos independentes. Ele é um padrão de *design* de *software* bem estabelecido e suas três camadas são [81]:

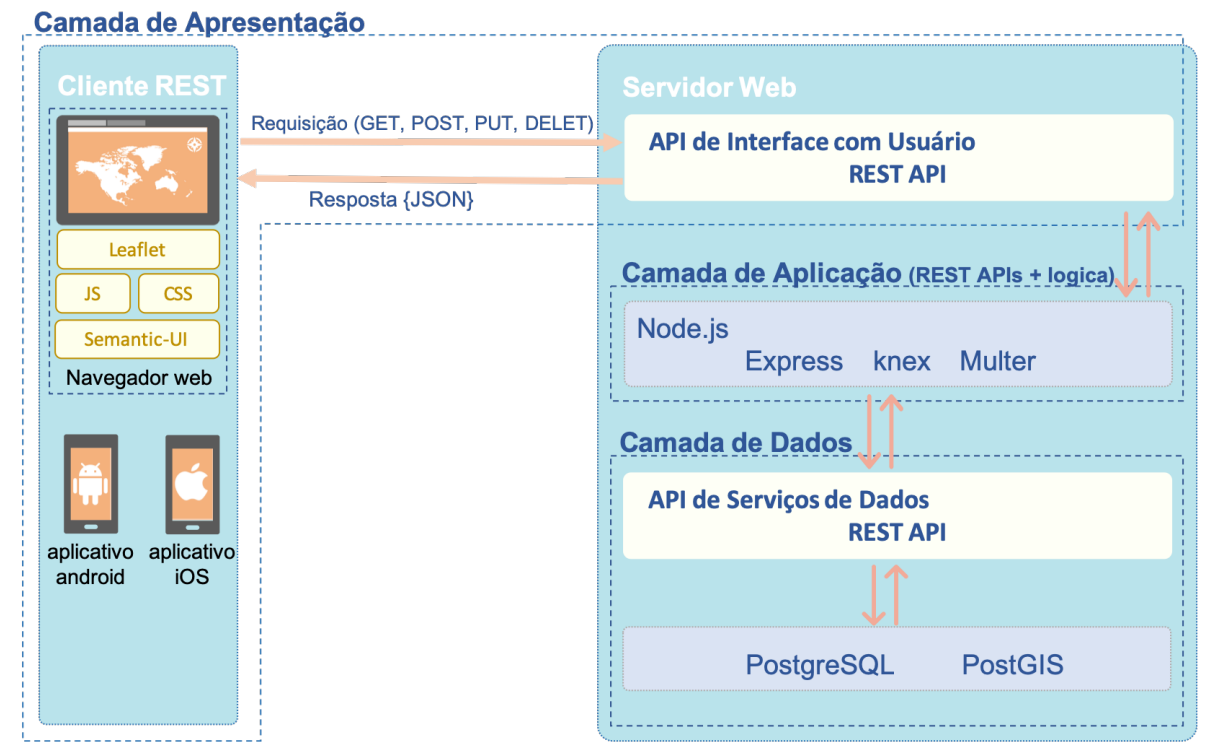


Figura 4.2: Arquitetura do Sistema Tô de Olho.

1. Camada de apresentação: ocupa o nível mais alto e é responsável pela interação com o usuário, trabalha por meio de *inputs* e *outputs*. comumente esta camada exibi as informações relacionadas aos serviços disponíveis no sistema;
2. Camada de aplicação: ocupa o nível intermediário e é responsável pela lógica do negócio. Nesta camada estão implementadas as validações necessárias aos *inputs* vindos da camada de apresentação para a obtenção dos dados armazenados na camada de serviço de dados;
3. Camada de dados: ocupa o nível mais baixo e é responsável pelo armazenamento dos dados. Normalmente as instruções utilizadas para comunicação nessa camada são instruções SQL.

Esse tipo de modelo apresenta as seguintes vantagens: permitir a manutenção do código em separado ou mesmo permitir a substituição de uma certa tecnologia em uma camada sem grande impacto nas camadas adjacentes; permitir que diferentes equipes de desenvolvimento trabalhem em suas próprias áreas de especialização; permitir escalar a aplicação por meio de vários servidores da web; oferecer confiabilidade e independência entre os servidores ou os serviços subjacentes [82].

Nas próximas seções são apresentados os detalhes da implementação e a estrutura de cada uma dessas camadas do Sistema Tô de Olho (Figura 4.2).

4.2.1.1 Camada de Apresentação

A camada de apresentação do Tô de Olho no DF é composta pelos aplicativos *mobile* (Android e iOS) e pela parte *web* (Figura 4.3).

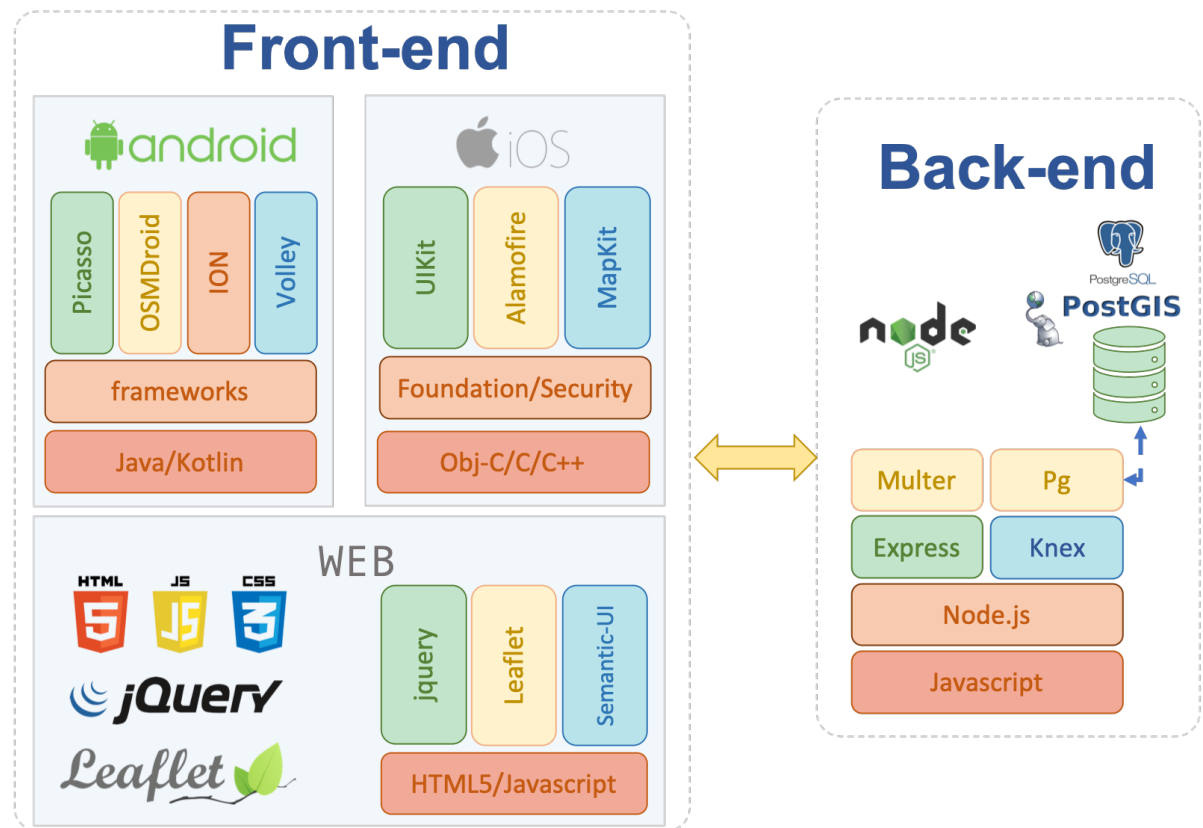


Figura 4.3: Front-End e Back-End do Sistema Tô de Olho.

Para desenvolver a parte *web* da camada de apresentação, utilizou-se o OpenStreetMap (OSM), o *framework* Semantic-UI e as bibliotecas em JavaScript Leaflet e Jquery. O OpenStreetMap é um projeto de mapeamento voluntário que tem a finalidade de mapear estradas, ruas, estações ferroviárias, trilhos, estabelecimentos comerciais, dentre outros,

de forma livre e editável, e é disponibilizado sob a licença *Open Data Commons Open Database License* (ODbL) pela *Fundation OpenStreetMap* (OSMF) [83]. O Semantic-UI é um *framework* de desenvolvimento *front-end*¹ que auxilia na elaborar de *layouts* responsivos por meio de HTML e é focado em chegar ao resultado final de forma progressiva, simples e fácil [84]. A JQuery é uma das biblioteca de funções JavaScript mais populares, projetada pela Fundação JS, ela manipula documentos HTML, manipula eventos e realiza animações, e busca ser rápida, pequena, simples e rica em recursos [85]. A Leaflet é outra biblioteca de funções JavaScript, concebida para criar aplicações de mapeamento na *web* e para suportar HTML5 e CSS3, por meio dela foi possível viabilizar a exibição das denúncias no mapa na forma de alfinetes e na forma de mapas de calor, que é uma representação gráfica de dados em que os valores individuais contidos em uma matriz são apresentados com base em cores quentes [86].

A parte mobile da camada de apresentação, como já dito, é composta de um aplicativo para Android e um aplicativo para iOS. Sendo que o aplicativo para Android já foi apresentado no Capítulo 3 do trabalho “ToDeOlho para Android: Um Software Para Mapeamento Colaborativo de Desordens Sociais por Meio de Sistemas Georreferenciados” [87]. E o aplicativo iOS é apresentado no Capítulo 5 do presente trabalho.

4.2.1.2 Camada de Aplicação

A camada de aplicação está do lado do servidor e é constituída do módulo de comunicação. Este detém as regras de negócio e foi implementado na forma de um RESTful *web service* [87]. Então, recebe as requisições dos clientes por meio da API REST e providencia o envio das respostas. Para responder a essas requisições, pode fazer requisições a camada de persistência por meio da API de serviço de dados (ver Figura 4.2). A implementação dessa camada foi desenvolvida utilizando o Node.js, algumas de suas bibliotecas, o *framework* Express, o Knex e o Multer.

O Node.js é um interpretador de código JavaScript para ser executado do lado do servidor, foi criado em 2009 por Ryan Dahl e outros colaboradores [88]. É de código aberto, possui uma comunidade ativa, é altamente escalável e permite ao programador desenvolver com diversos protocolos de rede e internet, e possui bibliotecas para acesso aos recursos do sistema operacional. Tem como principais características: suas aplicações serem *single-thread*, isso é, cada aplicação tem uma *thread* principal por processo; e ser orientado a eventos, de forma que seu *Event-Loop* fica em um *loop* infinito escutando e emitindo eventos [89].

¹Front-end é o termo usado para se referir a interface de interação com o usuário.

O Express é um *framework* para Node.js que foi criado com inspiração no *framework* Sinatra do Ruby. É livre e de código aberto, distribuído sob a MIT Licence². O Express viabiliza a construção de APIs e a criação de aplicações *web*. Segundo Pereira [89] seu foco é trabalhar com perfeição manipulando **views**, **routes** e **controlles**, ficando à escolha do desenvolvedor trabalhar com **models** e usar outro *framework* de persistência. Suas principais características são [89] [87]:

- Código minimalista;
- Possui *Routing* robusto;
- É capaz de gerenciar diversas requisições simultaneamente;
- Trabalha com conceito de *middlewares*;
- Adota padrões e boas práticas de serviços REST;
- Permite a definição de configurações da aplicação *web*, tais como: a porta a ser utilizada e a estruturação da URLs;
- É de fácil integração com os principais *Template Engines*.

No desenvolvimento do sistema “Tô de Olho no DF” o Express auxiliou na configuração de todas as rotas, e permitiu que fosse definido um tratamento e uma resposta diferente para cada uma delas, sendo todas as respostas estruturadas em formato *JavaScript Object Notation* (JSON). E o *middleware*³ *Multer* serviu para viabilizar o *upload* de arquivos dos clientes para o servidor. Optou-se por utilizar esse *middleware* para Express no desenvolvimento do sistema, pelo fato dele facilitar o manuseio de *multipart/form-data*, que é a codificação normalmente usada para envio de *upload* de um arquivo de um cliente *web* para o servidor [90].

O Knex ou Knex.js, segundo sua página oficial [91] é:

“Um construtor de consultas (*queries*) SQL para Postgres, MSSQL, MySQL, MariaDB, SQLite3, Oracle e Amazon Redshift, projetado para ser flexível, portátil e divertido de usar. Ele apresenta chamadas de retorno no estilo tradicional de nó e uma interface *promise* para controle de fluxo assíncrono mais limpo, uma interface de fluxo, construtores de consulta e esquema com recursos completos, suporte a transações (com *savepoints*), pool de conexão e respostas padronizadas entre diferentes consultas de clientes consulta.”

No “Tô de Olho no DF”, ele foi utilizado para construir as consultas ao banco de dados de forma mais fácil e prática.

²MIT Licence é uma licença de software criada pelo Massachusetts Institute of Technology (MIT). Ela é uma licença permissiva utilizada tanto em software livre quanto em software proprietário.

³Os *middlewares* são funções que podem tratar os requisições e as respostas das rotas antes e/ou depois que uma rota é processada.

4.2.1.3 Camada de Dados

A camada de dados é responsável pelo armazenamento de dados convencionais e por dados espaciais. Sendo os dados convencionais o conjunto de informações que já são bem conhecidos nos bancos de dados relacionais, tais como, **string**, **integer**, **float**, dentre outros; e os dados espaciais aquele conjunto de informações referente às classes geométricas, tais como, ponto, linha polígono.

Na arquitetura proposta para o presente trabalho, a camada de dados é composta por um SGBD PostgreSQL com a extensão espacial PostGIS, ambos já citados e explicados no Capítulo 2.

Modelo de Dados

O modelo de dados adotado para o "Tô de Olho" foi o Modelo Relacional (MR), que é um modelo de dados que tem como regra o arranjo dos dados em tabelas. Sendo cada tabela um conjunto não ordenado de linhas, em que cada campo de cada linha armazena o valor de um atributo; que cada coluna é a representação desses atributos; e que a cardinalidade é a correlação de instâncias dessas tabelas nos relacionamentos [92]. O MR do sistema "Tô de Olho" é apresentado na Figura 4.4, seus relacionamentos são apresentados conforme a Tabela 4.1 e a descrição de suas entidades são as seguintes:

- *usuario*: Armazena as informações referentes aos usuários do sistema;
- *tipo_usuario*: Armazena os tipos de usuários do sistema;
- *denuncia*: Armazena as informações referentes às denúncias de desordens informadas pelos usuários;
- *desordem*: Armazena a natureza das desordens, classificadas pela Secretaria de Segurança Pública;
- *tipo_desordem*: Armazena os tipos de desordens a serem mapeadas;
- *org_orgao*: Armazena a relação de órgãos públicos e privados que fazem parte do sistema;
- *video*: Armazena informações que relacionam um arquivo carregado no formato de vídeo a uma dada denúncia;
- *imagem*: Armazena informações que relacionam um arquivo carregado no formato de imagem a uma dada denúncia;
- *foto_usuario*: Armazena informações que relacionam um arquivo carregado no formato de imagem a um dado usuário;

- *regioes_administrativas*: Armazena informações relativas ao perímetro ocupado por cada região administrativa do DF;
- *regiao_alerta*: Armazena informações de regiões de alertas relacionadas a um dado usuário; e
- *confirmacao*: Armazena informações relacionadas ao nível de confiabilidade de uma denúncia realizada por um determinado usuário.

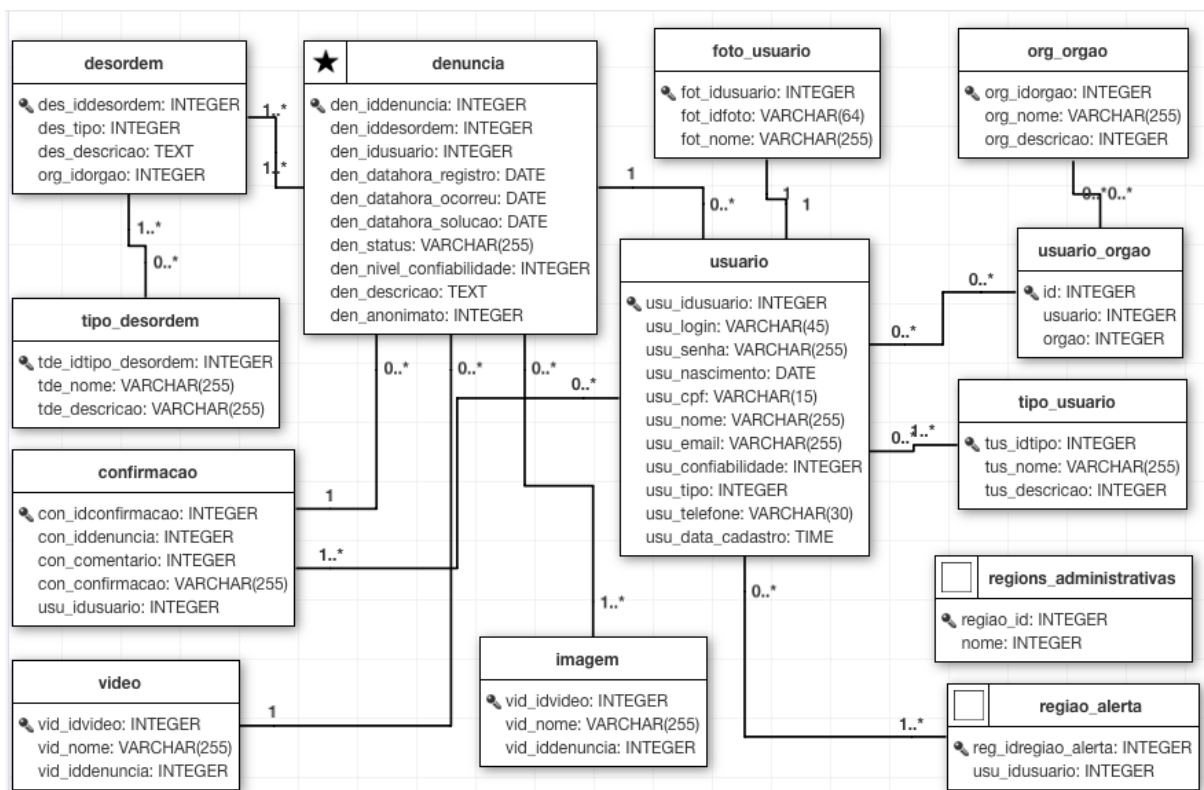


Figura 4.4: Modelo OMT-G do Tô de Olho.

