



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

FISCAL DA CIDADE: UMA APLICAÇÃO DE APOIO AO CIDADÃO COM
OBJETIVO DE REPORTAR PROBLEMAS URBANOS

Filipe Esteves Gonçalves e Leonardo Felipe Menezes Dos Santos

Orientador

Pedro Nuno De Souza Moura

RIO DE JANEIRO, RJ – BRASIL

DEZEMBRO DE 2018

Catálogo informatizada pelo(a) autor(a)

E Esteves Gonçalves, Filipe
FISCAL DA CIDADE: UMA APLICAÇÃO DE APOIO AO
CIDADÃO COM OBJETIVO DE REPORTAR PROBLEMAS URBANOS
/ Filipe Esteves Gonçalves. -- Rio de Janeiro, 2018.
55

Orientador: Pedro Nuno de Souza Moura.
Trabalho de Conclusão de Curso (Graduação) -
Universidade Federal do Estado do Rio de Janeiro,
Graduação em Sistemas de Informação, 2018.

1. Geolocalização. 2. React Native. 3. Node.js.
I. Nuno de Souza Moura, Pedro, orient. II. Título.

M Menezes dos Santos, Leonardo Felipe
FISCAL DA CIDADE: UMA APLICAÇÃO DE APOIO AO
CIDADÃO COM OBJETIVO DE REPORTAR PROBLEMAS URBANOS
/ Leonardo Felipe Menezes dos Santos. -- Rio de
Janeiro, 2018.
55

Orientador: Pedro Nuno De Souza Moura.
Trabalho de Conclusão de Curso (Graduação) -
Universidade Federal do Estado do Rio de Janeiro,
Graduação em Sistemas de Informação, 2018.

1. Geolocalização. 2. React Native. 3. Node.js.
I. Nuno De Souza Moura, Pedro, orient. II. Título.

FISCAL DA CIDADE: UMA APLICAÇÃO DE APOIO AO CIDADÃO COM
OBJETIVO DE REPORTAR PROBLEMAS URBANOS

Filipe Esteves Gonçalves e Leonardo Felipe Menezes Dos Santos

Projeto de Graduação apresentado à Escola de
Informática Aplicada da Universidade Federal do
Estado do Rio de Janeiro (UNIRIO) para obtenção do
título de Bacharel em Sistemas de Informação.

Aprovado por:

PEDRO NUNO DE SOUZA MOURA (UNIRIO)

LEONARDO LUIZ ALENCASTRO ROCHA (UNIRIO)

MORGANNA CARMEM DINIZ (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

DEZEMBRO DE 2018

Agradecimentos

Gostaríamos de agradecer aos nossos familiares por todo apoio, não apenas durante esse trabalho de conclusão de curso ou durante a graduação, mas durante todo o trajeto que nos trouxe até aqui, com altos e baixos e muito aprendizado. Também aos nossos amigos e colegas de curso que nos ajudaram e caminharam juntos, os quais tornaram alguns momentos que deveriam ser mais maçantes e monótonos, descontraídos de alguma forma.

Um agradecimento especial ao nosso professor e orientador Pedro Moura, que apesar do nosso cronograma apertado sempre se mostrou muito solícito a nos encontrar e tirar dúvidas, acompanhando e também cobrando nosso progresso.

Stack Overflow, aquele abraço!

RESUMO

Atualmente, o uso de aplicações para dispositivos móveis que possuem geolocalização tem crescido em importância e quantidade de usuários. Tendo isso por motivação e visando à disseminação de informação sobre as dificuldades do dia a dia do cidadão nas cidades e a atuação dos governos sobre as mesmas, foi desenvolvido um protótipo de sistema colaborativo para que seus usuários reportem e compartilhem reclamações georreferenciadas sobre problemas como postes sem luz, lixo não coletado, entre outros de interesse coletivo e cotidiano das cidades.

Palavras-chave: Geolocalização, React Native, Node.js .

ABSTRACT

Currently, the use of mobile applications that have geolocation has grown in importance and quantity of users, for this reason and believing that there is still a lot of information and knowledge to be disseminated among citizens and also with the government, we have developed a prototype of a collaborative system so that its users share georeferenced complaints about problems of collective and daily interest of the city as a broken light pole or waste not collected.

Keywords: Geolocation, React Native, Node.js

Índice

1	Introdução	9
	Motivação	9
	Objetivos	11
	Organização do texto	12
2	Revisão Bibliográfica	13
	2.1. Sistemas Colaborativos	13
	2.1.1 Sistemas Colaborativos e suas classificações.	14
	2.2 Sistemas que apoiam a solução de problemas urbanos utilizando a geolocalização.	14
	2.2.1 SUL (Sistema Útil de Localização)	14
	2.2.2 Sistema de Apoio à Mobilidade Urbana com Dados Abertos.	15
	2.2.3 Tráfego	16
	2.2.4 OTT (Onde Tem Tiroteio)	17
	2.2.5 Colab	18
3	Tecnologias Adotadas	20
	3.1 A aplicação	20
	3.2 React Native	21
	3.3 Mapbox	22
	3.4 Node.js	22
	3.5 MongoDB	24
	3.6 Heroku	25
4	Fiscal da Cidade	26
	4.1 Login	26
	4.2 Core da aplicação	30
	4.2.1 Criar registro	31
	4.2.2 Descrição da localidade	36
	4.3 Informações do usuário	39
	4.4 Link para o instalador da aplicação	41
5	Especificações do Sistema	42
	5.1 Diagrama de Classes	42
	5.2 Casos de Uso	43
6	Conclusão	51
	6.1 Considerações finais	51
	6.2 Trabalhos futuros	51

Índice de Figuras

Figura 1.	Telas do Sistema Útil de Localização.....	15
Figura 2.	Tela do Sistema de Apoio à Mobilidade Urbana com Dados Abertos...	16
Figura 3.	Telas do aplicativo Trafi	17
Figura 4.	Telas do aplicativo Onde Tem Tiroteio.....	18
Figura 5.	Telas do aplicativo Colab.....	19
Figura 6.	Arquitetura do sistema.....	20
Figura 7.	Tela de Login.....	26
Figura 8.	Usuário clica em criar conta e preenche os dados.....	27
Figura 9.	Usuário cria a conta com sucesso.....	28
Figura 10.	Usuário pode então logar na aplicação.....	29
Figura 11.	Tela principal da aplicação.....	30
Figura 12.	Usuário abre formulário de descrição do reporte ao clicar para marcar o ponto.....	31
Figura 13.	Usuário seleciona tipo do registro.....	32
Figura 14.	Usuário preenche descrição do registro.....	33
Figura 15.	Usuário pode adicionar foto.....	34
Figura 16.	Usuário escolhe se deseja manter a foto ou excluir.....	35
Figura 17.	Registro criado após usuário clicar em salvar.....	36
Figura 18.	Usuário clica em alguma marcação do mapa, visualiza a descrição dela e apoia o conteúdo como verdadeiro com um “like”	37
Figura 19.	Opção de excluir localidade é habilitada quando o criador daquele registro é o visualizador.....	38
Figura 20.	É necessário que o usuário criador da localidade confirme a ação para excluí-la.....	39
Figura 21.	Menu lateral exibe informações do usuário.....	40
Figura 22.	QR Code para o download do apk do aplicativo.....	41

Figura 23.	Diagrama de Classes da aplicação.....	42
Figura 24.	Visão geral dos casos de uso.....	43
Figura 25.	Caso de uso: Criar Cadastro.....	44
Figura 26.	Caso de uso: Fazer Login.....	45
Figura 27.	Caso de uso: Fazer Logout.....	46
Figura 28.	Caso de uso: Navegar pelo Mapa.....	47
Figura 29.	Caso de uso: Visualizar registro.....	48
Figura 30.	Caso de uso: Reportar Problema.....	49
Figura 31.	Caso de uso: Excluir Registro de Problema.....	50

1 Introdução

1.1 Motivação

Os *smartphones*, aparelhos celulares com capacidades semelhantes à de um computador e de acesso à internet tanto por conexões de redes locais (WLAN) como *wi-fi*, quanto por redes de longa distância (WWAN), atualmente já são uma realidade no cotidiano dos centros urbanos do Brasil e no mundo, e vem ganhando mais força, conforme vão se aprimorando as tecnologias para dispositivos físicos, como a melhora nos serviços de internet de telecom e seu aumento nas áreas de cobertura. Já é possível observar que existem mais aparelhos ativos do que habitantes no país. O Brasil superou a marca de um smartphone por habitante e hoje conta com 220 milhões de celulares inteligentes ativos, de acordo com a 29ª Pesquisa Anual de Administração e Uso de Tecnologia da Informação nas Empresas [1].