Tabela 4.1: Tabela de Relacionamento do Modelo de Dados.

Relacionamento	Detalhamento do Relacionamento
usuario e tipo_usuario	Cada usuário pertence à apenas um tipo de usuário
usuario e regioao_alerta	Cada usuário pode ter de 0 a n regiões de alerta
usuario e foto_usuario	Cada usuário pode ter de 0 a n fotos do perfil
usuario e confirmacao	Cada usuário pode confirmar de 0 a n denúncias de outros usuários
usuario_orgao e org_orgao	Cada usuário pode pertencer a n órgãos
usuario_orgao e usuario	Cada órgão possui de 0 a n usuários
usuario e denuncia	Cada usuário pode registrar n denúncias
tipo_desordem e desordem	Cada tipo de desordem possui de 0 a n naturezas de desordens
denuncia e desordem	Cada denuncia pertence a uma natureza de desordem
denuncia e imagem	Cada denuncia pode possuir de 0 a n imagens relacionadas
denuncia e video	Cada denuncia pode possuir de 0 a n vídeos relacionados
denuncia e confirmacao	Cada denuncia pode receber n confirmações

4.3 Interface Web do Tô de Olho no DF

Nesta seção são apresentados os aspectos da interface *web* desenvolvida para o sistema Tô de Olho e suas funcionalidades.

A Figura 4.5 mostra a página inicial do sistema, que é apresentada a qualquer usuário que não está logado no sistema. Nesta página, as denúncias são apresentadas na forma de alfinetes ou em círculos, conforme o zoom do mapa. No interior do círculo, há o quantitativo de denúncias daquela região. As denúncias são apresentadas sobre o mapa do OpenStreetMap que pode ser expandido desde o nível do mapa *mundi* até nível de ruas. Se o usuário clicar em uma dessas denúncias no mapa, poderá ver detalhes como: a descrição da denúncia, seu nível de confiabilidade, a natureza da desordem, a data e hora que ocorreu e o usuário que realizou a denúncia, caso não seja anônima (Figura 4.6).

No menu lateral da página inicial, o usuário pode selecionar alguns filtros que lhe permite apresentar as denúncias conforme seu *status*, a região administrativa onde se encontra, ou na forma de mapa de calor (Figura 4.7). No entanto, para registrar uma denúncia o usuário deve fazer *login* no sistema.

A Figura 4.8 mostra a Área do Gestor, que pode ser acessada por usuários cadastrados no sistema com perfil de gestor. Observa-se que ao realizar o *login* com esse tipo de perfil,

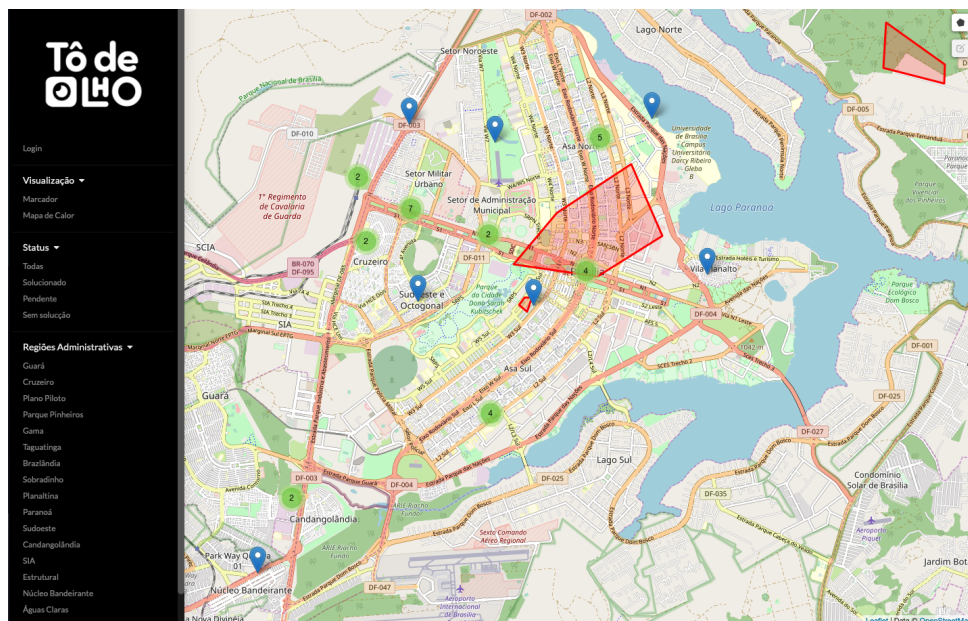


Figura 4.5: Página inicial do sistema TÔ de Olho.



Figura 4.6: Detalhes da denúncia.

será apresentado no menu lateral, a opção para acesso ao Painel de Gestor. Nesse painel o gestor pode visualizar: as denúncias cadastradas no sistema, por meio da lista de denúncias ou por meio da visualização no mapa; o quantitativo de denúncias por órgão públicos; e o gráfico de evolução do quantitativo de denúncias em cada mês do ano (Figura 4.9).

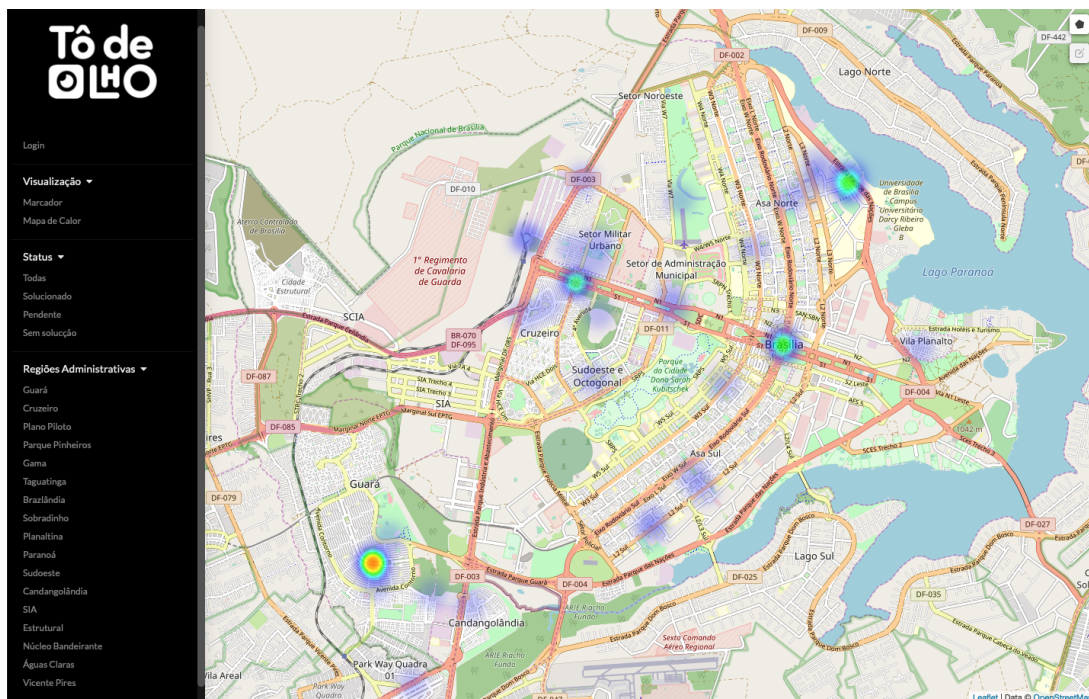


Figura 4.7: Apresentação das denúncias em forma de mapa de calor.

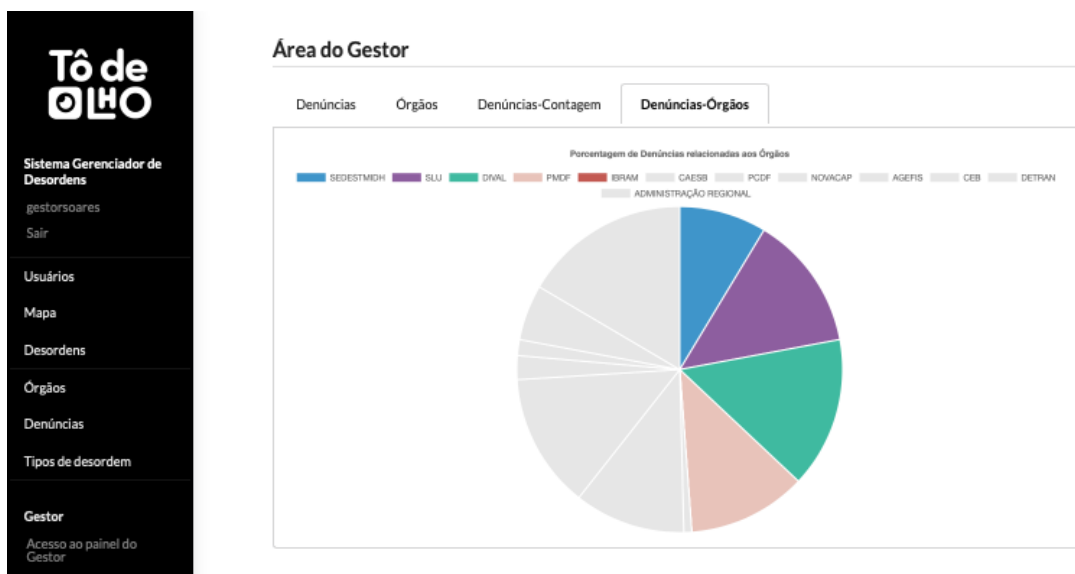


Figura 4.8: Apresentação do número de denúncias por órgão responsáveis .

4.4 Ambiente Utilizado para Disponibilizar os Serviços do Tô de Olho no DF

Para disponibilizar os serviços do Sistema Tô de Olho no DF e sua interface Web na internet, durante o período de desenvolvimento e testes do projeto, utilizou-se as

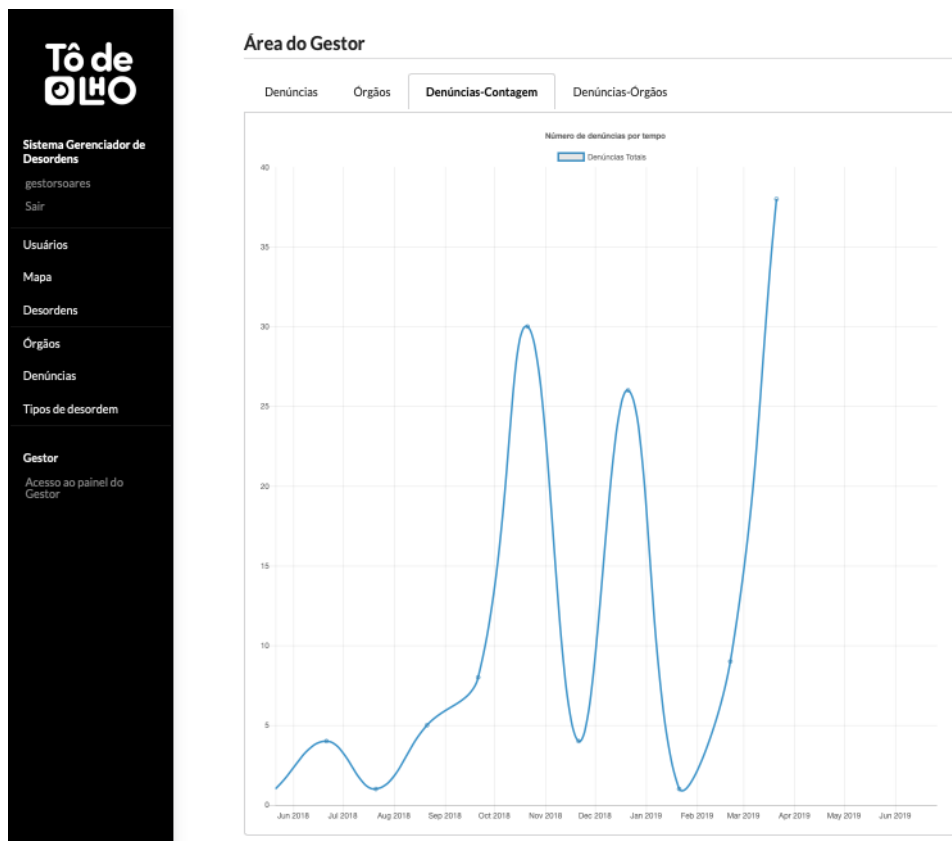


Figura 4.9: Apresentação do número de denúncias por mês do ano.

plataformas Heroku e Amazon Web Services.

4.4.1 Heroku

O Heroku é uma solução de plataforma como serviço em nuvem (*cloud platform as a service* - PaaS) que suporta várias linguagens de programação e permite que desenvolvedores criem, forneçam, monitorem e dimensionem aplicativos sem ter que se preocupar com a implantação, configuração, dimensionamento, ajuste e gerenciamento das aplicações no servidor [93]. Ele permite ao desenvolvedor que não se preocupe com detalhes de infraestrutura, pois oferece *containers* para instalação das aplicações, facilitando a manutenção, a extensão e a escalabilidade das aplicações.

Essa plataforma permitiu colocar em produção, na nuvem, o módulo do servidor e a parte *web* do sistema To de Olho no DF que eram fundamentais para a troca de informações e armazenamento dos dados coletados por meio do aplicativo. Sua escolha se deu pelo fato de que inicialmente precisava-se disponibilizar o módulo do servidor na internet temporariamente, para realizar os testes de funcionalidades do sistema como um todo, incluindo os aplicativos, e por esta plataforma oferecer segurança e facilidade

Tô de Olho
Sistema Gerenciador de Desordens
gestorsoares
Sair

Usuários
Mapa
Desordens
Orgãos
Denúncias
Tipos de desordem

Gestor
Acesso ao painel do Gestor

Área do Gestor

ID	Tipo	Status	Descrição	Usuário
1	Prédio, casa ou galpão abandonado	Solucionado	Galpão abandonado perto do bloco A	cidadaosouza
2	Cheiros desagradáveis	Com problemas	Há um cheiro desagradável saindo do esgoto.	usuario@gmail.com
4	Uso de bebidas alcoólicas na rua	Com problemas	Existem pessoas constantemente se embriagando nesse local	usuario@gmail.com
5	Pessoas fazendo xixi ou cocô na rua	Com problemas	Nesse local as pessoas estão fazendo necessidades fisiológicas indiscriminadamente	usuario@gmail.com
6	Animais abandonados	Com problemas	No local a uma ninhada de gatos abandonada	usuario@gmail.com
26	Lixeiras quebradas	Com problemas	Há uma lixeira quebrada aqui	gestorsoares
27	Uso de drogas ilegais na rua	Com problemas	Existem alunos usando maconha nesse local	gestorsoares
28	Poste de luz danificado	Com problemas	Existe um poste de luz e a lâmpada quebrada	gestorsoares
14	Poste de luz danificado	Com problemas	Dois vândalos quebraram a lâmpada do poste desta rua	tecnicosilva
29	Infestação de parasitas	Com problemas	Muitos insetos presentes nessa região	gestorsoares
186	Área de lazer e esporte quebrados	Com problemas	Área de lazer e esporte quebrados	cunha.passos@gmail.com
148	Via pública com mato alto	Pendente	mato alto perto da parada de ônibus	renata.emerick
149	Prostituição	Com problemas		
131	Flanelinhas	Com problemas	teste	sobierajski

Figura 4.10: Lista de denúncias cadastradas no sistema..

nesse sentido, não comprometendo assim, o tempo desenvolvimento com outras atividades relacionadas à infraestrutura.

4.4.2 Amazon Web Services

A *Amazon Web Services* (AWS) é uma plataforma de serviços de computação em nuvem, disponibilizada pela Amazon.com, que provê serviços online para *websites* ou aplicações cliente-servidor baseados em nuvem. Ela oferece serviços tais como *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS) e *Packaged Software as a Service* (SaaS) [94].

A plataforma foi utilizada para armazenar o banco de dados do Sistema Tô de Olho, por meio do serviço *Amazon Simple Storage Service* (Amazon S3). Ela foi selecionada para esse fim, por ser de fácil integração com o Heroku, por fornecer recursos de gerenciamento fáceis de usar e por possuir boa escalabilidade, disponibilidade de dados, segurança e performance.

Capítulo 5

O Aplicativo ToDeOlho para iOS

Neste capítulo apresenta-se o aplicativo Tô de Olho para iOS, que é parte do projeto “Arquitetura para Gerenciamento de Dados Colaborativos para Secretaria de Estado da Segurança Pública e da Paz Social baseada em Sistema de Informação Geográfico com Participação Popular” apoiado pela Fundação de Apoio a Pesquisa do Distrito Federal (FAP-DF) (protocolo no 19781.82.31496.17082017). O aplicativo tem por finalidade coletar dados de desordens urbanas, por meio de colaboradores (agentes públicos ou pessoas voluntárias) e apresentá-los em um mapa por meio do OpenStreetMap.

5.1 Engenharia de Software

O Tô de Olho no DF foi desenvolvido utilizando-se metodóloga ágil, pois no geral permite a utilização de uma equipe pequena e o desenvolvimento do software de forma rápida e incremental [95]. Assim, realizou-se os processo de especificação, projeto e implementação de forma intercalada, seguindo os conceitos chaves do Manifesto Ágil [96]:

- Indivíduos e interações ao invés de processos e ferramentas;
- Software executável ao invés de documentação;
- Colaboração do cliente ao invés de negociação de contratos;
- Respostas rápidas a mudanças ao invés de seguir planos.

O ciclo de desenvolvimento do aplicativo se realizou em três fases [97]:

- Pré-planejamento: fase em que os requisitos foram descritos em um documento chamado *backlog*, em que foram feitas as priorizações e as estimativas de esforços para o desenvolvimento de cada requisito;

- Desenvolvimento: fase em que o aplicativo foi de fato implementado, respeitando as variáveis técnicas e do ambiente identificadas na fase anterior;
- Pós-planejamento: última fase do ciclo, que consistiu em realizar os testes funcionais e de usabilidade e em apresentar o aplicativo aos usuários.

5.2 Pré-planejamento

Para subsidiar o desenvolvimento do aplicativo Tô de Olho para iOS, foram analisadas outras ferramentas inseridas em contexto semelhante ao do Tô de Olho, análise esta, que é apresentada no Capítulo 3 do presente trabalho. E foi elaborado um cronograma de trabalho que incluía reuniões semanais com a equipe do projeto e reuniões com os *Stakeholders*, em especial com os servidores da Subsecretaria de Gestão da Informação (SGI) da SSP/DF. Sendo que estas reuniões tinham por finalidade realizar a análise de domínio e de tarefas da aplicação.

5.2.1 Análise de Domínio

A análise de domínio buscou identificar de forma genérica em qual contexto o aplicativo seria inserido, quais os agentes estariam envolvidos e quais os relacionamentos que seriam estabelecidos entre eles. Este estudo foi realizado com auxílio dos servidores da SGI-SSP/DF, a Tabela 5.1 apresenta um resumo dos resultados.

Tabela 5.1: Resultado da análise de domínio.

Contexto	Mapeamentos de desordens urbanas no Distrito Federal
Agentes Envolvidos	- Colaboradores: articuladores da SGI-SSP/DF, membros dos CONSEGs e pessoas voluntárias; - Agentes públicos dos órgãos envolvidos.
Relacionamentos	- Os colaboradores fornecem informações de desordem à SGI-SSP/DF; - Os agentes da SGI-SSP/DF fazem a análise dos dados e encaminham ao órgão responsável; - Os agentes do órgão responsável tomam as devidas providências e retornam o <i>feedback</i> à SGI-SSP/DF; - Os agentes da SGI-SSP/DF retornam o <i>feedback</i> aos colaboradores;

5.2.2 Análise de Tarefas

A análise de tarefa foi responsável por coletar informações sobre os procedimentos adotados pela SGI no processo para mapear uma desordem pública. O levantamento dessas informações foi realizado por meio de entrevistas com os servidores da SGI-SSP/DF, identificando-se as tarefas que cada agente realiza no processo. O resultado dessa análise pode ser resumido no modelo de processo apresentado na Figura 5.1.

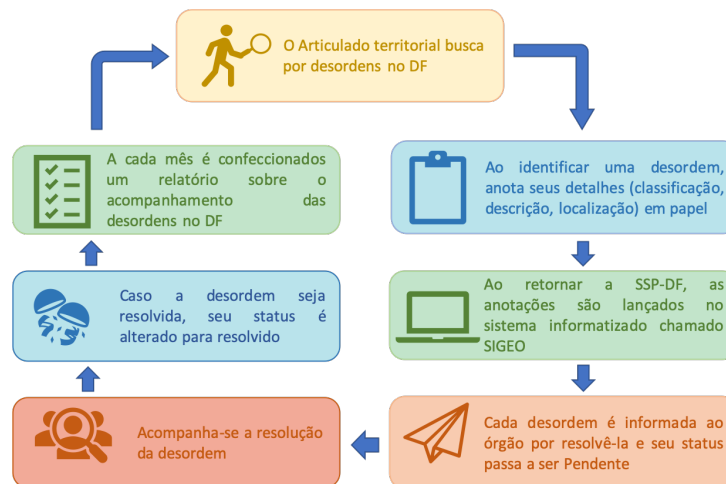


Figura 5.1: Análise de tarefas no mapeamento de desordens sociais feito pela SGI.

5.2.3 Requisitos

Segundo Staa [98], requisitos são funções condições, atributos, propriedades ou características que devem ser satisfeitas por um artefato de software. E os requisitos de software são um subconjunto dos requisitos do sistema e estes por sua vez, são a descrição do comportamento do sistema, do ponto de vista do negócio, do cliente, ou de outro sistema. Em outras palavras, os requisitos de software descrevem o que o software deve fazer. Nesse contexto, as análises de domínio, de tarefas e de usuários esclareceram quais eram os requisitos do aplicativo, sendo eles:

1. Cadastrar usuário;
2. Autorizar o acesso de usuários já cadastrados via *login*;
3. Identificar um usuário pelo endereço de e-mail e senha informados no momento do cadastro;
4. Identificar a posição geográfica do usuário por meio de coordenadas cartográficas;
5. Mostrar o mapa do OpenStreetMap ao usuário;

6. Exibir no mapa um conjunto de desordens urbanas já cadastradas na base de dados;
7. Fornecer a descrição de cada uma das desordens cadastradas na base de dados, a medida que cada desordem é selecionada;
8. Permitir que o usuário selecione no mapa um local e que, para este local, possa fazer um registro de denúncia de desordem urbana, contendo: a localização da desordem; a data e a hora que ocorreu a desordem; a natureza da desordem; e a descrição da desordem;
9. Permitir que o usuário possa incluir arquivos de imagens ao registro de denúncia;

Os requisitos citados acima são indispensáveis para a aceitação do aplicativo Tô de Olho no DF. Mais detalhes sobre o funcionamento do aplicativo e sua implementação serão tratados nas próximas seções.

5.3 Desenvolvimento

A implementação do Tô de Olho no DF para iOS é uma das três partes do *Front-End* do sistema do Tô de Olho no DF, conforme apresentado no Capítulo 4 (Figura 4.3).

5.3.1 Tecnologias Utilizadas para o Desenvolvimento do Aplicativo

Além das tecnologias já citadas anteriormente, especificamente para o desenvolvimento do aplicativo para iOS, utilizou-se a *Integrated Development Environment* (IDE) Xcode e algumas bibliotecas e *frameworks* da linguagem Swift, destacando-se os seguintes: UIKit, alamofire e MapKit.

O Xcode é um ambiente de desenvolvimento integrado da Apple Inc que é usado para desenvolver aplicações para aparelhos da Apple como o iMac, iPhone, iPad, Apple TV e Apple Watch. Ele é uma IDE gratuita que pode ser baixada do site developer.apple.com e fornece ferramentas para gerenciar todo o fluxo de desenvolvimento de aplicações, que vão da criação do aplicativo, passam pelos testes, pela otimização e vão até o envio do software para App Store [99].

O pacote do Xcode fornecido pela Apple inclui a IDE Xcode, os compiladores para as linguagens Swift e Objective-C, ferramentas de análise e depuração e simuladores. É capaz de integrar-se ao GitHub, Bitbucket e GitLab, o que facilita o gerenciamento de código, o *backup* de projetos e a colaboração com outros desenvolvedores.

O UIKit é um *frameworks* que fornece a infraestrutura necessária para o desenvolvimento de aplicativos para iOS ou tvOS. Ele define os componentes centrais desses aplicativos,

como tabelas e botões para *table views* e *navigation controllers*. Fornece a infraestrutura para manipulação de eventos *Multi-Touch* e outros tipos de entrada. É responsável pelo *loop* de execução principal que é fundamental para gerenciar as interações entre o usuário, o sistema e o aplicativo. E apresenta ainda outros recursos como: suporte à animação, suporte a documentos, suporte a desenhos e impressão, informações sobre o dispositivo atual, gerenciamento e exibição de texto, suporte à pesquisa, suporte à acessibilidade, suporte à extensão de aplicativos e gerenciamento de recursos [100].

O Alamofire é uma biblioteca desenvolvida completamente em linguagem *Swift* para comunicação de dados via protocolo HTTP. Ele fornece uma interface elegante sobre a pilha de rede da Fundação da Apple, que simplifica várias tarefas comuns de rede. Fornece métodos de requisição e resposta encadeados, parâmetros JSON e serialização de resposta, autenticação, dentre outros recursos [101]. No presente trabalho ele foi utilizado para realizar *upload* de arquivos e requisições a API RESTful do sistema.

O MapKit é um *framework* que permite a incorporação e manipulação de mapas e imagens de satélite em um aplicativo iOS. Ele viabiliza a criação de rotas, a inserção de sobreposições e *annotations* no mapa, para plotagem de pontos de interesse ou destinos do usuário, e permite a inserção de informações relativas a um marcador de coordenadas do mapa [102]. Por meio dele, foi possível carregar o OpenStreetMap no Tô de Olho, inserir os locais de desordens, fazendo marcações no mapa, e apresentar as desordens registradas no sistema na forma de *annotations*.

5.4 Resultados

Durante o desenvolvimento do aplicativo, construiu-se uma série de telas com a finalidade de atender aos requisitos citados na Seção 5.1.3. Na Figura 5.2 é apresentada a tela do `Main.storyboard` do Xcode, isso é, a visão geral de todas as telas construídas durante o desenvolvimento do aplicativo. No decorrer desta seção, dentre todas essas telas, são apresentadas as principais e descorre-se brevemente sobre cada uma.

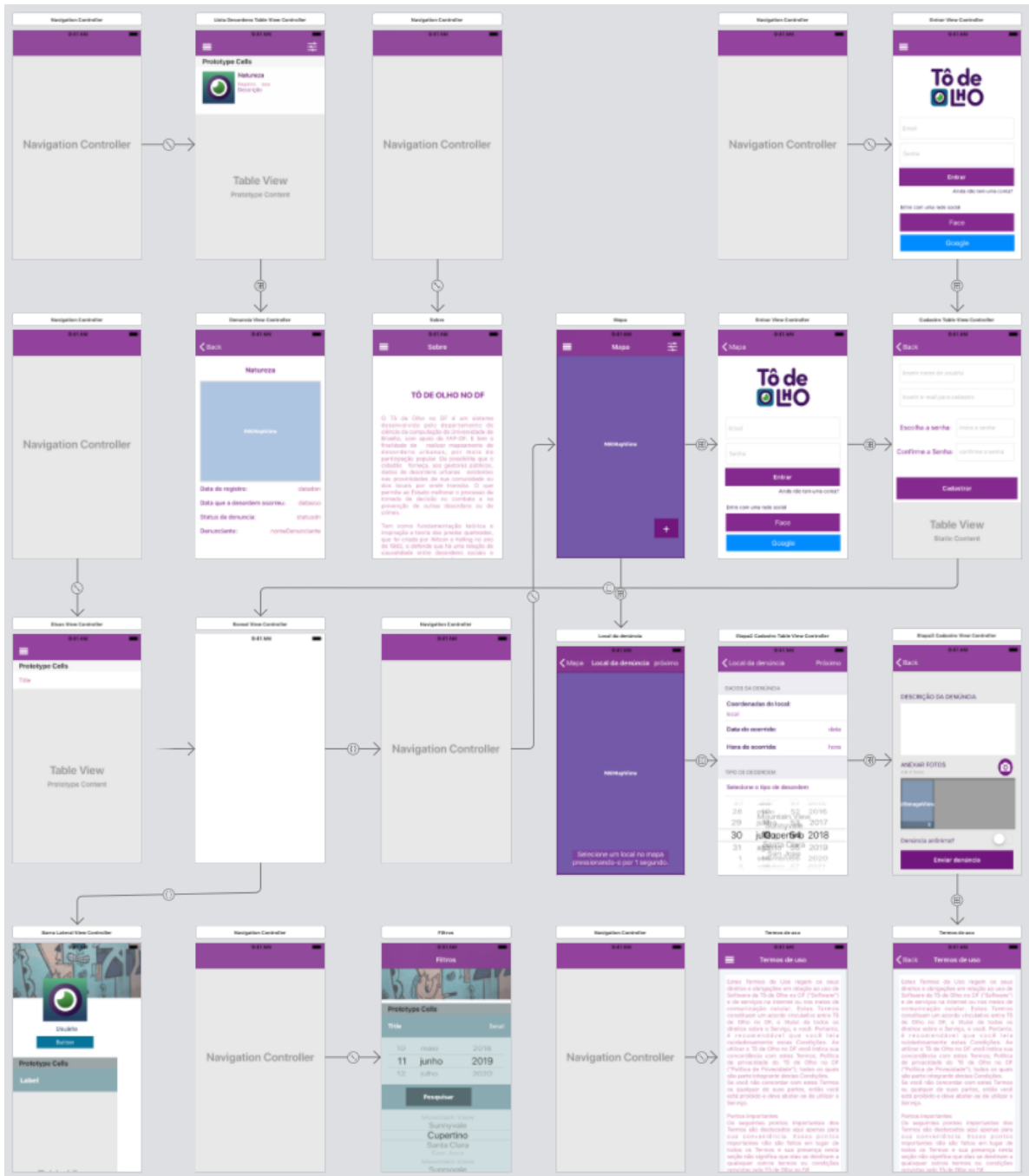


Figura 5.2: Visão geral das telas do aplicativo TÔ de Olho no DF

5.4.1 Tela Inicial

A Figura 5.3 mostra as três primeiras telas do aplicativo, sendo a Figura 5.3a a apresentação da tela de carregamento, que aparece enquanto se aguarda o carregamento completo do aplicativo; nela pode-se ver a assinatura do TÔ de Olho como imagem central. A Figura 5.3b mostra a tela do aplicativo quando este é carregado pela primeira vez, nela apresenta-se uma caixa de mensagem solicitando a autorização do usuário para acessar

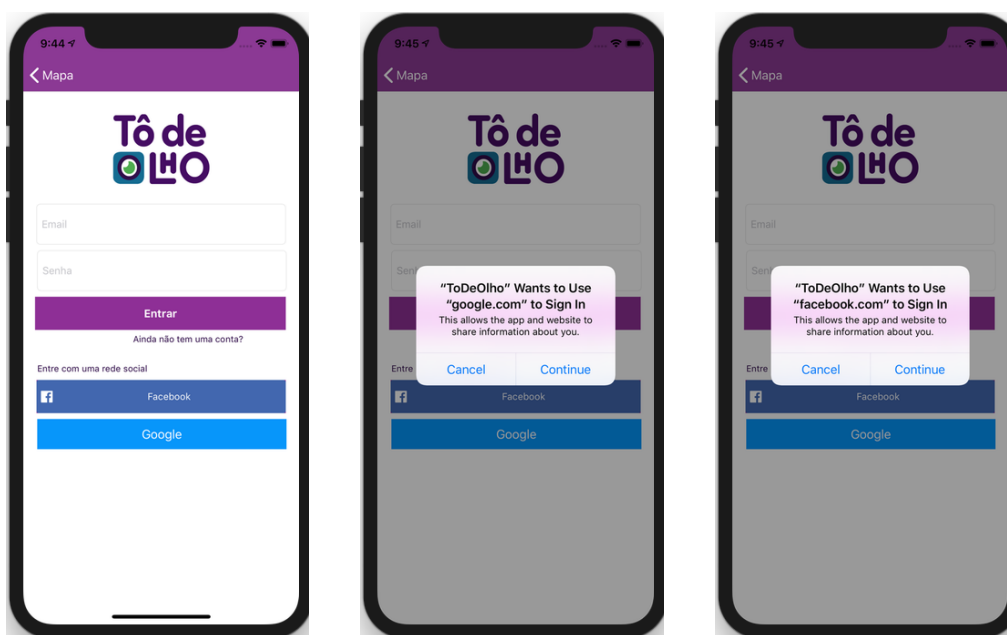
sua localização enquanto estiver usando o aplicativo. Após aceitar que o aplicativo acesse sua localização, o usuário pode navegar pelo mapa e pelas funcionalidades do aplicativo (Figura 5.3c, Figura 5.3d e Figura 5.3e). No entanto, para registrar uma denúncia, o usuário precisa ser cadastrado no sistema e estar logado (Figura 5.3f).