E dentro da gama de utilização dos dispositivos móveis ligados à internet por seus usuários podemos listar diversas finalidades entre as quais estão a fonte de conhecimento tanto para o cotidiano, através do consumo de notícias, quanto para o aprofundamento em conhecimento técnicos nas mais diversas áreas do saber, além da busca da informação como um fim em si. Se destacam também o entretenimento pelo emprego dos serviços *over-the-top*, como redes sociais, serviços de *stream* de música e áudio visual e o aproveitamento para o trabalho, onde se podem fazer tarefas remotas colaborativamente, que anteriormente não eram possíveis de serem realizadas de forma tão eficiente e instantânea.

Toda a interatividade e colaboração proveniente do uso combinado entre dispositivos móveis e troca de informação, além de incentivar os cidadãos a despender seu tempo em temas como diversão, trabalho e busca por instrução, também contribui na busca por melhoria na qualidade de vida, seja pela busca na otimização dos gastos

com movimentos pendulares entre a residência e o trabalho e/ou estudo, com a utilização de aplicações geolocalizadas, tanto por transporte público (Moovit e CittaMobi) como consumo de transporte privado (Uber e 99) ou veículos próprios (Google maps e waze). Assim como na busca por transparência de órgãos públicos no que concerne às contas públicas. Outros exemplos correspondem à busca por melhoria nos serviços de empresas tanto pela capacidade de divulgação de problemas, e reclamações públicas, como pela avaliação quantitativas (número de downloads de uma aplicação) e qualitativas (comentários acerca de um serviço ou de um produto consumido).

A ampla variedade de serviços prestados de grande aceitação entre os usuários que fazem uso de ferramentas de geolocalização por GPS (*Global Positioning System*) e de imagens dá amparo para que novas soluções que se utilizam desse ferramental sejam aplicados em outras soluções que favoreçam o convívio urbano e a qualidade de vida da população.

Ao mesmo tempo que vemos aplicações que dão conta de diminuir o impacto causado por problemas sociais urbanos, por outro lado, ainda existem tantos outros desafios em que uma abordagem tecnológica pode contribuir para resolver ou ao menos amenizar tanto a falta de conhecimento conclusivo sobre nossos problemas relacionados aos serviços básicos oferecidos, pelo estado e por concessões públicas, como buracos em ruas, vazamento de esgoto e falta de saneamento, falta de iluminação, falta de coleta de lixo. Eliminação de resíduos feitos de forma irregular ou em lugares inadequados.

A Central de Atendimento 1746 do Rio de Janeiro é um exemplo de canal de ouvidoria da cidade para escutar os cidadãos e é justamente daí que se podem encontrar mais facilmente algumas informações estatísticas sobre serviços de remoção de entulho, remoção de roedores, reparo de poste de luz, ônibus, vias públicas, pavimentação entre diversos outros serviços. Porém até mesmo o serviço de ouvidoria da cidade é alvo de matérias jornalísticas sobre os serviços prestados de forma insatisfatória [2]. E entre os dados apresentados pela própria ouvidoria 32% dos cidadãos estão insatisfeitos com as informações fornecidas, 31% insatisfeitos com os serviços prestados, porém 94% estão satisfeitos com com a central de atendimento 1746. Esses números fornecidos, que dão conta de todos os dados coletados até 2018, num total de mais de 12 milhões de

solicitações recebidas, diz que somente 42% das demandas não são atendidas. E ainda assim esses números não são condizentes com pesquisas sobre enchentes e deslizamentos que vitimizam cidadãos todos os anos no país. Se olharmos o rio de janeiro, como exemplo, aqui mais de 400 mil pessoas vivem em áreas de risco relacionadas às enchentes e deslizamentos [3].

1.2 Objetivos

Estudos internacionais dão conta do baixo conhecimento do brasileiro sobre sua própria realidade [4].

Esse baixo conhecimento sobre si se dá muito pela falta de um plano estrutural de políticas públicas para superação do baixo investimento em educação e conhecimento, tanto em estrutura, em divulgação e desenvolvimento humano, que são históricos. O baixo grau de estudo da população como um todo, portanto, prejudica na produção desse discernimento e também no interesse e capacidade de absorver e fomentar o conhecimento produzido. Esse conjunto de fatores atrapalha na proposição de soluções criativas ou não para que sejam descobertas e supridas as necessidades do cidadão de forma célere, transparente e que transformem em melhoria da qualidade de vida.

Neste trabalho, desenvolvemos um protótipo de aplicação móvel com foco em dispositivos *android*, mas utilizando uma tecnologia que permite o desenvolvimento nativo também para aplicações em dispositivos IOS. A aplicação nativa consome uma API *rest* de serviço em nuvem também desenvolvida por nós. O objetivo da aplicação é possibilitar aos usuários que reivindiquem por condições melhores para manutenção do estado das coisas públicas dos locais em que transitam assim como também deem publicidade e disseminem as condições de seu cotidiano, através da submissão de registros com capacidade de registrar imagens pelo denunciante, de forma que fiquem registradas em histórico, localizadas espacialmente, e acessíveis por outros usuários por tanto tempo quanto for necessário para que se seja dada uma solução satisfatória. Por fim, também é possível que se avalie coletivamente a gravidade da reivindicação pela interação por reações aos registros disponibilizados pelos cidadãos.

1.3 Organização do texto

O presente trabalho está estruturado em capítulos e, além desta introdução, será desenvolvido da seguinte forma:

Capítulo 2: No capítulo de revisão bibliográfica serão abordadas as discussões acadêmicas e jornalísticas que fundamentam tanto os temas ligados aos trabalhos de sistemas colaborativos e de geolocalização aplicados na atualidade quanto ao tema das dificuldades encontradas nas cidades pelos cidadãos que possam estar relacionados à aplicação desenvolvida.

Capítulo III: Tecnologias adotadas - Enumeradas, justificadas e explicadas as tecnologias e soluções encontradas.

Capítulo IV: Descrição da ferramenta - Reúne documentos e exemplos das funcionalidades da ferramenta desenvolvida.

Capítulo V: Especificações do Sistema - Reúne os diagramas de classe e casos de uso do sistema.

Capítulo VI: Conclusões - Reúne as considerações finais, assinala as contribuições da pesquisa e sugere possibilidades de aprofundamento posterior.

2 Revisão Bibliográfica

Neste capítulo serão apresentados alguns trabalhos relacionados ao tema abordado neste projeto. Primeiramente, será apresentado o conceito de Sistemas Colaborativos e serão comentados alguns trabalhos oriundos dessa área. Posteriormente, serão expostos exemplos de aplicações similares que utilizam a geolocalização como ferramenta para solucionar problemas urbanos e do cotidiano.

2.1. Sistemas Colaborativos

“Sistemas Colaborativos são ferramentas de software utilizadas em redes de computadores para facilitar a execução de trabalhos em grupos. Essas ferramentas devem ser especializadas o bastante, a fim de oferecer aos seus usuários formas de interação, facilitando o controle, a coordenação, a colaboração e a comunicação entre as partes envolvidas que compõe o grupo, tanto no mesmo local, como em locais geograficamente diferentes e que as formas de interação aconteçam tanto ao mesmo tempo ou em tempos diferentes. Percebe-se com isso que o objetivo dos Sistemas Colaborativos é diminuir as barreiras impostas pelo espaço físico e o tempo” [5].

Diversos termos denominam Sistemas Colaborativos, alguns deles são “*Grupoware*”, “Sistemas *Workflow*”, “CSCW” (*Computer Supported Cooperative Work* - Trabalho Cooperativo Apoiado por Computador), “*Online Collaboration*”, “*Web Collaboration*”, “*Collaboration tools*”, dentre outros. Mas a ideia fundamental continua sendo a mesma, prover suporte ao desenvolvimento de trabalho colaborativamente.

2.1.1 Sistemas Colaborativos e suas classificações.

As ferramentas de colaboração (sistemas colaborativos) são classificadas de acordo o tempo [6]:

- a) Colaboração Síncrona: são aquelas que requerem tempo de resposta imediato, tais como videoconferência, aplicações de chat (Whatsapp, Messenger), ferramentas de controle de versão, jogos *multiplayer* e etc.
- b) Assíncrona: não necessitam de um tempo de resposta imediato: e-mail, fóruns de discussão, calendários de grupo, sistemas de escrita colaborativa (Essas podem ser síncronas, como exemplo, Google Docs ou assíncronas, como dontpad.com).

2.2 Sistemas que apoiam a solução de problemas urbanos utilizando a geolocalização.

2.2.1 SUL (Sistema Útil de Localização)

No trabalho do Sistema útil de Localização [7] foi proposto um sistema que funcione como gestor de rotas para a parte interna de estruturas como prédios, shoppings e estádios. Tendo como objetivo facilitar o deslocamento de pessoas dentro dessas estruturas fechadas.

O aplicativo(Figura 1) funciona da seguinte forma: são espalhados QR Codes por pontos de referência específicos da estrutura interna. O usuário poderá traçar a rota escolhendo o ponto de partida numa lista ou via QR Code, e o destino em outra lista. Após a escolha do destino, a rota é apresentada ao usuário.



Figura 1 - Telas do Sistema Útil de Localização

2.2.2 Sistema de Apoio à Mobilidade Urbana com Dados Abertos.

Esse trabalho de Sistema de apoio à Mobilidade Urbana com Dados Abertos [8] utilizou o portal de dados abertos da Prefeitura do Rio, que disponibiliza informações sobre transporte, mobilidade, educação, turismo, meio ambiente, saúde entre outros. Usando os dados dos GPS instalados nos ônibus que fazem parte da frota da Cidade do Rio de Janeiro, pontos de parada das linhas de ônibus e pontos dos trajetos das linhas de ônibus.

Ele tem como objetivo auxiliar os usuários a encontrar as linhas de ônibus que os levem ao seu ponto de destino e ainda informar o tempo que o ônibus vai levar para chegar até o ponto de partida. Mas devido a problemas técnicos, o objetivo inicial foi revisto e reduzido para apenas resolver o problema dos itinerários (Figura 2).

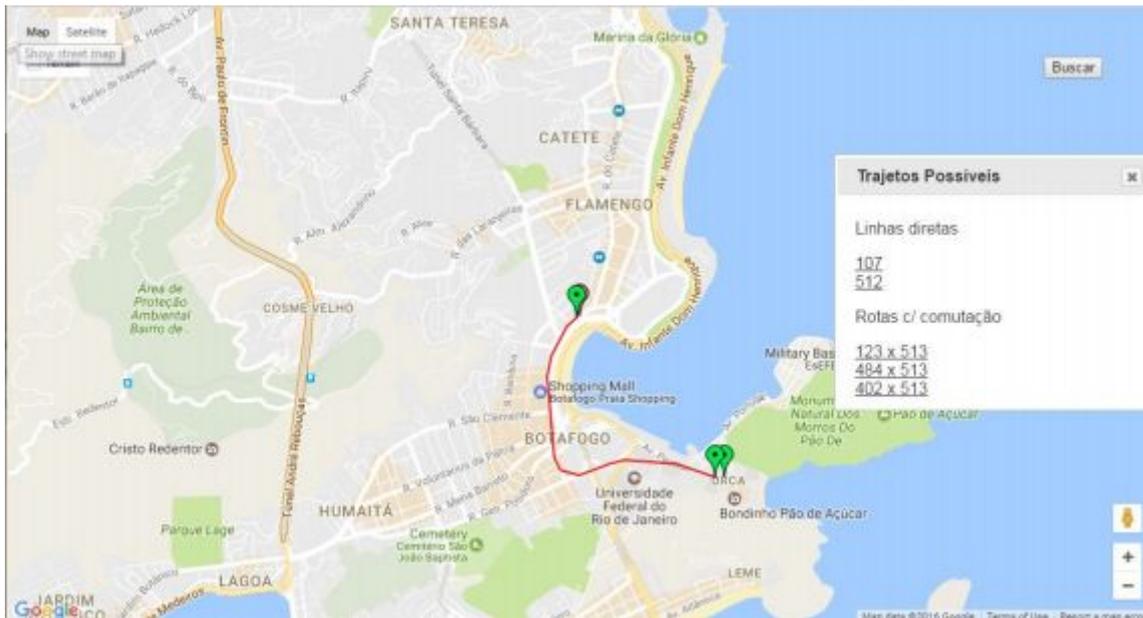


Figura 2 - Tela do Sistema de Apoio à Mobilidade Urbana com Dados Abertos

2.2.3 Trafi

Assim como o objetivo do trabalho anterior sobre mobilidade urbana com dados abertos, o aplicativo Trafi [9] conecta e compara as opções de mobilidade da cidade e exibe os detalhes em tempo real.

O aplicativo(Figura 3) fornece algumas funcionalidades que facilitam a mobilidade urbana, tendo como uma das que mais chamam atenção a previsão e visualização em tempo real dos ônibus rodando no momento e sua localização via GPS. Fazendo com que o usuário consiga se planejar para sair de casa e ficar o menor tempo possível no ponto de ônibus, tendo assim benefícios como ficar menos tempo exposto ao clima (chuva, sol, etc), e mais segurança por ficar menos tempo exposto aos riscos relacionados à segurança (roubo, sequestro relâmpago, etc).

Trafi I>>

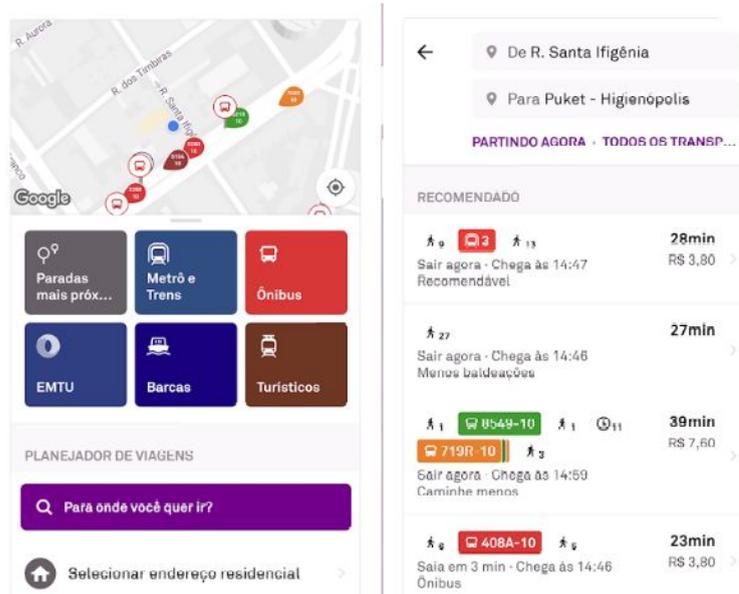


Figura 3 - Telas do aplicativo Trafi

2.2.4 OTT (Onde Tem Tiroteio)

“Fundada em Janeiro de 2016 por quatro amigos preocupados com o crescimento descontrolado da violência no estado do Rio de Janeiro, a OTT-Brasil tem como principal missão retirar todos os cidadãos das rotas dos arrastões, das falsas blitz e das balas perdidas, com informações que são colhidas, analisadas e divulgadas num curtíssimo espaço de tempo.

O conceito SP 4.0, Segurança Pública 4.0, que norteia a nossa dinâmica de trabalho, se baseia na segurança feita do cidadão para o cidadão (C2C), uma espécie de segurança “Smart”, onde cada cidadão atualiza em tempo real a segurança em seu entorno, ajudando a ele e a todos os outros participantes de nossa rede dinâmica de informações.” [10]

O aplicativo(Figura 4) tem como objetivo informar ao usuário tiroteios na sua proximidade para que ele possa escolher o melhor caminho a seguir, buscando evitar passar em regiões de assalto ou tiroteio. Alguns exemplos de notificação correspondem a :“OTT Informa: tiros na Cidade de Deus” e “OTT Informa: Carros voltando na contramão na Avenida Brasil sentido Centro”.