Figura 5.3: Telas iniciais.

5.4.2 *Login*

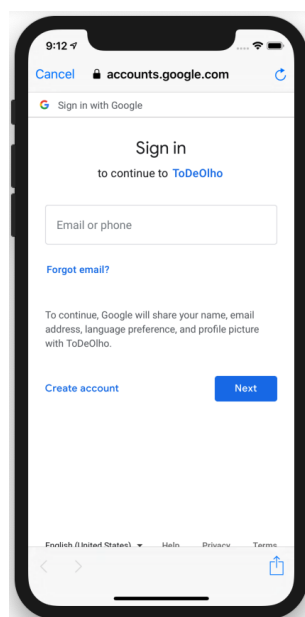
A tela de *login* (Figura 5.4a) é utilizada pelo usuário para realizar a autenticação no sistema. Essa autenticação pode ser realizada utilizando-se um e-mail e senha já cadastrados no sistema ou utilizando-se de contas da Google ou do Facebook (Figura 5.4).



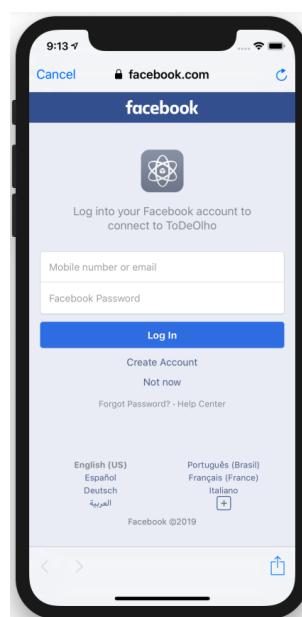
(a) Tela de *login*

(b) Mensagem *login* google

(c) Mensagem *login* facebook



(d) Tela de *login* google



(e) Tela de *login* facebook

Figura 5.4: Telas de *login*.

5.4.3 Cadastro de Usuário

Caso o usuário não tenha registro no sistema e não possua ou não deseje usar contas da Google ou do Facebook, ele pode clicar no botão “Ainda não tem uma conta?” (Figura 5.4a) para abrir a tela de cadastro (Figura 5.5). Por meio desta tela, o usuário pode criar uma conta no sistema, fornecendo seu nome, e-mail e senha. Caso o e-mail inserido já esteja

cadastrado no sistema, o aplicativo emitirá uma mensagem informando ao usuário sobre isso (Figura 5.5b), caso contrário será emitida uma mensagem informando que o cadastro foi bem sucedido (Figura 5.5c).

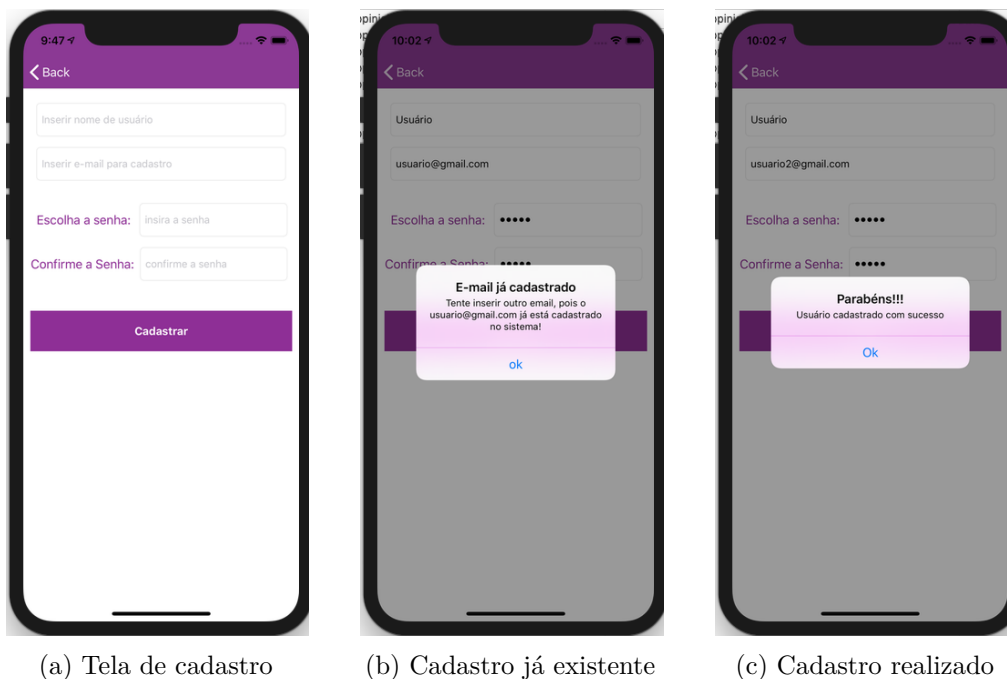
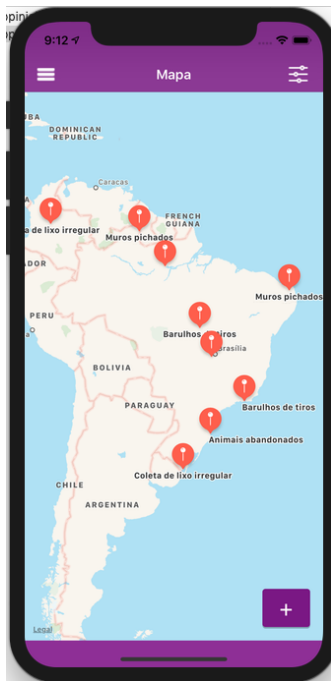


Figura 5.5: Telas do cadastro.

5.4.4 Cadastro de Desordem

Uma das principais funcionalidades do aplicativo é o registro de desordem urbana. Para realizar o usuário deve estar logado no sistema e deve seguir quatro etapas. A primeira consistem em clicar no botão “+” no canto inferior da mapa para iniciar o processo de reportagem de denúncia; a segunda em selecionar no mapa a localização da desordem que deseja reportar e clicar em “próximo”; a terceira em selecionar a data e hora que ocorreu a desordem e sua natureza; e a quarta e última etapa consiste em dar uma descrição ao fato reportado, e, opcionalmente, inserir fotos relacionadas ao que está reportando. O usuário pode optar pelo anonimato da denúncia, para que seu nome não apareça relacionado publicamente àquela denúncia.



(a) Mapa



(b) Localização da desordem



(c) Detalhes da desordem



(d) Fotos da desordem



(e) Descrição da desordem



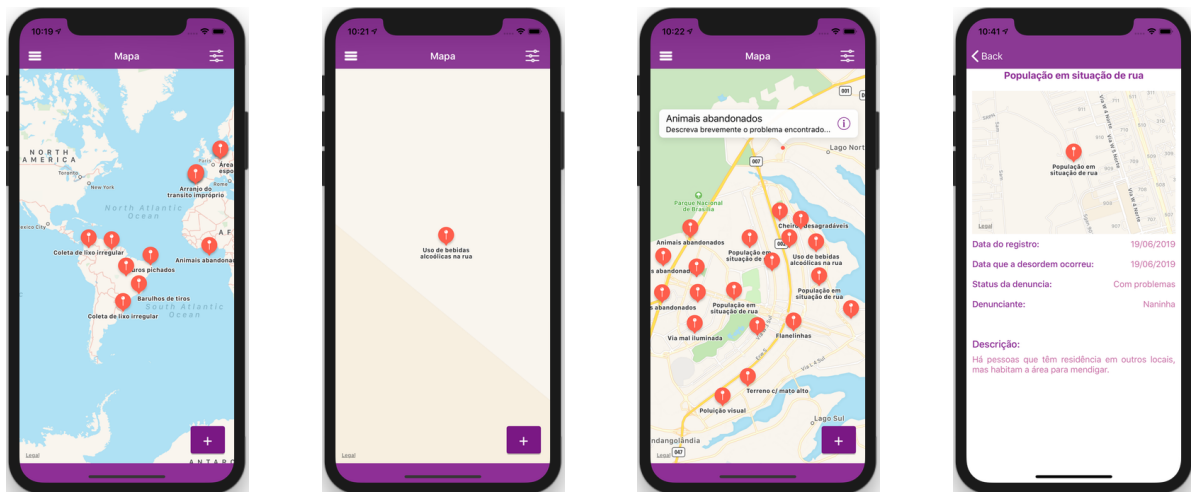
(f) Envio da denúncia

Figura 5.6: Registro de denúncia.

5.4.5 Mapa

A tela de mapa (Figura 5.7) é a principal tela do aplicativo e apresenta um mapa-múndi com marcadores na forma de bolhas, que são implementados por meio de `annotations` da biblioteca `MapKit`. Estes marcadores representam as desordens já registradas no sistema e sempre que o usuário clica sobre um deles é apresentado um “balão” (*callout bubble*) que contém a natureza da denúncia, sua descrição e um botão do tipo `detailDisclosure` (Figura 5.7c). Este botão ao ser acionado carrega a tela de detalhes da denúncia.

Ao usuário é permitida a navegação no mapa, que é realizada por meio de gestos dos dedos do usuário na tela do iPhone. O reconhecimento desses gestos é implementado através da classe `UIGestureRecognizer` da biblioteca `UIKit` da linguagem *Swift*. E toda vez que o usuário manipula o mapa, são recarregadas as denúncias relativas aquela área para a qual ele movimentou o mapa (Figura 5.7d).



(a) Mapa com zoom mínimo

(b) Mapa com zoom máximo

(c) Mapa de desordens

(d) Detalhes de uma desordem

Figura 5.7: Telas de exibição do mapa.

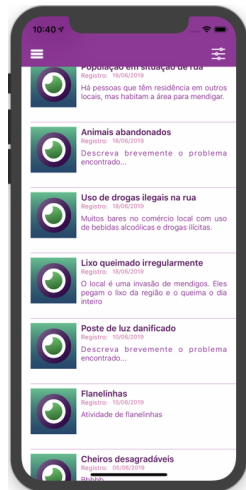
5.4.6 Lista de Desordens

A tela que lista as desordens é acessada por meio do menu lateral (Figura 5.8a) e expõe a lista das mesmas desordens que são apresentadas no mapa principal naquele momento (Figura 5.8b). No entanto, caso o usuário deseje, pode listar outras denúncias utilizando a tela de filtros que é apresentada na Figura 5.8c, nela pode escolher um período de tempo, os tipos de desordens e o *status* das denúncias que deseja pesquisar.

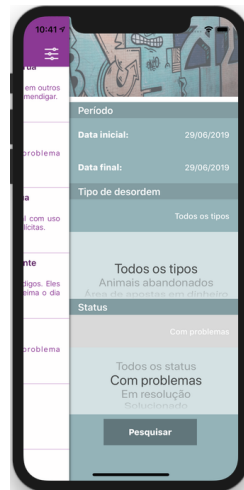
Na tela que lista as denúncias (Figura 5.8b) o usuário pode selecionar uma delas, clicando sobre a denúncia desejada, e poderá visualizar os detalhes dessa denúncia conforme apresentado na Figura 5.8d.



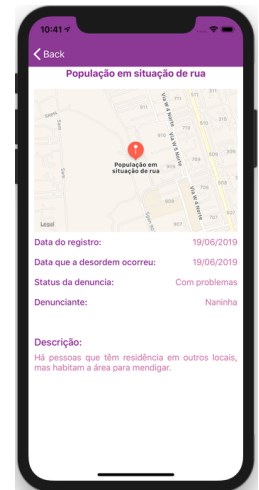
(a) Mapa com zoom mínimo



(b) Mapa com zoom máximo



(c) Mapa de desordens



(d) Detalhes de uma desordem

Figura 5.8: Telas para listar desordens.

Capítulo 6

Prova de Conceito

Neste capítulo são descritos os testes realizados no aplicativo Tô de Olho no DF para garantir um padrão mínimo de aceitação. Os testes foram realizados no intuito de identificar eventuais problemas ou falhas no aplicativo. Para alcançar esse intento, optou-se por realizar dois tipos de testes: de funcionalidade e de usabilidade. Esses testes são apresentados nas duas Seções: Seção 6.1 Teste Funcional e Seção 6.2 Teste de Usabilidade.

6.1 Teste Funcional

O teste funcional é uma técnica usada para verificar as funcionalidades e regras de negócio que são exigidas para um determinado *software*. Nesse tipo de técnica o *software* é avaliado por meio do ponto de vista do usuário, isto é, não é levado em conta os detalhes da implementação. Dessa forma, o aplicativo é visto como uma caixa-preta [103].

Para realizar o teste funcional do aplicativo Tô de Olho no DF, foi utilizado um aparelho *smartphone* com a seguinte descrição:

Tabela 6.1: Descrição do *smartphone* utilizado para o teste funcional

Marca	Apple
Modelo	iPhone 8
Sistema Operacional	iOS 12.3.1
Chipset	Apple A11 Bionic
Processador (CPU)	Hexa-core
Memória RAM	2 GB de RAM
Armazenamento Interno	64 GB

Os testes de funcionalidade se basearam nos requisitos descritos na Seção 5.1.3, foram realizados pelo desenvolvedor e são descritos abaixo:

- Teste 1: Realizar cadastro;
- Teste 2: Realizar *login* com mesmo usuário que foi cadastrado;
- Teste 3: Inserir uma denúncia no sistema;
- Teste 4: Carregar o mapa e verificar se a denúncia inserida aparece corretamente no mapa;
- Teste 5: Movimentar o mapa, aumentar e diminuir o zoom para verificar se novas ocorrências são carregadas corretamente;
- Teste 6: Listar denúncias e verificar se condizem com as denúncias apresentadas no mapa;
- Teste 7: Clicar em uma das denúncias para verificar se os detalhes são exibidos;
- Teste 8: Utilizar o filtro de denúncias para pesquisar denúncias por meio de critérios diferentes;
- Teste 9: Fazer *logout* no aplicativo;
- Teste 10: Fazer *login* no aplicativo usando uma conta do Google;
- Teste 11: Fazer *login* no aplicativo usando uma conta do Facebook;

Esses testes foram repetidos por diversas vezes durante o processo de desenvolvimento do aplicativo, e os erros e falhas detectados foram corrigidos a medida que eram identificados. Esse processo foi repetido pelo desenvolvedor até o ponto em que não se identificou mais erros ou falhas.

6.2 Teste de Usabilidade

O teste de usabilidade é outra técnica de teste de caixa-preta, focada na experiência do usuário em relação ao *software*, e seu objetivo é submeter o produto aos usuários reais para identificar problemas, erros e pontos de melhorias [104].

Em relação ao aplicativo Tô de Olho no DF, esses testes foram realizados após superada a fase de desenvolvimento e de testes funcionais. E pode-se considerar que foi realizado em duas fases: a primeira, quando o aplicativo foi submetido à *App Store Connect*, para avaliação e publicação do aplicativo na *App Store*; e a segunda, quando o aplicativo já estava disponível na *App Store* e foi divulgado ao público para que estes usassem o aplicativo e respondessem o questionário de usabilidade (Anexo A - Formulário de Usabilidade do Aplicativo Tô de Olho no DF).

6.2.1 Testes Realizados pela Equipe da *App Store Connect*

Os testes realizados pela equipe da *App Store Connect* apontaram um erro a ser corrigido e duas melhorias a serem realizadas. O erro se tratava do fechamento inesperado do aplicativo quando se clicava na opção “Dicas de Segurança” (Figura 6.1 e Figura 6.4a). E as mensagens sobre melhorias sugeridas pela equipe da *App Store Connect* estavam relacionadas a política de privacidade da Apple. A primeira, solicitava que explicasse melhor por qual motivo o aplicativo necessitava ter acesso a localização do usuário (Figura 6.2 e Figura 6.5a). E a outra, solicitava que se explicasse por qual motivo era preciso a data de nascimento do usuário para fazer cadastro no sistema (Figura 6.3).

Guideline 2.1 - Performance - App Completeness

We discovered one or more bugs in your app when reviewed on iPad running iOS 12.3.1 on Wi-Fi.

Specifically, your app closed unexpectedly when we tapped on “Dicas de Segurança” tab.

Figura 6.1: Mensagem sobre *bug* enviada pela equipe da *App Store Connect*.

Guideline 5.1.1 - Legal - Privacy - Data Collection and Storage

We noticed that your app requests the user's consent to access their location but does not clarify the use of the location in the applicable purpose string.

Figura 6.2: Mensagem sobre melhoria a ser realizada em relação a solicitação de acesso a localização do usuário.