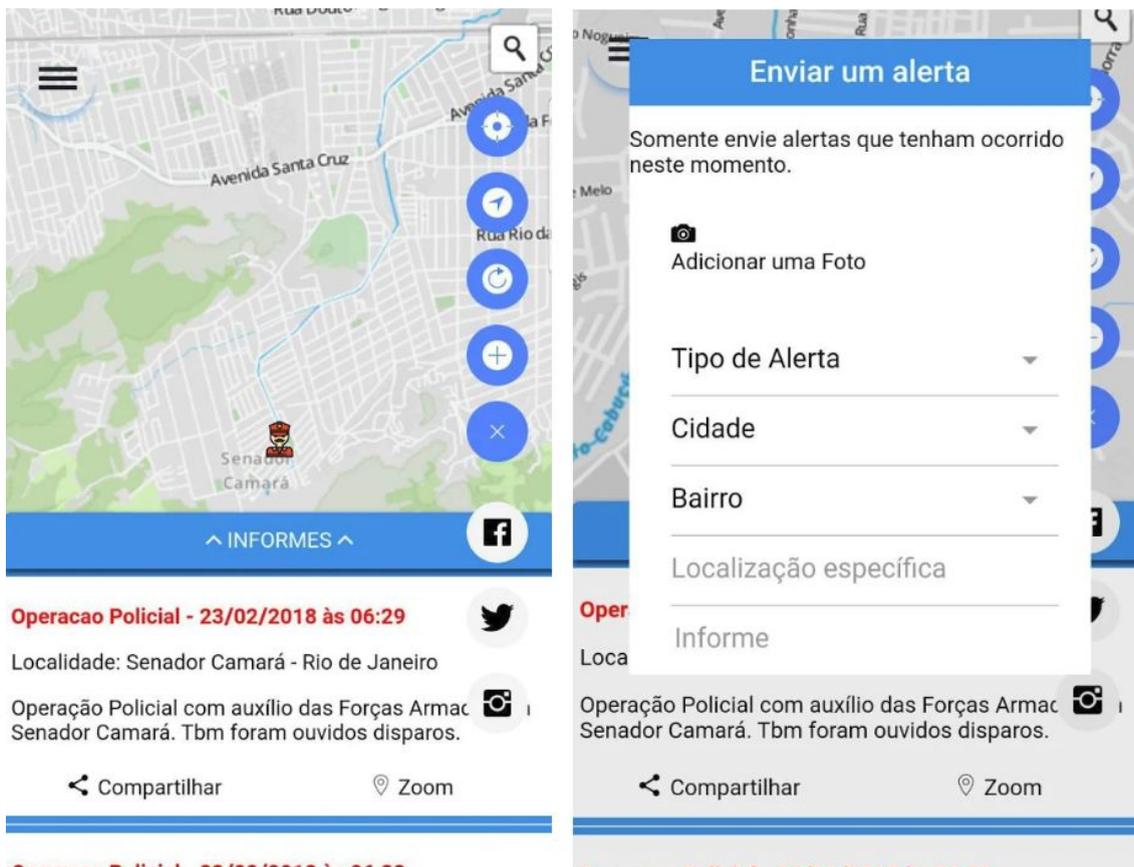


Figura 4 - Telas do aplicativo Onde Tem Tiroteio

2.2.5 Colab

Durante o desenvolvimento deste trabalho, encontramos um aplicativo(Figura 5) que segue os nossos objetivos. Ele foi criado por uma *startup* de gestão colaborativa que trabalha para criar pontes entre cidadãos e governos, a Colab [11].

“Com 200 mil usuários no Brasil, a empresa mantém uma rede social onde é possível publicar sugestões ou pedidos de soluções sobre problemas como falta de iluminação, buracos nas estradas e ruas e estações de metrô e ônibus mal cuidadas.

Quando a Prefeitura da cidade participa da Colab, as demandas são enviadas diretamente para os órgãos e servidores competentes. Os Executivos municipais também recebem materiais e oficinas sobre como incluir a participação dos cidadãos na gestão pública.

Com a tecnologia e metodologia da *startup*, algumas prefeituras aumentaram significativamente seus índices de atendimento às solicitações da população. Em Teresina, a resolução de demandas subiu de 39% em 2016 para atuais 74%.” [12]

A rede social Colab tem a proposta de ser uma ponte de comunicação entre o cidadão e a prefeitura da sua cidade. Tornando fácil o registro de irregularidades urbanas como buraco na calçada, nas vias, estacionamento irregular, ausência de iluminação, dentre outras categorias. As prefeituras das cidades podem se conectar à plataforma para conhecer quais os problemas que mais incomodam a população ou permitir que os usuários participem de tomadas de decisão que os mais beneficiem.

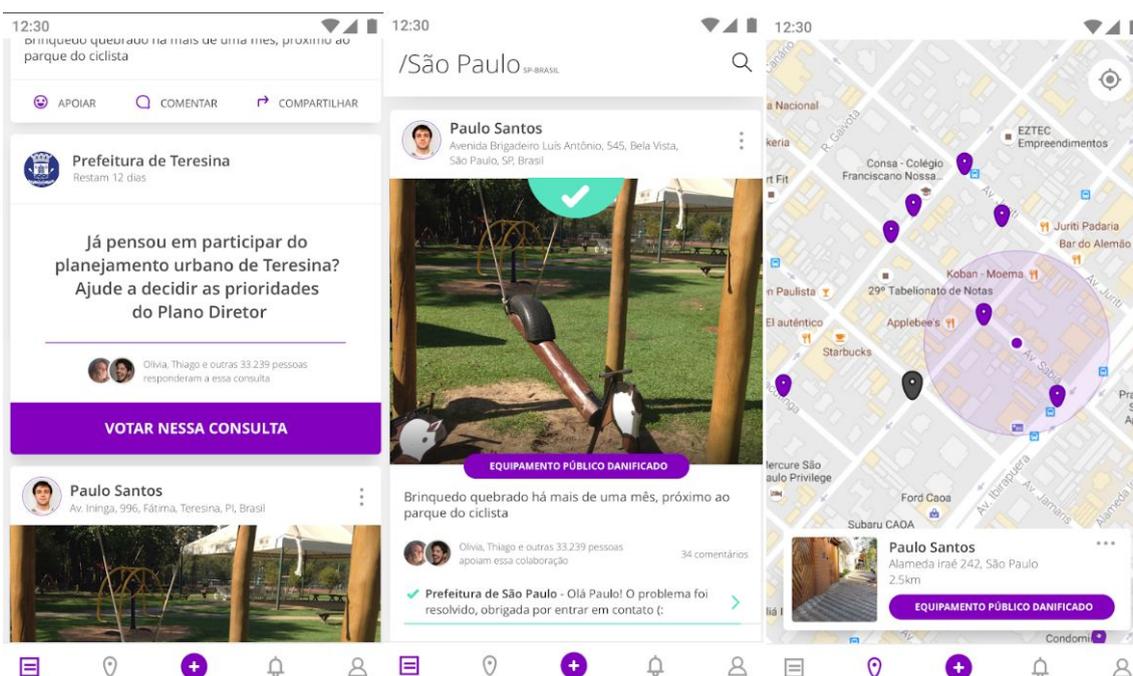


Figura 5 - Telas do aplicativo Colab

3 Tecnologias Adotadas

Neste capítulo, serão descritas as tecnologias utilizadas no desenvolvimento do aplicativo proposto.

3.1 A aplicação

As tecnologias adotadas para a aplicação de fiscalização da cidade pelo cidadão podem ser dividida em dois sistemas, um sistema móvel desenvolvido em React Native/Mapbox que consome os serviços REST (*Representational State Transfer*) do segundo sistema desenvolvido em Node.js que faz a persistência de dados através do SGBD (Sistema de Gerenciamento de Banco de Dados) MongoDB, assim como também disponibiliza os *endpoints* das API (*Application Programming Interface* - Interface de Programação de Aplicações). (Figura 6)

A primeira aplicação citada acima, sendo uma aplicação móvel nativa deve ser instalada em um sistema *android*, já a segunda aplicação está hospedada no Heroku, um serviço de hospedagem para aplicações em nuvem, a ser descrito na Seção 3.6.

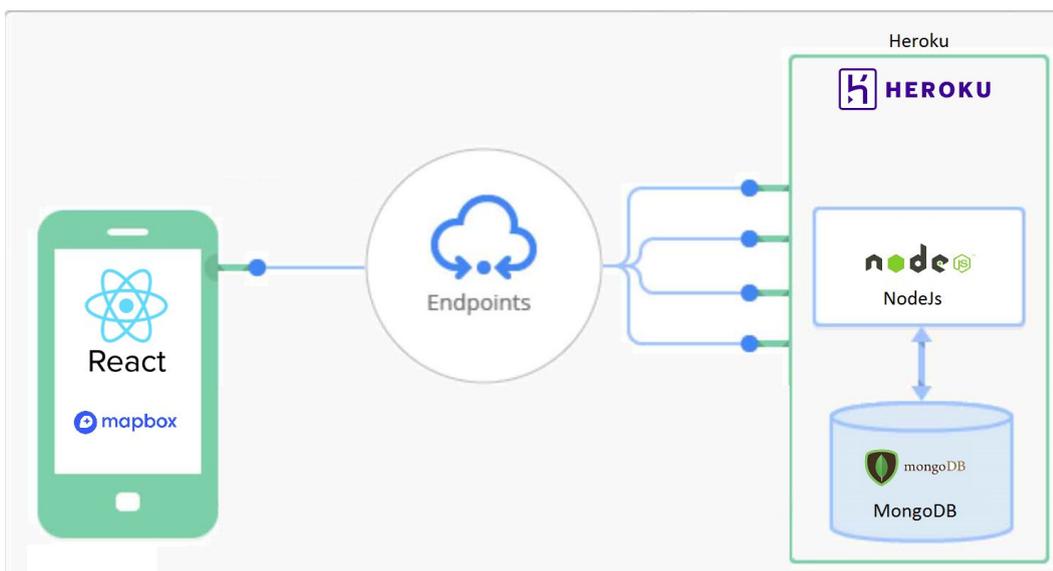


Figura 6 - Arquitetura do sistema [13]

3.2 React Native

Aproveitando o crescimento do React e das necessidades do Facebook em desenvolver aplicações para dispositivos móveis, eles desenvolveram também o React Native, apesar do React native ser derivado do React eles não são completamente reutilizáveis, pois existem diferenças fundamentais entre plataformas móveis nativas e aplicações web acessadas pelos navegadores. De forma que o React não se propõe a fazer com que uma aplicação que seja escrita uma vez execute em ambientes web, *android* e/ou IOS sem tratamentos e alterações específicas. De fato, essa, se aproveita das partes em que há uma sobreposição de conceitos. Então o que o React e React Native se propõe é criar unidades de desenvolvimento que possam ser compartilhadas, essas unidades são chamadas de componentes. Se no desenvolvimento web existe a manipulação da árvore DOM pelos componentes, nas aplicações móveis nativas são utilizadas API's de chamadas assíncronas.

React Native [14] foi desenvolvido pelo Facebook após o grande sucesso do React também desenvolvido por eles. O React é uma biblioteca leve que tem como principais vantagens uma alta performance e a pretensão de uma curva de aprendizado mais suave do que os concorrentes, pois se propõe a resolver adversidades do desenvolvimento web, mas somente focado no desenvolvimento da camada de visão da aplicação. O que separa o React das demais bibliotecas que agem sobre a camada de visão são os componentes, unidades de desenvolvimento reutilizáveis que são declarados no início da renderização, e permanecem assim após as sucessivas renderizações dos componentes. O React mantém uma árvore DOM (*Document Object Model*) virtual, que é utilizada para manter a representação da DOM real, de forma que a comparação seja feita com o estado anterior às mudanças na árvore e assim garantindo uma otimização, visto que a manipulação de árvores DOM é reconhecidamente algo que exige bastante esforço computacional. Essa otimização da aplicação é feito pelo gerenciamento e comparação de estados da árvore DOM real com a DOM virtual, possibilitando que somente os componentes que sofreram alteração seja realmente renderizados novamente.

3.3 Mapbox

Mapbox [15] é uma plataforma de mapeamento de localizações, trajetos e áreas, renderizado em tempo real, que disponibiliza uma pluralidade de ferramentas que envolvam geolocalização e uma suave visualização gráfica de mapas e das informações inseridas, assim como também navegação de trajetos, logística, visualização de dados e realidade aumentada. Utilizado no desenvolvimento de software de sucesso em grandes empresas, e tem entre seus clientes marcas mundiais como Tinder, MongoDB, Microsoft, IBM, BOSCH, CNN, National Geographic, Github, General Electric e Mastercard

O Mapbox se diferencia das demais plataformas de localização online por ser focada nos desenvolvedores de aplicações, sendo mais amigável e apresentando uma curva de aprendizado mais suave, assim como uma maior portfólio de customização, sendo fácil manipular suas fontes e sua aparência. Outra vantagem também é que o Mapbox é menos caro e mais flexível que o Google Maps.

Apesar do Mapbox ter diversos clientes gigantes em suas áreas de atuação, em maio de 2018, apenas 3% de seus usuários eram usuários pagos. Isso acontece porque ele disponibiliza serviços de forma gratuita para usuários com baixo consumo, visando penetrar no mercado de desenvolvimento de aplicações com baixo investimento, mas que porventura venham a fazer sucesso. Dessa forma, seu orçamento seria alimentado na cobrança de serviços que utilizam sua API e que por seus méritos escalem, ganhando mais usuários [16].

A utilização do Mapbox junto às soluções React Native são incentivadas. Assim, o Mapbox Maps SDK for React Native é uma plataforma de solução transversal para criação de componentes React Native. Parte do seu incentivo à utilização vem do seu site, que tem conteúdo que auxilia na inicialização de projetos desenvolvidos com Mapbox e React Native.

3.4 Node.js

Node.js [17] é um ambiente de execução de Javascript no lado servidor que

funciona orientada a eventos. Desenvolvido pela Google e que é distribuído em regime de código aberto. A execução do Javascript no lado servidor em Node.js é possibilitado pela utilização de um motor V8. O V8 originalmente foi desenvolvido para rodar no navegador, mas posteriormente passou a ser utilizado também no lado servidor com o Node.js. O motor otimiza a execução do Javascript implementando um compilador *Just in time* (JIT) para essa linguagem.

O Node.js é executado em um processo de *thread* única que age por chamadas em *callback*, e nunca realiza bloqueio da *thread* principal, fazendo com que tenha uma boa performance em várias arquiteturas em aplicações web. *Callback* é como é chamada uma função que é passada como parâmetro para outra função, de forma que essa possa ser executada assincronamente, mas somente ao término da função inicial que recebeu a função como parâmetro. Esse modelo é conhecido por “*single thread event loop model*”.

Tal abordagem assíncrona, diferencial do Javascript, é interessante pois permite que funções independentes entre si não sejam executadas em fila, mas sim paralelamente, por exemplo. Para uma função que faz 5 chamadas a outras funções, em um ambiente síncrono, o intervalo de tempo entre o início da primeira execução e o final da última é dado pela soma do tempo de cada uma das funções internas, já que uma função seguinte espera o término do processamento da anterior. Por outro lado, em um ambiente assíncrono, o intervalo de tempo é igual à da função mais demorada, pois elas começam a ser executadas quase que simultaneamente de forma que o processo demora tanto quanto a mais demorada das funções.

Uma adversidade imposta pela execução assíncrona é que desenvolvedores são treinados normalmente para pensar seus códigos de forma sequencial, de modo que depurar códigos assíncronos pode ser desafiador.

As aplicações em Node.js são constituídas basicamente de 3 categorias de módulos: os módulos nativos que já vem embutidas no Node.js em si; os módulos de terceiros que são instalados via Npm (*Node Package Manager*), gerenciador importação de pacotes Javascript; e os módulos locais que são os módulos desenvolvidos pelos desenvolvedores da própria aplicação.

3.5 MongoDB

MongoDB [18] é um SGBD NoSQL (*Not only SQL*), ou seja um gerenciador de banco de dados que relaxa os princípios ACID (atomicidade, consistência, isolamento e durabilidade) em favor de performance, escalabilidade, flexibilidade, e redução da complexidade.

Sendo orientado a documentos, este delega poder e responsabilidade para a aplicação implementar as propriedades encontradas nas coleções. Tendo por característica chave ser um SGBD de propósito genérico, ou seja, não tem um fim específico como trabalhar com grafos, ou com pares chave e valor, ou grandes massas de dados como outros bancos não-relacionais.