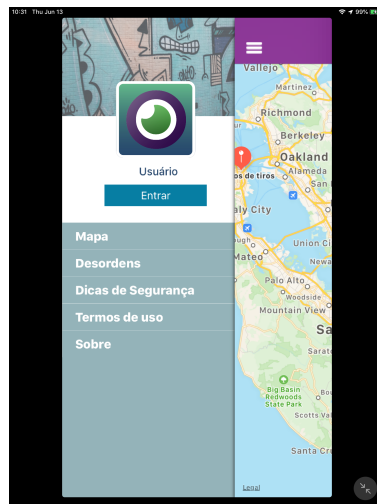
Guideline 5.1.1 - Legal - Privacy - Data Collection and Storage

We noticed that your app requires users to register with personal information that is not directly relevant to your app's core functionality. Specifically, the following fields are required but do not appear to be directly relevant to your app's core functionality:

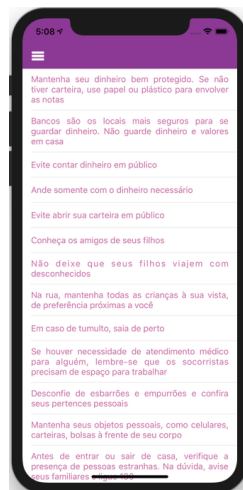
- Data de nascimento

Figura 6.3: Mensagem sobre melhoria a ser realizada em relação a solicitação de data de nascimento do usuário para cadastro no sistema.

O erro apontado pela equipe da *App Store Connect* foi corrigido e as melhorias sugeridas foram acatadas. Os resultados são apresentados nas Figuras 6.4, 6.5 e 6.6. Após realizar essas correções, o aplicativo foi submetido novamente à *App Store Connect* e foi aprovado para a *App Store*.

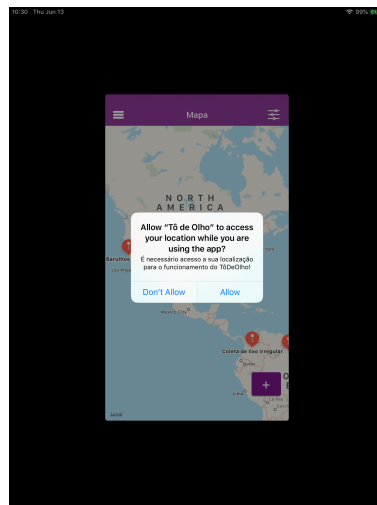


(a) *Print* enviado pela Apple indicando o local que o erro ocorria.

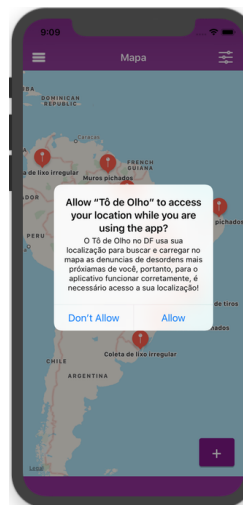


(b) Tela de Dicas de Segurança apresentada após o erro ter sido corrigido.

Figura 6.4: Comparação entre telas.

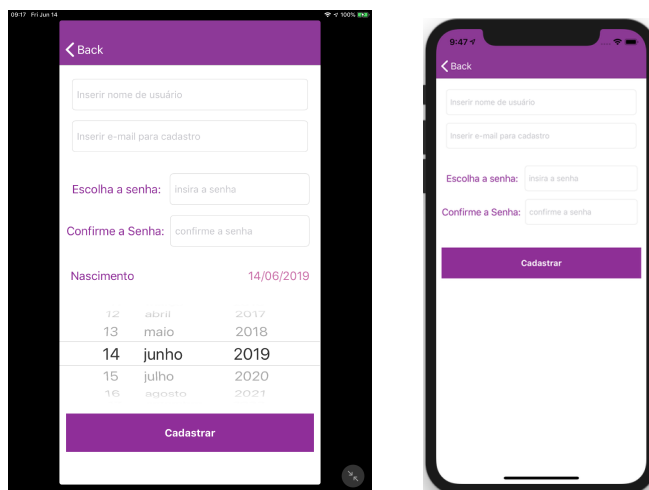


(a) *Print* enviado pela Apple contendo a mensagem a ser melhorada.



(b) Tela apresentada após a melhoria da mensagem que solicita acesso à localização do usuário.

Figura 6.5: Comparação entre telas.



(a) *Print* enviado pela Apple contendo a tela de cadastro com data de nascimento. (b) Tela de cadastro apresentada após a retirada do campo referente a data de nascimento.

Figura 6.6: Comparação entre telas.

6.2.2 Testes Realizados pelos Usuários Finais

O teste de usabilidade para usuário final consistiu em enviar uma mensagem, nas redes sociais (Facebook e Whatsapp), contendo o *link* para *download* do aplicativo Tô de Olho no DF na *Apple Store*, e contendo o *link* para o formulário de usabilidade no Google Docs. A mensagem foi enviada à cerca de quatrocentas pessoas, e o formulário esteve disponível para preenchimento no período de 18 a 29 de julho de 2019.

Apesar da solicitação de teste do aplicativo ter sido enviada à cerca de quatrocentas pessoas, observou-se que o aplicativo foi baixado apenas trinta e nove vezes na *Apple Store* (Figura 6.7) e que apenas vinte e três pessoas preencheram o formulário no Google Docs. Este formulário é constituído de duas partes: a caracterização de perfil do usuário e a percepção dos usuários sobre o aplicativo Tô de Olho no DF. A apresentação dos resultados provenientes de seu preenchimento será feita nas próximas subseções.

Antes de seguir às próximas subseções, é conveniente ressaltar que além da quantidade de *downloads* realizados, o *App Analytics* da *App Store Connect* apresenta outros dados sobre o aplicativo, que podem ser vistos nas Figuras 6.8 e 6.9. Nesta última, observa-se dados que indicam a origem dos usuários que visitaram o aplicativo na *Apple Store*. Verifica-se que 62% desses usuários foram direcionados a *Apple Store* por meio de um *link* compartilhado fora da loja e que 38% dos usuários visualizaram o aplicativo após realizar pesquisa na própria loja da Apple.



Figura 6.7: Quantidade de *downloads* do aplicativo Tô de Olho no DF realizados na *Apple Store*.

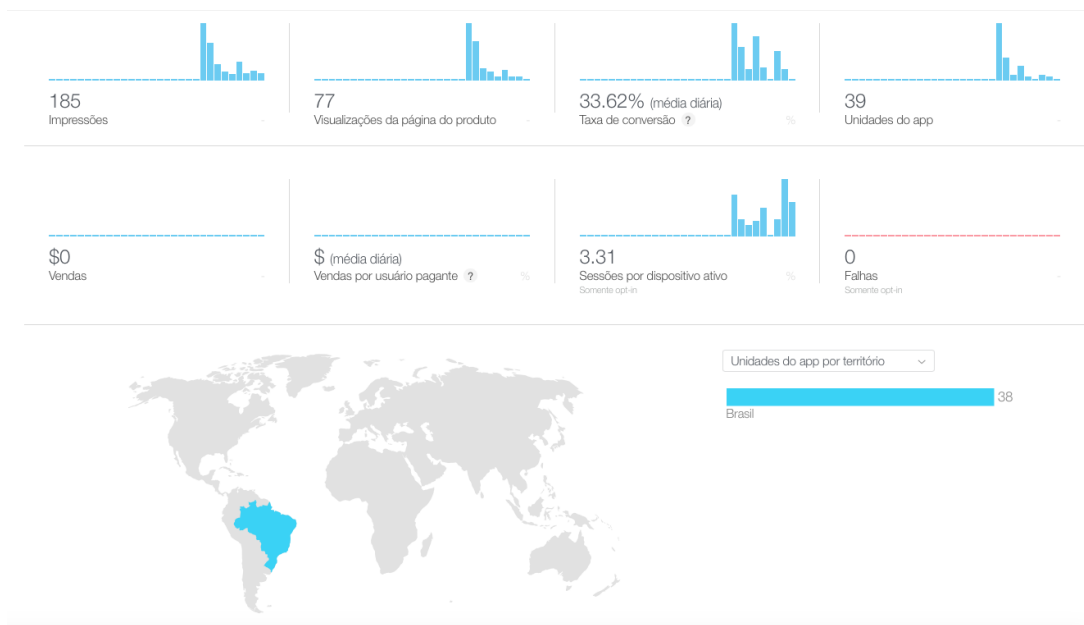


Figura 6.8: Dados do *App Analytics* sobre o aplicativo Tô de Olho no DF .

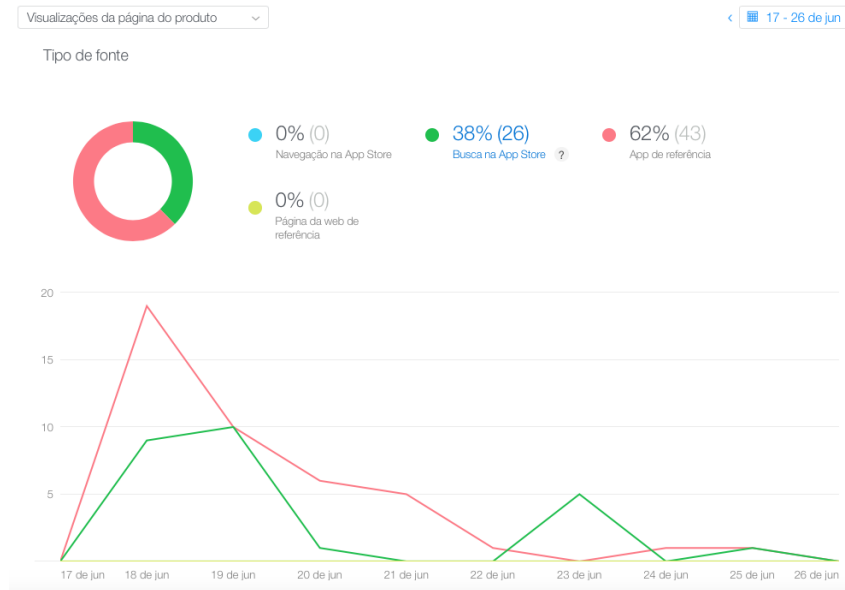


Figura 6.9: Dados do *App Analytics* sobre o aplicativo Tô de Olho no DF .

6.2.2.1 Caracterização de Perfil dos Usuários

Pelos resultados obtidos, observa-se que, das 23 pessoas que preencheram o formulário de usabilidade do aplicativo, apenas uma possuía nível médio e apenas uma possuía nível superior incompleto, portanto, houve uma prevalência de 91,3% de pessoas com nível superior completo em relação as demais (Figura 6.10).

Qual o seu grau de escolaridade?

23 respostas

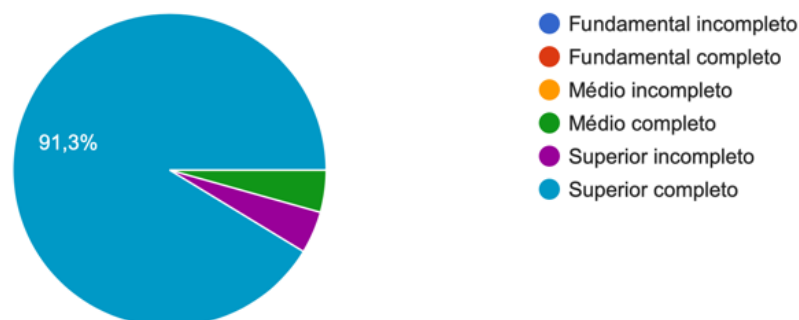


Figura 6.10: Grau de Escolaridade.

A maioria dos respondentes, 91,3%, são bem familiarizados com *smartphones* (Figura 6.11) e costumam fazer uso de aplicativos em seus *smartphones* (Figura 6.12).

Qual a sua familiaridade com Smartphones?

23 respostas

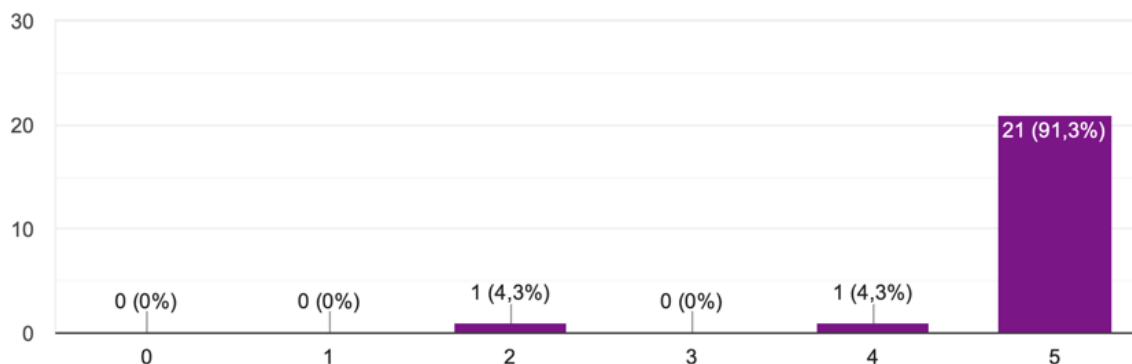


Figura 6.11: Familiaridade com *Smartphones*.

Faz uso de aplicativos de Smartphones? Se sim, aponte em média em quantos dias da semana você utiliza aplicativos no Smartphone?

23 respostas

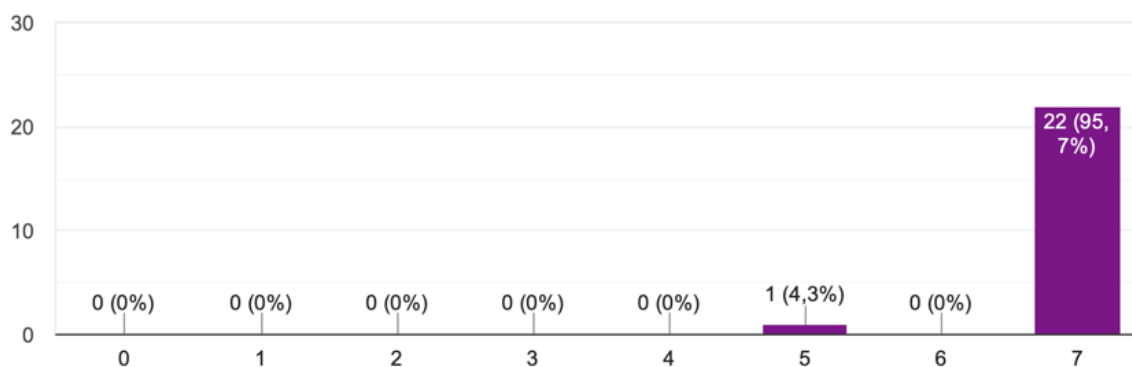


Figura 6.12: Frequência com que usa *Smartphones*.

Quando questionados sobre o seu histórico de uso de aplicativos de colaboração popular, observa-se que 69,6% dos respondentes já usaram algum aplicativo desse tipo (Figura 6.13), e que 63,6% deles também já utilizaram algum aplicativo de mapeamento colaborativo (Figura 6.14). No entanto, apesar dos respondentes em geral usarem aplicativos colaborativos a Figura 6.15 demonstra que apenas quatro dessas pessoas, 17,4%, já utilizaram algum aplicativo semelhante ao Tô de Olho. Esse resultado poder ser fruto de diversos

fatores, tais como, a baixa popularidade dos aplicativos de mapeamento de desordens, a falta desse tipo de aplicativo no mercado, ou a falta de divulgação desse tipo de aplicativo, dentre outros.

Já utilizou algum aplicativo de colaboração popular, onde as pessoas forneçam dados de forma participativa?

23 respostas

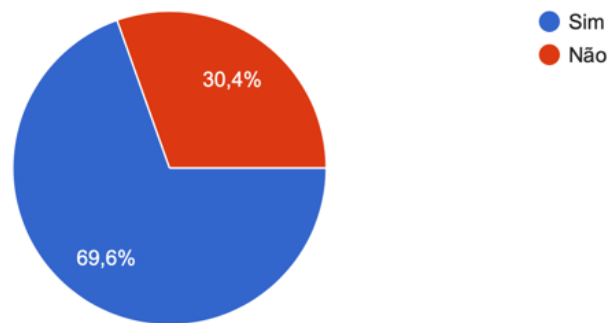


Figura 6.13: Histórico de uso de aplicativos colaborativos.

Já utilizou algum aplicativo de mapeamento colaborativo?

22 respostas

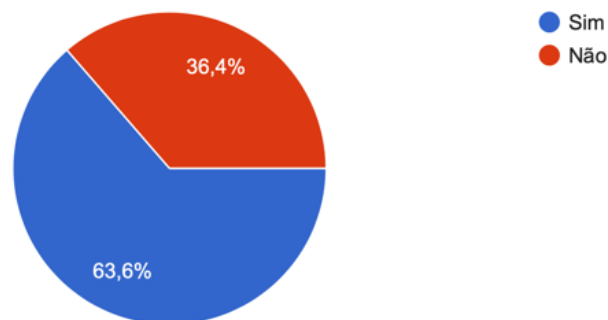


Figura 6.14: Histórico de uso de aplicativos de mapeamento colaborativo.

Já utilizou algum aplicativo semelhante ao Tô de Olho?

23 respostas

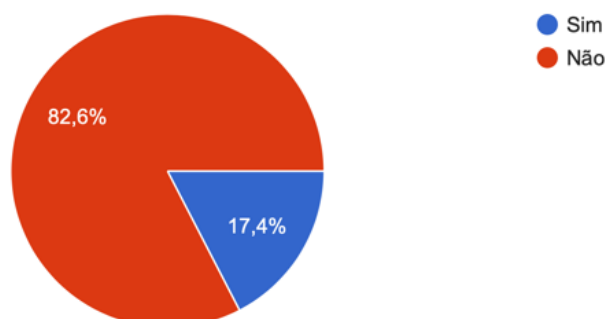


Figura 6.15: Histórico de uso de aplicativos semelhantes ao Tô de Olho no DF.