Seus esquemas flexíveis fazem contraste com os bancos relacionais, pois seus atributos não são bem delimitados e definidos, focando desta maneira na alta disponibilidade. Até mesmo por sua vocação genérica oferece uma variedade de opções, como operadores equivalentes aos operadores SQL (*Structured Query Language*), ao mesmo tempo que também disponibiliza opções como MapReduce, TTL (*Time to Live*) e indexação secundária, que são características de bancos de dados NoSQL para *Big Data*.

Os documentos *JSON* (*Javascript Object Notation*) utilizados pelo SGBD nativamente é um fator positivo quando combinado a aplicações Node.js, já que ambos compartilham diretamente de um formato que é amplamente utilizado nos ambientes web.

Este SGBD também permite o armazenamento de *GeoJSON* que possibilita busca por *Geospatial Queries*, formato com diversas opções de busca geolocalizadas por distância, perímetros virtuais e interseção entre áreas.

O banco de dados deste projeto está desenvolvido apoiado em basicamente 2 conceitos, o *schema* de usuário, que é composto por nome, nome de usuário, email e senha, data de criação e data da última atualização. e o *schema* de localidade que é composto por título, descrição, latitude, longitude, uri das imagens e um id do usuário que criou a localidade com o registro de irregularidade.

3.6 Heroku

Heroku [19] é uma Plataforma de Serviço de aplicações em nuvem que executam em leves *containers* chamados de dynos, com suporte a diversas linguagens como Node.js, Ruby, Java, Python, Goo, dentre outras, assim como também suporta SGBD como Postgres, Redis, Kafka e o próprio MongoDB. A plataforma também permite integração com o Docker, tecnologia de containerização para aplicações, com a *ide* Eclipse ou até mesmo com o Github, que é o onde estão versionada nossas aplicações. Permitindo *deploy* automático do código, bastando apenas que código seja versionado na *master* da própria plataforma em nuvem.

Todas as aplicações do Heroku rodam sobre leves containers Linux que fazem containerização. Portanto, dino é o como é chamado o simples servidor privado e virtual (compartilham o hardware com outras aplicações) que executam as aplicações individualmente suas respectivas aplicações web.

O Heroku permite a escalabilidade tanto horizontalmente pelo número de *containers* empregados pela aplicação hóspede, como também verticalmente com a utilização *containers* mais robustos com capacidade de até 14 GB de *ram*. Os *dynos* free, *containers* gratuitos, tem capacidade de 512 mb e podem ser compartilhados entre usuários. Existem três tipos de *dynos*, os web que suportam requisições http, os para transição de *containers* e outros para os trabalhos de processamento background.

Essa plataforma então permite que nossa aplicação *backend* que tem por objetivo integrar os dados inseridos pelos usuários possam ser persistidos, gerenciados, e disponibilizados de forma instantânea. Dessa forma, a qualquer momento em que a aplicação móvel esteja sendo utilizada, esses dados podem ser acessados para que a aplicação realize suas funcionalidades.

4 Fiscal da Cidade

Este capítulo descreve a aplicação, suas funcionalidades, exibindo as telas desenvolvidas e conta com o link e QrCode para o instalador do aplicativo.

4.1 Login

É a primeira tela apresentada ao usuário ao abrir o aplicativo. (Figura 7)



Figura 7 - Tela de Login.

Nela é possível efetuar o cadastro caso não se tenha uma conta criada o usuário pode clicar em criar conta, preencher os dados de registro (Figura 8), criar a conta (Figura 9) e logar (Figura 10).

Fiscal da Cidade

Bem vindo, fiscal. Neste aplicativo você poderá denunciar irregularidades a sua volta pela cidade.

leonardo.fmsantos@uniriotec.br

.....

Leonardo Santos

[Criar conta](#) [Esqueceu a senha?](#)

Figura 8 - Usuário clica em criar conta e preenche os dados.

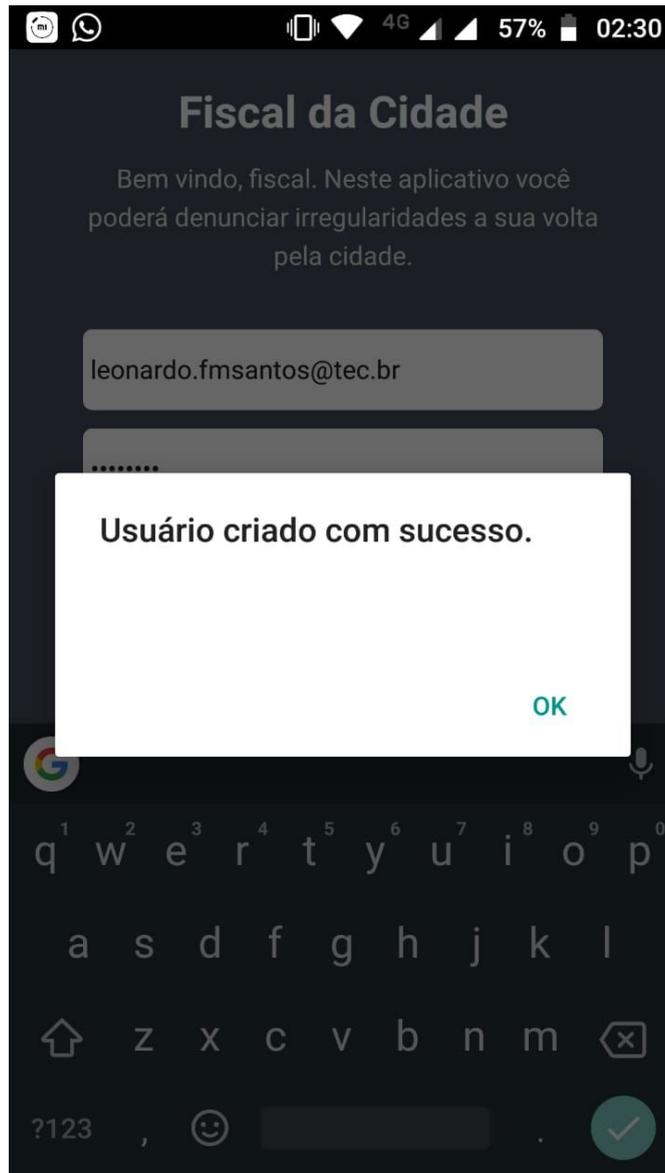


Figura 9 - Usuário cria a conta com sucesso.



Figura 10 - Usuário pode então logar na aplicação.

4.2 Core da aplicação

É a tela em que é apresentado o mapa, onde se cria o registro e são exibidos os pontos de irregularidades reportados. (Figura 11)

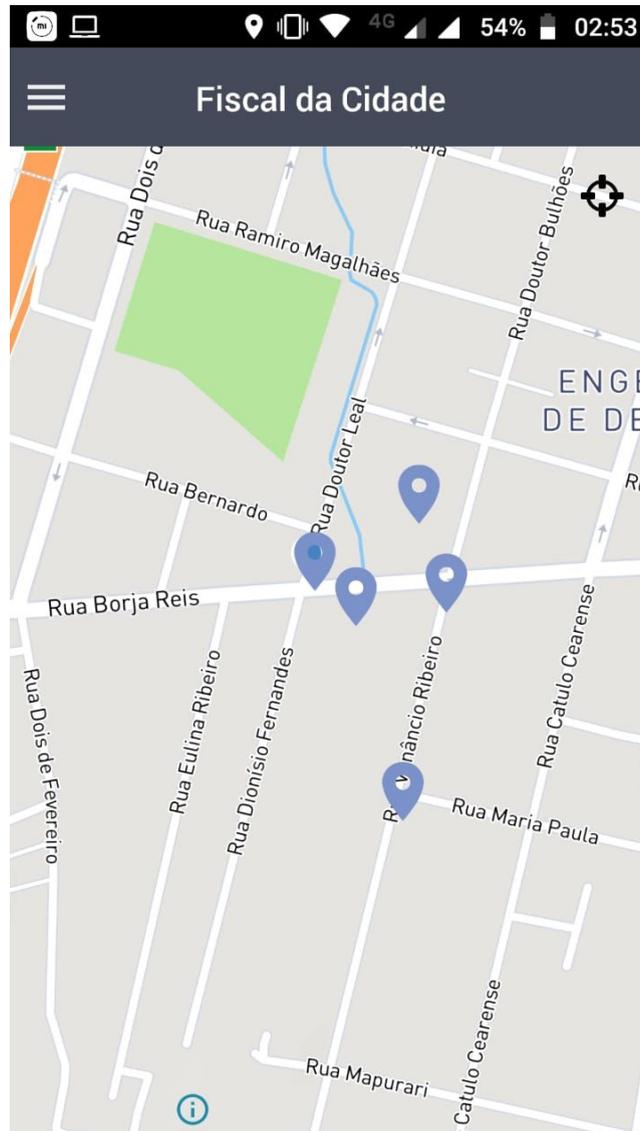


Figura 11 - Tela principal da aplicação.

4.2.1 Criar registro

Ao se clicar no mapa, o usuário será apresentado a um formulário (Figura 12) em que deve preencher corretamente para criar um registro. Nele é possível escolher um título de acordo com os problemas listados (Figura 13), preencher sua descrição (Figura 14) com as informações pertinentes e adicionar fotos (Figuras 15 e 16). Após essas etapas o usuário clica em salvar e o registro é criado (Figura 17).



The image shows a mobile application interface for creating a record. At the top, there is a status bar with icons for messages, photos, and social media, along with location, signal strength, Wi-Fi, 4G, and battery level (57%) indicators. The time is 02:31. Below the status bar, the word "CADASTRO" is displayed in white text on a dark blue background. The form consists of several elements: a white text input field with the placeholder text "Selecione o tipo de problema"; a larger white text input field with the placeholder text "Descrição"; a blue button with the text "Adicionar foto"; and a final blue button with the text "Salvar".

Figura 12 - Usuário abre formulário de descrição do registro ao clicar para marcar o ponto.

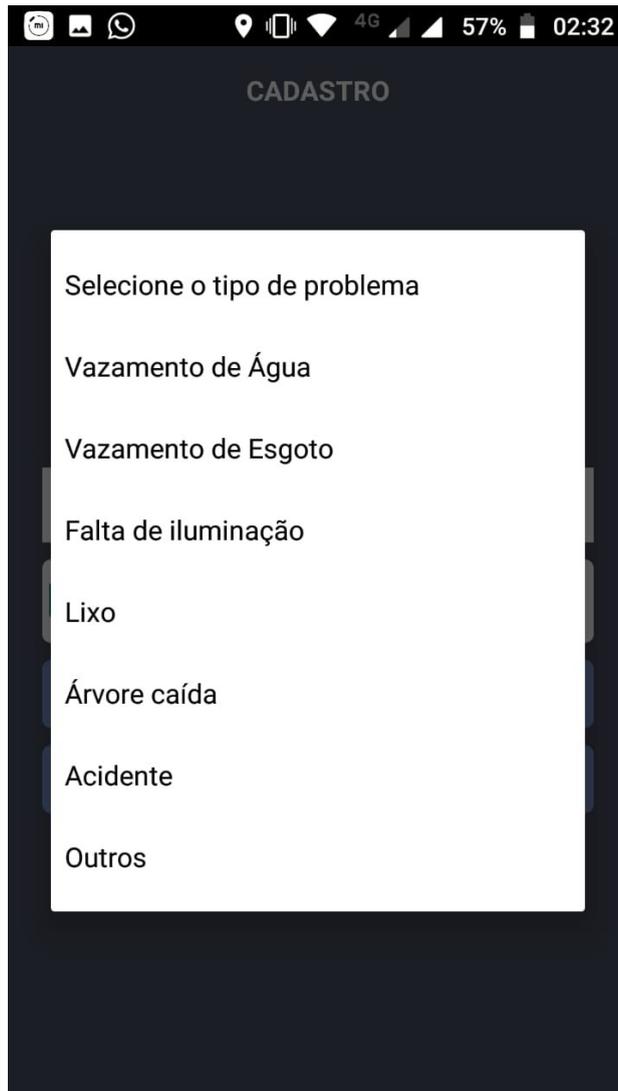


Figura 13 - Usuário seleciona tipo de registro.



Figura 14 - Usuário preenche descrição do registro.



Figura 15 - Usuário pode adicionar foto.



Figura 16 - Usuário escolhe se deseja manter a foto ou excluir.



Figura 17 - Registro criado após usuário clicar em salvar.

4.2.2 Descrição da localidade

Ao clicar nas localizações é possível ver a classificação do registro, descrição e foto desses pontos (Figura 18). Nas mesmas ainda é possível apoiar, no caso do usuário que está acessando aquela informação considerá-la válida (*like*) ou não (*dislike*) (Figura 18).



Figura 18 - Usuário clica em alguma marcação do mapa, visualiza a descrição dela e apoia o conteúdo como verdadeiro com um “like”.

Caso o usuário seja o criador daquele registro, há possibilidade do próprio excluir (Figura 19). É necessário que o usuário criador da localidade confirme a ação para excluí-la (Figura 20).

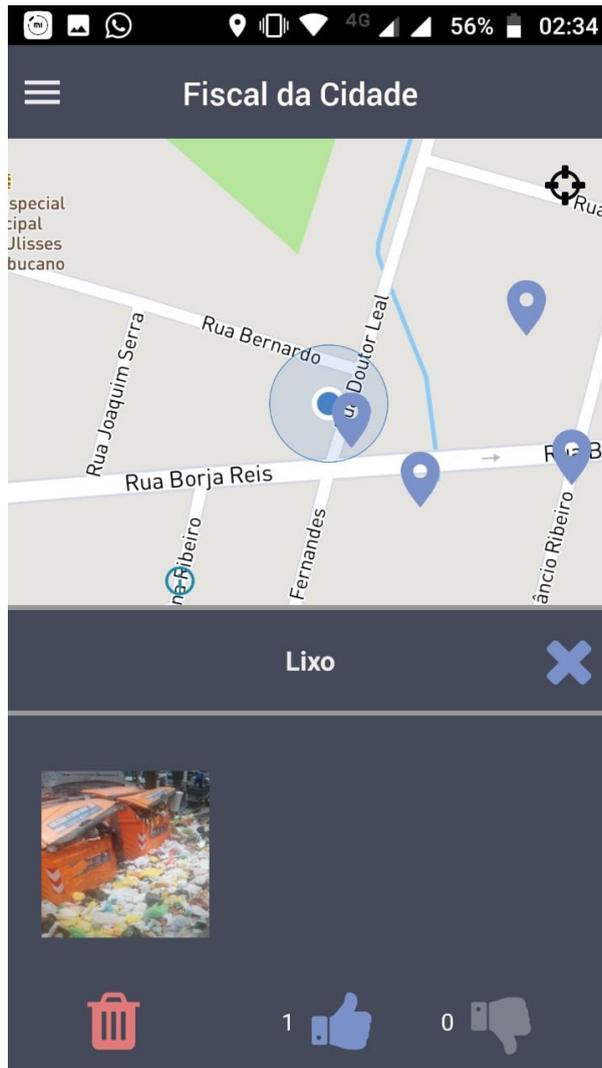


Figura 19 - Opção de excluir localidade é habilitada quando o criador daquele registro é o visualizador.



Figura 20 - Confirmação para excluir registro.

4.3 Informações do usuário

Há um menu lateral (Figura 21) que apresenta as informações do usuário, nome, email, data de cadastro no sistema. Esse menu lateral também apresenta uma lista dos pontos de registros criados por ele com o tipo de problema e a data de criação. Ao clicar o mapa é direcionado para o local do registro e abre sua descrição.