Quando questionados sobre o seu nível de participação em redes sociais ou em aplicativos de colaboração popular, observa-se que não há uma distribuição uniforme ou uma prevalência absoluta em um único nível. Apesar disso, verificou-se que a maioria dos respondentes se classificou entre os três níveis que indicam maior participação, isto é, 17 pessoas se classificaram do nível 3 ao nível 5 de participação, sendo oito (34,8%) no nível 3, quatro (17,4%) no nível 4 e 2 (8,7%) que se consideram totalmente participativos (Figura 6.16).

Como você avalia seu nível de participação em redes sociais ou em aplicativos colaboração popular?

23 respostas

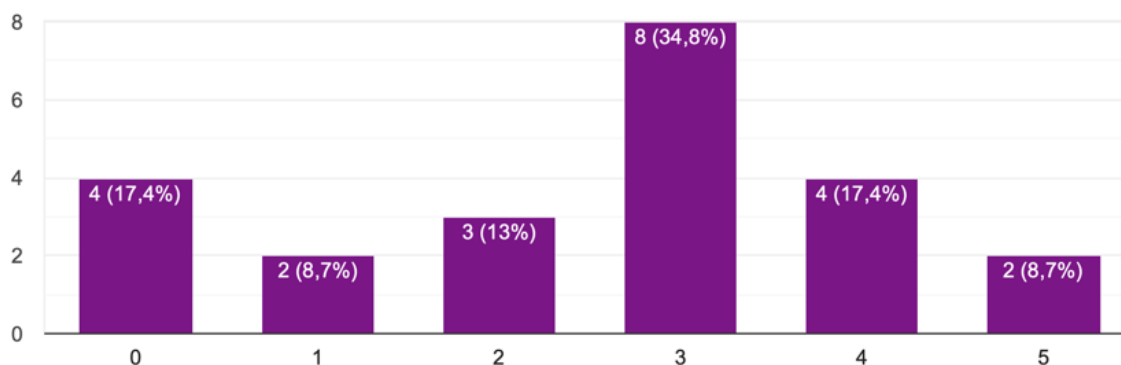


Figura 6.16: Nível de participação em redes sociais ou aplicativos de colaboração popular.

Como você classificaria a ordem/conservação do bairro onde mora?

23 respostas

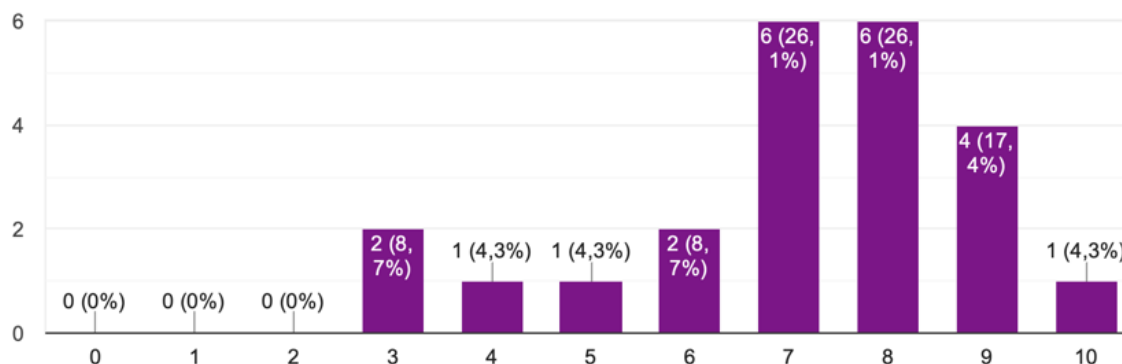


Figura 6.17: Classificação da ordem/conservação no bairro onde reside.

O dados da Figura 6.17 indicam que os respondentes residem, no geral, em local limpo, bem conservado e com baixa desordem, havendo uma prevalência daqueles que classificam o seu bairro com nota maior ou igual a 6.

Você estaria disposto a relatar as desordens presentes no seu bairro em uma plataforma de mapeamento colabo...s órgãos da administração pública?

23 respostas

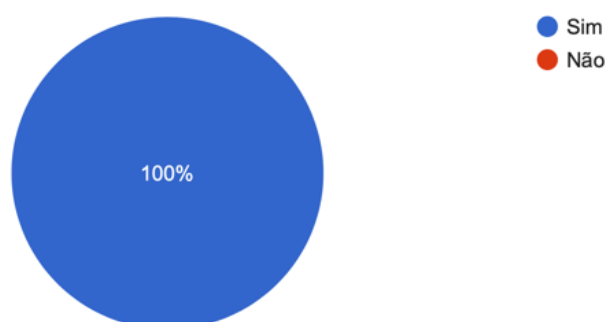


Figura 6.18: Pré-disposição para relatar desordens em seu bairro por meio de aplicativo de colaboração.

Pelo exposto, pode-se concluir que, no geral, o perfil dos avaliadores do Tô de Olho no DF é um perfil de usuário de nível superior, que possui familiaridade no uso de *smartphones*, que já utilizou aplicativos de colaboração popular, inclusive aplicativos de

mapeamento colaborativo, que residem em bairros que não existem muitas desordens e que estão dispostos a relatar desordens às autoridades públicas, por meio de um aplicativo de colaboração popular.

6.2.2.2 Percepção dos Usuários Sobre o Aplicativo Tô de Olho no DF

A segunda parte do formulário de usabilidade do aplicativo fez questionamentos aos respondentes a respeito de sua percepção em relação ao aplicativo Tô de Olho no DF.

Dos resultados obtidos, verifica-se inicialmente, Figura 6.19, que os respondentes utilizaram diversos modelos de iPhone, o que foi bom para o teste de usabilidade, uma vez que haverá diferentes tamanhos de telas e provavelmente diferentes sistemas operacionais realizando o teste. Da lista modelos disponibilizada no formulário, só os modelos de iPhone 4, 4S, 5C, 6S Plus, XS e XS Max não foram usados pelos respondentes. E os modelos de iPhone 7 e X foram os mais utilizados, representando 17,4% cada, em relação a todos os outros.

Qual o modelo de iPhone você utilizou para testar o Tô de Olho?

23 respostas

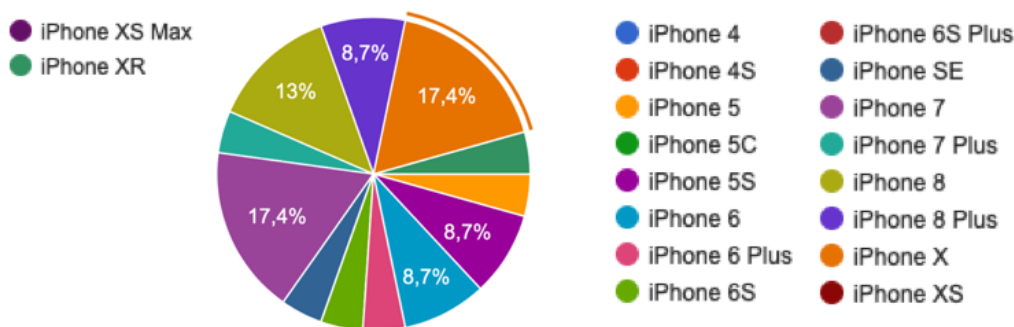


Figura 6.19: Modelo de iPhone usado.

Observou-se que todos os respondentes conseguiram se cadastrar no sistema (Figura 6.20) e que a maioria (82,6%) considerou as cores e botões do aplicativo agradáveis (Figura 6.21). O mesmo percentual de respondentes relatou que conseguiu visualizar bem os botões e informações do aplicativo (Figura 6.22). No entanto, quatro (17,4%) dos respondentes responderam que não conseguiram visualizar bem todos os botões e informações dentro do aplicativo. E achou-se por bem investigar melhor as causas dessa dificuldade. Em um primeiro momento, levantou-se a hipótese do ocorrido estar relacionado ao tamanho da tela do modelo de iPhone usado para o teste. Porém, ao se verificar os modelos relacionados a essas respostas identificou-se os modelos de 5S, 7 e 8, que também foram usados em outras

respostas que não indicaram dificuldades de visualização. Portanto, neste momento, não foi possível indicar com precisão quais dificuldade de visualização de botões ou informações, esses usuários tiveram.

Você conseguiu se cadastrar se no aplicativo?

23 respostas

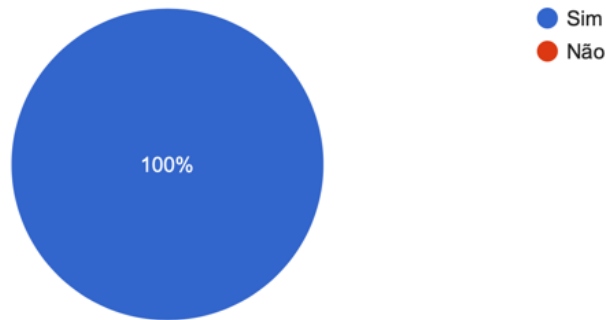


Figura 6.20: Cadastro no Aplicativo.

Você considera as cores e botões do aplicativo agradáveis?

23 respostas

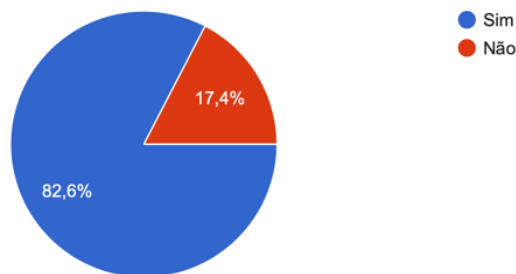


Figura 6.21: Cores e botões do Aplicativo.

Quanto a facilidade de entendimento das palavras, nomenclaturas e ícones usados no aplicativo, houve unanimidade em dizer que conseguiram entender (Figura 6.23). Igualmente houve unanimidade de todos os respondentes em relatar que conseguiram visualizar o mapa com os marcadores (Figura 6.24) e que conseguiram acessar a lista de desordens cadastradas (Figura 6.26).

Conforme a Figura 6.25 observa-se que a maioria dos respondentes (91,3%) compreendeu o que era apresentado no mapa de desordens, restando apenas dois respondentes que não compreenderam bem o conteúdo exposto.

Você conseguiu visualizar bem todos os botões e informações dentro do aplicativo?

23 respostas

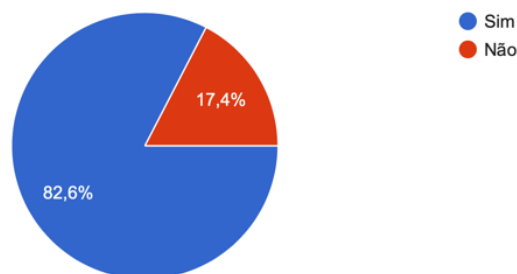


Figura 6.22: Visualização de botões e informações.

Você conseguiu entender com facilidade as palavras, nomenclaturas e ícones do aplicativo?

23 respostas

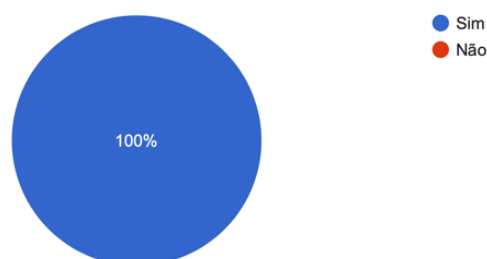


Figura 6.23: Entendimento da nomenclatura do aplicativo.

Quando questionados se conseguiram ver os detalhes de uma desordem em particular, 87% dos respondentes reponderam que sim, restando apenas três que relataram que não (Figura 6.27). De forma semelhante, a maioria (90,9%) respondeu que conseguiu fazer *login* no aplicativo com a conta cadastrada (Figura 6.28). Observa-se que apenas dois (9,1%) dos respondentes não conseguiram fazer *login* no aplicativo; ao se analisar melhor as respostas desses dois respondentes, verifica-se também que não conseguiram visualizar bem todos os botões e informações dentro do aplicativo, que não conseguiram ver os detalhes de uma desordem em particular e que relataram que não conseguiram fazer *login* com o Facebook.

Quanto ao cadastramento de uma nova denúncia, a maioria dos questionados (86,4%) respondeu que conseguiu realizar o cadastro, sendo que apenas três deles não tiveram sucesso no cadastro da denúncia (Figura 6.29). E quando questionados se conseguiram

Você conseguiu visualizar o mapa com os marcadores?

23 respostas

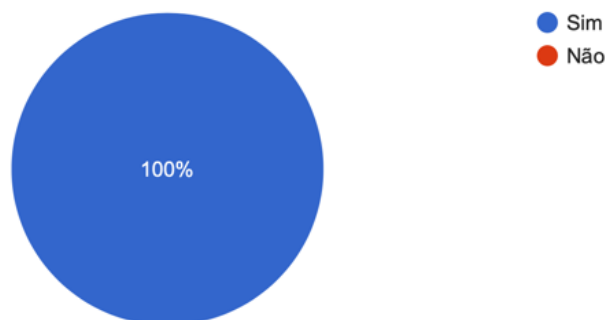


Figura 6.24: Visualização do mapa com marcadores.

Você conseguiu compreender o conteúdo que era apresentado no mapa?

23 respostas

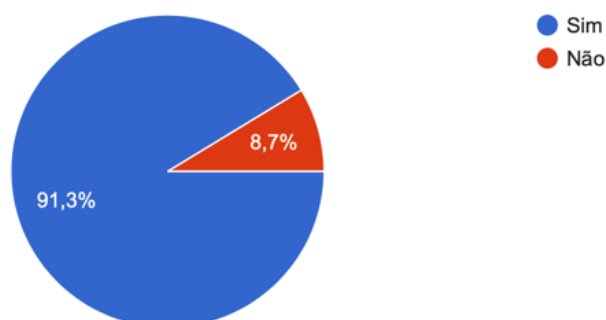


Figura 6.25: Compreensão do conteúdo do mapa.

mover o mapa da aplicação e ver outras denúncias conforme alterava o zoom do mapa, apenas um respondeu que não conseguiu (Figura 6.30).

Quando questionados se consideravam que o aplicativo Tô de Olho poderia ajudar a administração pública a mapear e solucionar desordens sociais, todos os respondentes foram unânimes em responder que consideravam o aplicativo totalmente útil para o propósito (Figura 6.31).

Pela respostas obtidas e exposta acima, verificou-se que o aplicativo Tô de Olho no DF teve boa aceitação e que na maioria dos casos funcionou conforme o esperado. Apesar disso, optou-se por investigar melhor os casos de insucesso relatados, fazendo a análise de todas

Você conseguiu acessar a lista das desordens cadastradas?

23 respostas

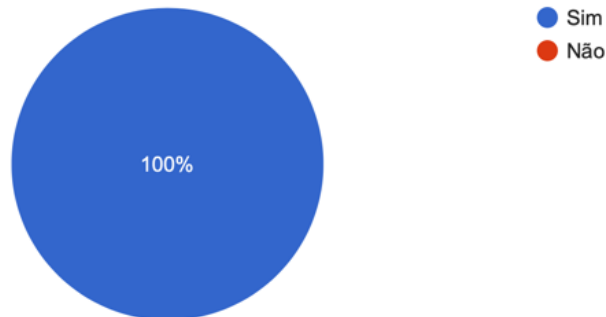


Figura 6.26: Listar desordens.

Você conseguiu ver os detalhes de uma desordem em particular?

23 respostas

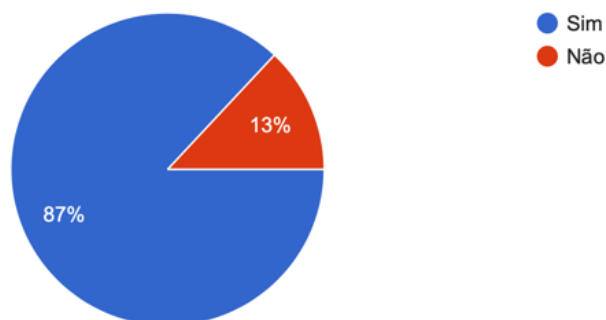


Figura 6.27: Apresentação de detalhes de uma desordem.

as respostas, de cada usuário que relatou algum insucesso, incluindo suas considerações subjetivas expostas nas Figuras 6.32 e 6.33. Diante dessa análise, observou-se que a maioria dos casos de insucesso estavam relacionados à realização de *login* por meio do Facebook. Dessa forma, diante da hipótese de haver alguma inconsistência no *login* realizado por meio do Facebook, fez-se uma análise da parte do código relacionada ao *login* e identificou-se uma falha. Esta falha foi corrigida e o aplicativo foi disponibilizado novamente na Apple Store.

Você conseguiu efetuar o login com a conta cadastrada?

22 respostas

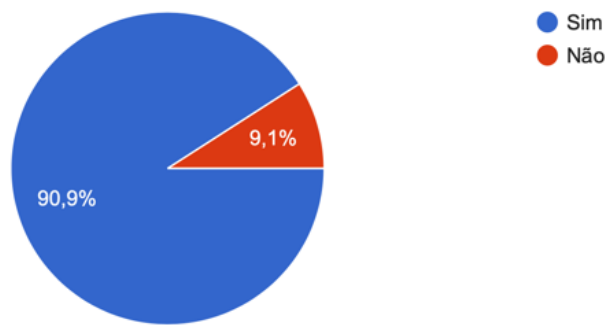


Figura 6.28: Realização de *login*

Você conseguiu cadastrar uma nova denúncia?

22 respostas

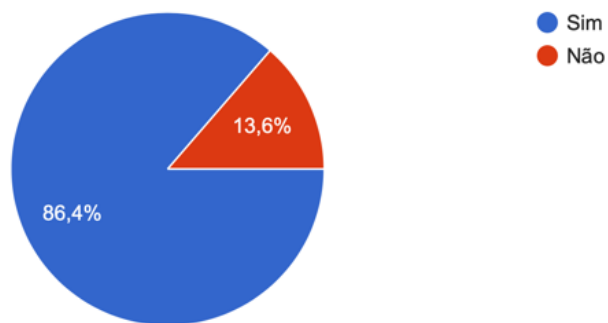


Figura 6.29: Cadastro de denúncia.

Você conseguiu mover o mapa da aplicação e ver outras denuncia conforme alterava o zoom do mapa?

22 respostas

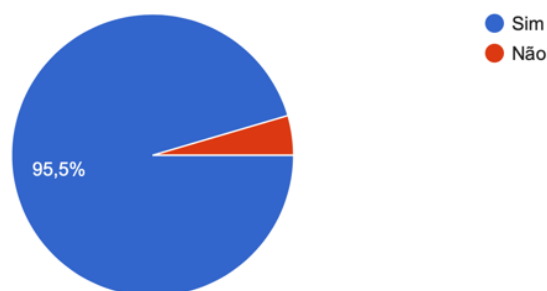


Figura 6.30: Movimentação do mapa e carregamento de denúncias.

Você considera que o aplicativo Tô de Olho pode ajudar a administração pública a mapear e solucionar desordens sociais?

23 respostas

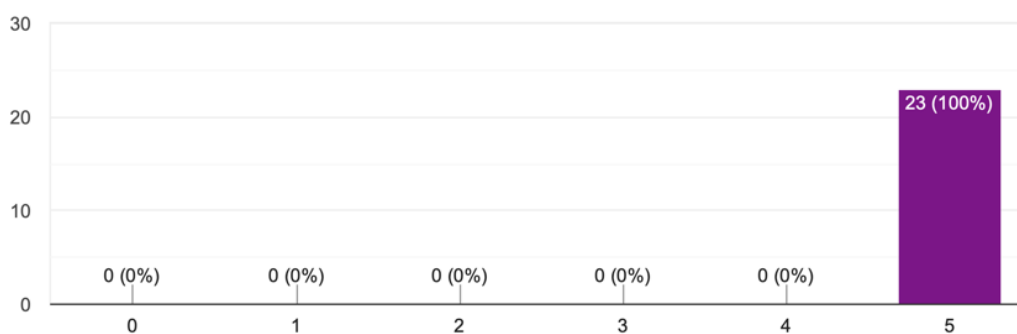


Figura 6.31: Opinião sobre potencial do Tô de Olho no DF para auxiliar a administração pública.

Qual foi a sua primeira impressão sobre a aplicação?

23 respostas

Tentei fazer login com minha conta do Facebook,porém não deu certo,mas depois eu consegui criar!
Achei o aplicativo bastante intuitivo.
Interface de fácil uso e boa apresentação
O app não está funcionando para inserções de desordens.
Excelente aplicativo e de fácil interação e usabilidade
Intuitiva.
Achei super interessante, dinâmico e de aplicação real com retorno para sociedade.
Ele é fácil e importante para a sociedade.
De utilidade
Funcional, e funcionalidades bem integradas.
Fácil de utilização com boa experiência para o usuário
Uma boa ideia, mas depende da adesão das autoridades pra ter eficácia.

Figura 6.32: Primeira impressão sobre Tô de Olho no DF.

Na sua opinião, o que poderia ser feito para melhorar o ToDeOlho?

13 respostas

O cadastro poderia funcionar com a entrada pelo Facebook; quando fui cadastrar a denúncia tive que abrir a câmera e cancelar para seguir, do contrário ficaria parado no teclado como se fosse obrigatório inserir foto; a tecla próximo poderia ser de outra cor para chamar mais atenção (qualquer outra que destacasse)
Maior divulgação para mais conquistar mais colaborações
O aplicativo ficou bem completo, achei uma excelente iniciativa
Acompanhar em tempo real a "solução" do poder público. Ou receber por email a solução indicada pelo gestor público.
Somente disponibilizar a visualização do mapa também na versão satélite
Relacionar as desordens por tipo ou categoria.
Poderia ser criado uma página para o usuário acompanhar o andamento de suas próprias denúncias, para verificar o tempo que os órgãos públicos levaram para resolver o problema denunciado
A principal melhorar que vejo no aplicativo está relacionado ao envolvimento dos órgãos públicos, afim de eles Solucionarem as desordens
A título de sugestão, para criar o login ele pede para fornecer um nome de usuário mas dai aparece o e-mail no campo abaixo da for na tela inicial. Seria legal se mostrasse o nome de usuário.

Figura 6.33: Considerações para melhoria do Tô de Olho no DF.

Capítulo 7

Conclusão

Este trabalho apresentou o Tô de Olho no DF, um Sistema de Informação Geográfica com Participação Popular para realização de mapeamento de desordens urbanas no Distrito Federal. O Tô de Olho no DF possibilita a colaboração do cidadão com os órgãos de segurança pública, e por meio dos dados fornecidos, permite ao gestor público adotar medidas para corrigir e controlar essas desordens sociais, tendo como principal consequência disso a prevenção criminal, conforme explica a teoria das janelas quebradas. Isso é, se há uma relação de causalidade entre desordem e crime, então, combatendo-se as desordens pode-se prevenir certos tipos de crimes ligados a essas desordens.

Para verificar quais as tecnologias e soluções que estão sendo utilizadas no mercado para realização de mapeamento de desordens e crimes levou-se em conta o trabalho de Passos e Holanda [41] chamado “Mapeamento Colaborativo de Desordens Sociais por meio de Sistema de Informação Geográfica com Participação Popular no Brasil”, em que realizou-se a análise de uma lista de aplicações gratuitas que realizavam mapeamento criminal e de desordem social por meio da participação popular.

Diante disso, o sistema foi projetado e implementado para cativar os cidadãos à colaborarem com os órgãos de segurança pública. Foi desenvolvido levando em consideração requisitos que permitissem criar um sistema amigável e intuitivo para todos os usuários. A arquitetura foi pensada para fornecer serviços à plataformas *mobile* e a plataformas *web*, de forma segura e sem muitas dificuldades de integração. Por isso, utilizou-se o estilo de arquitetura REST, que possibilita a criação de uma API de comunicação que usa protocolo HTTP e utiliza-se do modelo cliente-servidor.

O aplicativo para iOS, por sua vez, foi desenvolvido para se comunicar com o servidor, fornecendo dados de desordens inseridos pelo usuário e consumindo serviços oferecidos pelo servidor. A interface do aplicativo também foi desenhada para ser simples e intuitiva, de forma que o colaborador pudesse inserir no mapa uma desordem que presenciou ou que teve conhecimento sem dificuldades. Foi desenvolvido em linguagem *swift* e utilizando apenas

uma biblioteca que não era nativa da linguagem, o que facilita a manutenção do código. E sua validação foi feita em duas etapas: a primeira por meio de testes de funcionalidade e a segunda por meio de testes de usabilidade. As duas etapas de validação colaboraram para a identificação de erros e falhas, o que possibilitou a realização de correções e melhorias no aplicativo.

A implantação do Sistema Tô de Olho no DF pela Secretaria de Estado da Segurança Pública e da Paz Social do Distrito Federal (DF) seria de grande valia no processo de modernização das ações de segurança pública no DF, pois permitiria a aproximação do Estado e do cidadão, diminuiria a cifra negra de desordens não conhecidas pelo governo, daria maior capilaridade à coleta dos dados de desordens e permitiria a tomada de decisão com base em um universo de dados maior. Além disso, poderia-se ter como consequência um maior envolvimento e comprometimento do cidadão nos assuntos de segurança e bem estar social da sua comunidade local.

Como trabalho futuro, espera-se realizar as seguintes implementações e melhorias no aplicativo:

- Desenvolver funcionalidade para que o usuário possa acompanhar suas próprias denúncias, bem como editá-las;
- Desenvolver uma tela de perfil do usuário, para que possa editar ou atualizar seus dados;
- Desenvolver funcionalidade que permita a cada usuário confirmar ou refutar um denúncia de outro usuário; e
- Implementar a funcionalidade de recuperação de senha;

Referências

- [1] Odon, Tiago Ivo: *Tolerância zero e janelas quebradas: sobre os riscos de se importar teorias e políticas*, 2016. <http://www2.senado.leg.br/bdsf/handle/id/519162>, acesso em 2019-04-07. 1, 11
- [2] Goodchild, Michael F: *Citizens as sensors: the world of volunteered geography*. *GeoJournal*, 69(4):211–221, 2007. 2
- [3] Carton, LJ e WAH Thissen: *Emerging conflict in collaborative mapping: Towards a deeper understanding?* *Journal of Environmental Management*, 90(6):1991–2001, 2009. 2
- [4] Lemos, André: *Arte e Mídia Locativa no Brasil*. *Arte e mídia locativa no Brasil*, Comunicação e mobilidade, páginas 89–108, 2009. 2
- [5] Moreira, Cibele: *Ação integrada de segurança pública reduz índices criminais no DF*. <https://www.agenciabrasilia.df.gov.br/2018/12/22/acao-integrada-de-seguranca-publica-reduz-indices-criminais-no-df>, acesso em 2019-05-13. 2, 39
- [6] Brown, Greg: *Public participation gis (ppgis) for regional and environmental planning: Reflections on a decade of empirical research*. *Journal Of The Urban & Regional Information Systems Association*, 24(2), 2012. 2, 20
- [7] Santos, Rachel Boba: *Crime analysis with crime mapping*. Sage Publications, 3rd edição, 2012. 5, 6, 7, 12, 13, 14, 15
- [8] LeBlanc, Jessica B., Jason Elder, Christopher W Bruce, Rachel Boba Santos, Trina Cook, Elizabeth Rodriguez, Friedrich Steiner e Laura Wyckoff: *Definition and types of crime analysis*. Standards, Methods, & Technology (SMT) Committee White Paper 2014-02, 2014. 6, 12
- [9] Santos, Rachel Boba: *Crime analysis with crime mapping*. Sage Publications, 4rd edição, 2016. 6, 7, 9
- [10] Peterson, Marilyn B: *Applications in criminal analysis: A sourcebook*. Greenwood Press Westport, CT, 1994. 6, 8
- [11] Gottlieb, Steven, Sheldon I Arenberg, Raj Singh *et al.*: *Crime analysis: From first report to final arrest*. Alpha Publishing Montclair, CA, 1994. 6, 8, 13

- [12] Bernardes, Paulo Ventura Silva: *Análise Criminal como Instrumento de Produção De Conhecimento*. Instituto de Pós-Graduação & Graduação (IPOG), 2015. 7, 8
- [13] Souza, Nelson Gonçalves de e Álvaro Pereira da Silva Júnior: *Teorias Criminais de Suporte à Análise Criminal*. I Curso de Extensão - Nível Multiplicadores, 2008. <https://acervodigital.ssp.go.gov.br/pmgo/bitstream/123456789/351/31/Manual%20-%20Curso%20Multiplicador%20de%20Análise%20Criminal%20-%20SENASP.pdf>, acesso em 2019-04-21. 8, 9, 10, 11
- [14] GUILLÉN, César SAN JUAN: *Criminología ambiental: un área en expansión*. AIS: Ars Iuris Salmanticensis, 1(1), 2013, ISSN 2340-5155. <http://revistas.usal.es/index.php/ais/article/view/10106>. 9
- [15] Lino, Denis e Lucas Heiki Matsunaga: *Perfil criminal geográfico: novas perspectivas comportamentais para investigação de crimes violentos no Brasil*. Revista Brasileira de Criminalística, 7(1):7–16, 2018. 9, 11
- [16] Cohen, Lawrence E e Marcus Felson: *Social change and crime rate trends: A routine activity approach*. American sociological review, páginas 588–608, 1979. 10
- [17] Felson, Marcus e Ronald V Clarke: *Opportunity makes the thief*. Police research series, paper, 98:1–36, 1998. 10
- [18] Young, Jock: *A sociedade excludente: exclusão social, criminalidade e diferença na modernidade recente*. Revan, 2002. 11
- [19] Shaw, Clifford Robe e Henry Donald McKay: *Juvenile delinquency and urban areas: A study of rates of delinquency in relation to differential characteristics of local communities in american cities (1969)*. Em *Classics in environmental criminology*, páginas 103–140. CRC Press, 2016. 11
- [20] Peixoto, Betânia Totino: *Análise Criminal*. Brasília: Senasp, 2008. 12, 13
- [21] Dantas, George Felipe de Lima e Nelson Gonçalves de SOUZA: *As bases introdutórias da análise criminal na inteligência policial*. Instituto Brasileiro de Ciências Criminais (IBCCRIM), 2004. 13
- [22] Gottlieb, Steven e Shel Arenberg: *Crime Analysis: From Concept to Reality*. Office of Criminal Justice Planning, 1990. 14
- [23] Groff, Elizabeth R e Nancy G La Vigne: *Forecasting the future of predictive crime mapping*. Crime Prevention Studies, 13:29–58, 2002. 15
- [24] Worboys, Michael F e Matt Duckham: *GIS: a computing perspective*. CRC press, 2004. 15
- [25] Mata, Daniel Queiroz da e Frederico Gouveia Neves Ferreira: *SIU: um aplicativo móvel para coleta e visualização de dados geográficos aplicado ao transporte público do DF*, 2013. 15

- [26] Longley, Paul A, Michael F Goodchild, David J Maguire e David W Rhind: *Geographic information systems and science*. John Wiley & Sons, 2005. 16
- [27] Ramos, Thales Marques: *Sistema de Informações Geográficas Voluntárias: um Estudo de Caso para Plataforma iOS da Apple*. Tese de Doutorado, Universidade de Brasília, 2016. 16
- [28] Lisboa Filho, Jugurta: *Projeto de banco de dados para sistemas de informação geográfica*. SBC-Revista Eletrônica de Iniciação Científica, 2001. 16
- [29] Câmara, Gilberto, Clodoveu Davis e Antônio Miguel Vieira Monteiro: *Introdução à ciência da geoinformação*. <http://197.249.65.74:8080/biblioteca/bitstream/123456789/570/1/publicacao.pdf>, acesso em 2019-04-28. 16, 17, 20
- [30] Ramakrishnan, Raghu e Johannes Gehrke: *Sistemas de gerenciamento de banco de dados-3*. AMGH Editora, 2008. 16
- [31] Barrera, Renato, Andrew Frank e Khaled Al-Taha: *Temporal relations in geographic information systems: A workshop at the university of maine, orono, october 12-13, 1990*. Em *ACM Sigmod Record*, 1991. 17
- [32] Group, The PostgreSQL Global Development: *Postgresql: The world's most advanced open source relational database*. <https://www.postgresql.org>, acesso em 2019-04-28. 17, 18
- [33] PostGis: *Postgis: Spatial and geographic objects for postgresql*. <http://www.postgis.net>, acesso em 2019-04-28. 17
- [34] Politowski, Cristiano e Vinicius Maran: *Comparaç ao de Performance entre PostgreSQL e MongoDB*. X Escola Regional de Banco de Dados. SBC, páginas 1–10, 2014. 17
- [35] Queiroz, Gilberto Ribeiro e Karine Reis Ferreira: *Tutorial sobre Bancos de Dados Geográficos*. Instituto Nacional de Pesquisas Espaciais: São José dos Campos, 2006. 18, 19
- [36] Martinez, JC, E Coll e J Irigoyen: *Exploring postgis with a full analysis example*. Em *Proceedings of the 6th WSEAS international conference on Automation & information, and 6th WSEAS international conference on mathematics and computers in biology and chemistry, and 6th WSEAS international conference on acoustics and music: theory and applications, and 6th WSEAS international conference on Mathematics and computers in business and economics*, páginas 6–11. World Scientific and Engineering Academy and Society (WSEAS), 2005. 18
- [37] Steiniger, Stefan e Andrew JS Hunter: *The 2012 free and open source gis software map—a guide to facilitate research, development, and adoption*. *Computers, environment and urban systems*, 39:136–150, 2013. 18
- [38] Schlossberg, Marc e Elliot Shuford: *Delineating "public" and "participation" in ppgis*. *URISA*, 16(2), 2005. 19

- [39] Sieber, Renee: *Public participation geographic information systems: A literature review and framework*. *Annals of the association of American Geographers*, 96(3):491–507, 2006. 20
- [40] Bugs, Geisa Tamara: *Tecnologias da informação e comunicação, sistemas de informação geográfica e a participação pública no planejamento urbano*. PhD dissertation, Universidade Federal do Rio Grande do Sul, 2014. 20
- [41] Passos, Paulo da Cunha e Maristela Holanda: *Public participation geographic information systems and collaborative mapping of social disorders in brazil*. Em *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, páginas 1–7. IEEE, 2018. 20, 30, 33, 34, 35, 36, 37, 84
- [42] Fielding, Roy T e Richard N Taylor: *Architectural styles and the design of network-based software architectures*, volume 7. University of California, Irvine Doctoral dissertation, 2000. 20
- [43] Saudate, Alexandre: *REST: Construa API's inteligentes de maneira simples*. Editora Casa do Código, 2014. 20
- [44] Richardson, Leonard e Sam Ruby: *RESTful web services*. " O'Reilly Media, Inc.", 2008. 20
- [45] Rodrigues, Lucas e Antonio Francisco do Prado: *Desenvolvimento de Aplicações Móveis com Serviços RESTful e HTML5*. *Revista TIS*, 3(2), 2014. 21
- [46] Čamaj, Juraj, Jaroslav Mašek e Martin Kendra: *Analyzing of customer requirements to eric mobile application*. *Transport technic and technology*, 14(2):15–22, 2018. 21
- [47] Laverty, Joseph Packy, David F Wood, Frederick G Kohun e John Turchek: *Comparative analysis of mobile application development and security models*. *Issues in Information Systems*, 12(1):301–312, 2011. 21
- [48] Lettner, Michael, Michael Tschernuth e Rene Mayrhofer: *Mobile platform architecture review: android, iphone, qt*. Em *International Conference on Computer Aided Systems Theory*, páginas 544–551. Springer, 2011. 21
- [49] statcounter: *Mobile operating system market share worldwide*. <http://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-201201-201903>, acesso em 2019-04-07. 22
- [50] Anvaari, Mohsen e Slinger Jansen: *Evaluating architectural openness in mobile software platforms*. Em *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, páginas 85–92. ACM, 2010. 21
- [51] Souza, Ezequiel Correia de e Lucas Rodrigo Cortinove Cadasqueves: *Aplicativo para atendimento ao cliente de operadora de planos de saúde*. B.S. thesis, Universidade Tecnológica Federal do Paraná, 2018. 21, 25
- [52] Rocha, Adriano Mendonça e Roberto Mendes Finzi Neto: *Introdução a Arquitetura Apple iOS*, 2014. 22, 24, 25

- [53] Inc., Apple: *ios technology overview*. <http://pooh.poly.asu.edu/Mobile/ClassNotes/Papers/MobilePlatforms/iOSTechnicalOverview.pdf>, acesso em 2019-04-07. 22, 23, 24
- [54] Pereira, Marco e Joel Arrais: *Desenvolvimento de um Sistema de Autorizações Médicas com Aplicação Móvel*. B.S. thesis, Universidade de Coimbra, 2016. 22, 25
- [55] Inc., Apple: *Swift*. <https://swift.org>, acesso em 2019-04-07. 25, 26, 27
- [56] García, Cristian González, Jordán Pascual Espada, Begoña Cristina Pelayo García Bustelo e Juan Manuel Cueva Lovelle: *Swift vs. objective-c: A new programming language*. IJIMAI, 3(3):74–81, 2015. 26, 27, 28, 29
- [57] Inc., Apple: *The Swift Programming Language*, volume 1. Apple Inc., 2019. 26, 29
- [58] Inc., Apple: *Swift 1.2 and xcode 6.3 beta*. <https://developer.apple.com/swift/blog/?id=22>, acesso em 2019-04-13. 26
- [59] Inc., Apple: *Swift 2.0*. <https://developer.apple.com/swift/blog/?id=29>, acesso em 2019-04-13. 27
- [60] Inc., Apple: *Swift 3.0 released!* <https://swift.org/blog/swift-3-0-released>, acesso em 2019-04-13. 27
- [61] Inc., Apple: *Swift 4.0 released!* <https://swift.org/blog/swift-4-0-released>, acesso em 2019-04-13. 27
- [62] Inc., Apple: *Swift 5 released!* <https://swift.org/blog/swift-5-released>, acesso em 2019-04-13. 27
- [63] Inc., Apple: *A swift tour*. <https://docs.swift.org/swift-book/GuidedTour/GuidedTour.html>, acesso em 2019-04-13. 28, 29
- [64] Inc., Apple: *A swift tour*. <https://docs.swift.org/swift-book/LanguageGuide/ClassesAndStructures.html>, acesso em 2019-04-13. 28
- [65] WikiCrimes: *WikiCrimes: Mapeando crimes colaborativamente*. <http://wikicrimes.org>, acesso em 2019-04-07. 30
- [66] Furtado, Vasco, Leonardo Ayres, Marcos De Oliveira, Carlos Gustavo e Juliana Oliveira: *Towards semantic wikicrimes*. Em *AAAI Spring Symposium: Social Semantic Web: Where Web 2.0 Meets Web 3.0*, páginas 27–32, 2009. 30, 31
- [67] Furtado, Vasco, L Ayres, JE Vasconcelos, R Alves e M De Oliveira: *WikiCrimes-Um Sistema Colaborativo para Mapeamento Criminal*. Proc. 35th InfoBrasil. Brazil, 2008. 31
- [68] Furtado, Vasco, Marcos de Oliveira, André Fonteles, Cícero Duque, Thiago Assunção, Mairon Belchior, D’Orleans e Jonathan: *Um Método para Identificar Atividades Maliciosas em Sistemas Colaborativos com Mapas*. Em *V Workshop de Algoritmos e Aplicações de Mineração de Dados*, 2009. 31

- [69] Roubado, Onde Fui: *Onde Fui Roubado*. <http://www.ondefuiroubado.com.br>, acesso em 2019-04-07. 31
- [70] Alerta, Equipe UnB: *Colabore com a sua segurança. Permita-nos te ajudar*. <https://unbalerta.unb.br>, acesso em 2019-04-07. 32
- [71] Brito, Edivaldo: *QAP Mobile*. <http://www.techtudo.com.br/tudo-sobre/qap-mobile.html>, acesso em 2019-04-07. 32
- [72] Jesus, Aline: *B.O Coletivo: app de segurança pública*. <https://www.techtudo.com.br/tudo-sobre/bo-coletivo.html>, acesso em 2019-04-30. 32
- [73] Coletivo, B.O: *B.O Coletivo*. <https://www.facebook.com/B.O.Coletivo>, acesso em 2019-04-07. 32
- [74] Roubado, Eu Fui: *Eu Fui Roubado: Monitore o que acontece na sua Cidade. Proteja-se*. <http://eufuiroubado.com>, acesso em 2019-04-07. 33
- [75] Labs, Roga: *IMA - Denuncie*. https://play.google.com/store/apps/details?id=br.com.app.ima.denuncie&hl=pt_BR, acesso em 2019-04-07. 33
- [76] Pontual, Elayne: *Aplicativo de denúncias do IMA ganha atualizações*. <http://www.ima.al.gov.br/aplicativo-de-denuncias-do-ima-ganha-atualizacoes>, acesso em 2019-04-07. 33
- [77] Crimes, Reporting: *Reporting Crimes*. <http://reportingcrimes.com/br/sao-paulo/sp>, acesso em 2018-01-20. 33
- [78] Federal, Distrito: *Viva Brasília – Nosso Pacto pela Vida*. <http://vivabrasilia.ssp.df.gov.br>, acesso em 2019-05-13. 39
- [79] Federal, Distrito: *Documento Orientador do Viva Brasília – Nosso Pacto pela Vida*. <http://www.adaltech.com.br/anais/sociologia2017/resumos/PDF-eposter-trab-aceito-1000-1.pdf>, acesso em 2019-05-13. 39
- [80] Informação, Subsecretaria de Gestão da: *Sistema Integrado de Geoestatística*. <https://sigeo.ssp.df.gov.br>, acesso em 2019-05-13. 40
- [81] Sandi, Gitarja, I Gusti Bagus Baskara Nugraha e Suhono Harso Supangkat: *Mobile health monitoring and consultation to support hypertension treatment*. Em *International Conference on ICT for Smart Society*, páginas 1–5. IEEE, 2013. 42
- [82] Bento, Miguel José Nogueira: *App Coudelaria Militar*. B.S. thesis, Atlântica - University Higher Institution, 2019. 43
- [83] OpenStreetMap, Fundação: *OpenStreetMap*. <https://www.openstreetmap.org>, acesso em 2019-05-17. 44
- [84] *Semantic UI: Interface do usuário é a linguagem da web*. <https://semantic-ui.com>, acesso em 2019-05-17. 44
- [85] *Jquery: write less, do more*. <https://jquery.com>, acesso em 2019-05-17. 44

- [86] *Leaflet: an open-source javascript library for mobile-friendly interactive maps*. <https://leafletjs.com>, acesso em 2019-05-17. 44
- [87] Lourenço, Alon Mota: *ToDeOlho para Android: Um Software Para Mapeamento Colaborativo de Desordens Socais por Meio de Sistemas Georreferenciados*. Monografia (Bacharel em Engenharia da Computação), Universidade de Brasília, Brasília, Brazil, 2018. 44, 45
- [88] Crawford, Tyler e Tauqeer Hussain: *A comparison of server side scripting technologies*. Em *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)*, páginas 69–76. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2017. 44
- [89] Pereira, Caio Ribeiro: *Node.js: Aplicações web real-time com Node.js*. Editora Casa do Código, 2014. 44, 45
- [90] *Multer*. <https://www.npmjs.com/package/multer>, acesso em 2019-05-19. 45
- [91] *Knex.js*. <https://knexjs.org>, acesso em 2019-05-19. 45
- [92] Macário, Carla Geovana do N e Stefano Monteiro Baldo: *O modelo relacional*. Campinas, SP: Embrapa Informática Agropecuária, 2005. 46
- [93] *Wath is heroku?* <https://www.heroku.com>, acesso em 2019-05-21. 51
- [94] *Computação em nuvem com a aws*. <https://aws.amazon.com/pt/what-is-aws/>, acesso em 2019-05-21. 52
- [95] Sommerville, Ian: *Engenharia de Software*. Pearson, 2011. 53
- [96] Beck, Kent, W Cunningham, A Hunt, R Martin, D Thomas, M Beedle e J Sutherland: *Manifesto ágil*. Manifesto para Desenvolvimento Ágil de Software, 2001. 53
- [97] Santos Soares, Michel dos: *Metodologias ágeis extreme programming e scrum para o desenvolvimento de software*. Revista Eletrônica de Sistemas de Informação, 3(1), 2004. 53
- [98] Staa, Arndt Von: *Programação Modular*. Desenvolvendo programas completos de forma organizada e segura, 2000, ISBN 8535206086. 55
- [99] Inc., Apple: *Xcode overview*. https://developer.apple.com/library/archive/documentation/ToolsLanguages/Conceptual/Xcode_Overview/index.html#//apple_ref/doc/uid/TP40010215-CH24-SW1, acesso em 2019-05-21. 56
- [100] Inc., Apple: *Framework uikit*. <https://developer.apple.com/documentation/uikit>, acesso em 2019-05-21. 57
- [101] *Alamofire*. <https://github.com/Alamofire/Alamofire>, acesso em 2019-05-21. 57
- [102] Inc., Apple: *Frameworkmapkit*. <https://developer.apple.com/documentation/mapkit>, acesso em 2019-05-21. 57

- [103] Delamaro, Marcio, Mario Jino e Jose Maldonado: *Introdução ao teste de software*. Elsevier Brasil, 2017. 65
- [104] Castro, Ana Karina Silva de: *Testes Exploratórios: Características, Problemas e Soluções*. B.S. thesis, Universidade Federal do Rio Grande do Norte, 2018. 66

Apêndice A

Formulário de Usabilidade do Aplicativo Tô de Olho no DF

1. Endereço de e-mail *



Seja bem vindo ao questionário sobre o teste de usabilidade do aplicativo Tô de Olho.

O aplicativo Tô de Olho faz parte de um projeto para criação de um sistema de informação georreferenciado, montado em uma arquitetura cliente e servidor, que permite a participação população no cadastro e mapeamento de desordens urbanas e sociais. Foi desenvolvido para a plataforma iOS, e visa auxiliar a Secretaria de Estado da Segurança Pública e da Paz Social do Distrito Federal a mapear desordens sociais e urbanas no Distrito Federal. Sendo que o projeto recebeu o apoio da Fundação de Apoio a Pesquisa do Distrito Federal (FAPDF), prot. nº 19781.82.31496.17082017.

O presente questionário tem por objetivo identificar defeitos e analisar a percepção dos usuários acerca da: usabilidade, utilidade, e interface do Tô de Olho no DF.

A aplicativo se encontra disponível na Apple Store, no seguinte link:

<https://apps.apple.com/br/app/t%C3%B4-de-olho-no-df/id1468199780>

Para responder este questionário é recomendado que instale a aplicação e tente realizar as seguintes ações:

- 1 - Criar uma conta.
- 2 - Cadastrar uma desordem ou problema social.
- 3 - Navegar no mapa de desordens (mexer o mapa, aumentar e diminuir o zoom do mapa)

Caracterização de Perfil do Usuário

2. Qual o seu grau de escolaridade?

Marcar apenas uma oval.

- Fundamental incompleto
- Fundamental completo
- Médio incompleto
- Médio completo
- Superior incompleto
- Superior completo

3. Qual a sua familiaridade com Smartphones?

Marcar apenas uma oval.

	0	1	2	3	4	5	
Não uso	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Uso frequentemente

4. Faz uso de aplicativos de Smartphones? Se sim, aponte em média em quantos dias da semana você utiliza aplicativos no Smartphone?

Escolha uma das alternativas de forma que 0 significada nenhum dia da semana e 7 significa todos os dias da semana.

Marcar apenas uma oval.

0	1	2	3	4	5	6	7
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5. Já utilizou algum aplicativo de colaboração popular, onde as pessoas forneçam dados de forma participativa?

Marcar apenas uma oval.

- Sim
- Não

6. Já utilizou algum aplicativo de mapeamento colaborativo?

Marcar apenas uma oval.

- Sim
- Não

7. Já utilizou algum aplicativo semelhante ao Tô de Olho?

Marcar apenas uma oval.

- Sim
- Não

8. **Como você avalia seu nível de participação em redes sociais ou em aplicativos colaboração popular?**

Escolha uma das alternativas de forma que 0 significada você não é nada participativo e 5 que você é muito participativo.

Marcar apenas uma oval.

0	1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

9. **Como você classificaria a ordem/conservação do bairro onde mora?**

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	8	9	10	
Moro em um bairro precário com muita desordem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Moro em um bairro limpo e bem conservado

10. **Você estaria disposto a relatar as desordens presentes no seu bairro em uma plataforma de mapeamento colaborativo, com intuito de auxiliar os órgãos da administração pública?**

Marcar apenas uma oval.

- Sim
 Não

Percepção dos usuários sobre o ToDeOlho

Nessa seção será avaliada o Aplicativo Tô de Olho

11. **Qual o modelo de iPhone você utilizou para testar o Tô de Olho?**

Marcar apenas uma oval.

- iPhone 4
- iPhone 4S
- iPhone 5
- iPhone 5C
- iPhone 5S
- iPhone 6
- iPhone 6 Plus
- iPhone 6S
- iPhone 6S Plus
- iPhone SE
- iPhone 7
- iPhone 7 Plus
- iPhone 8
- iPhone 8 Plus
- iPhone X
- iPhone XS
- iPhone XS Max
- iPhone XR

12. **Você conseguiu se cadastrar se no aplicativo?**

Marcar apenas uma oval.

- Sim
- Não

13. **Qual foi a sua primeira impressão sobre a aplicação?**

14. **Você considera as cores e botões do aplicativo agradáveis?**

Marcar apenas uma oval.

- Sim
- Não

15. **Você conseguiu visualizar bem todos os botões e informações dentro do aplicativo?**

Marcar apenas uma oval.

- Sim
- Não

16. **Você conseguiu entender com facilidade as palavras, nomenclaturas e ícones do aplicativo?**

Marcar apenas uma oval.

Sim

Não

17. **Você conseguiu visualizar o mapa com os marcadores?**

Marcar apenas uma oval.

Sim

Não

18. **Você conseguiu compreender o conteúdo que era apresentado no mapa?**

Marcar apenas uma oval.

Sim

Não

19. **Você conseguiu acessar a lista das doenças cadastradas?**

Marcar apenas uma oval.

Sim

Não

20. **Você conseguiu ver os detalhes de uma doença em particular?**

Marcar apenas uma oval.

Sim

Não

21. **Você conseguiu efetuar o login com a conta cadastrada?**

Marcar apenas uma oval.

Sim

Não

22. **Você conseguiu cadastrar uma nova denúncia?**

Marcar apenas uma oval.

Sim

Não

23. **Você conseguiu mover o mapa da aplicação e ver outras denúncias conforme alterava o zoom do mapa?**

Marcar apenas uma oval.

Sim

Não

24. **Você considera que o aplicativo Tô de Olho pode ajudar a administração pública a mapear e solucionar desordens sociais?**

Escolha uma das alternativas de forma que 0 significada que você não considera o aplicativo nada útil e 5 que é totalmente útil

Marcar apenas uma oval.

0 1 2 3 4 5

25. **Na sua opinião, o que poderia ser feito para melhorar o ToDeOlho?**

Envie para mim uma cópia das minhas respostas.

Powered by

 **Google Forms**