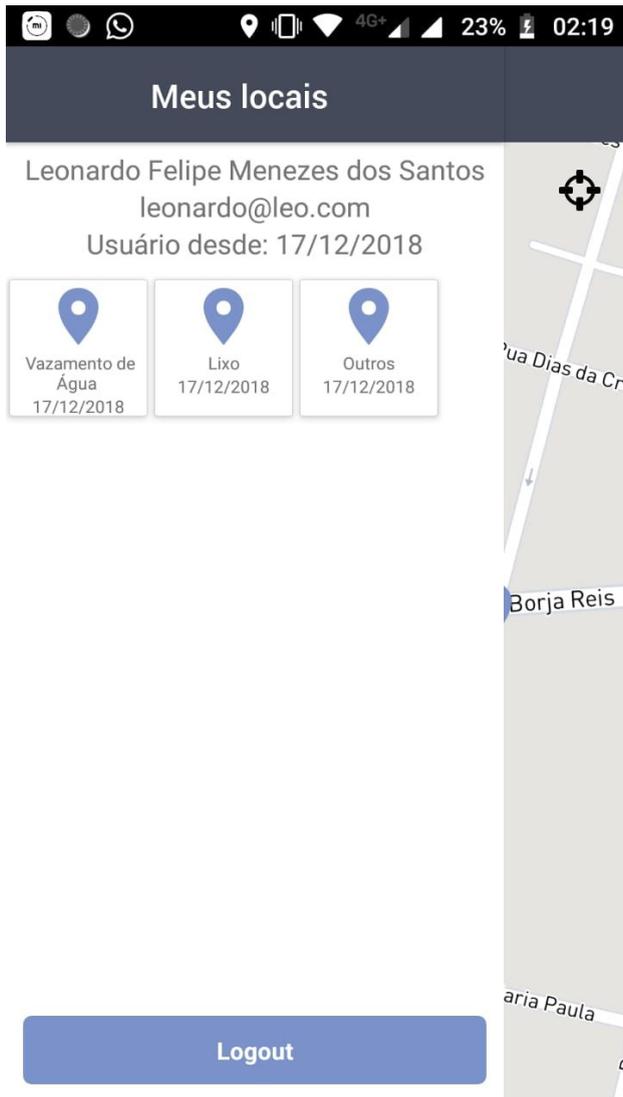


Figura 21 - Menu Lateral.

4.4 Link para o instalador da aplicação

A fim de promover a distribuição do sistema e também os testes pelos leitores deste trabalho, disponibilizou-se um *apk* para instalação do aplicativo em dispositivos móveis. O *link* para download do *apk* pode ser obtido no QR Code exibido na Figura 22.



Figura 22 - QR Code para o download do apk do aplicativo.

<https://goo.gl/PpurVU>

5. Especificações do Sistema

5.1 Diagrama de Classes

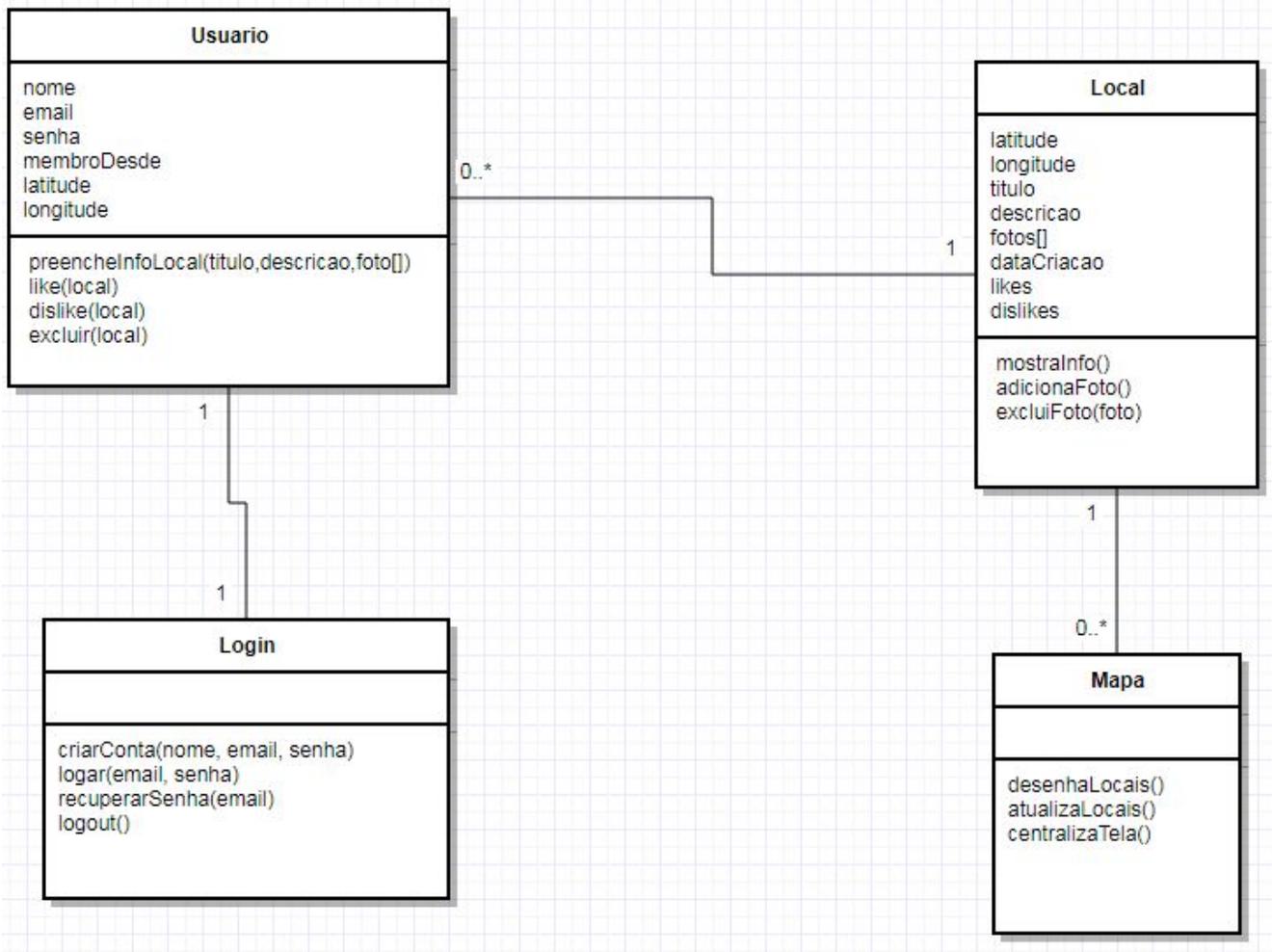


Figura 23 - Diagrama de Classes da aplicação

5.2 Casos de Uso

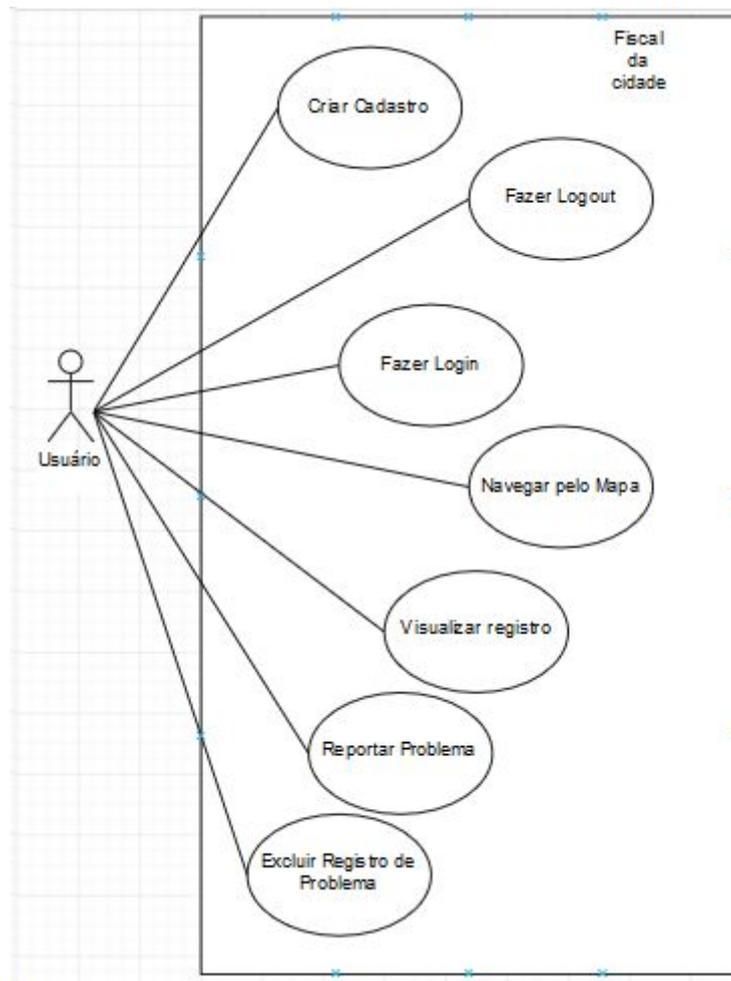


Figura 24 - Visão geral dos casos de uso

5.2.1 Caso de Uso: Criar Cadastro

Atores do Caso de Uso: Usuário.

Grau de Abstração: Essencial.

Tipo do caso de uso: Primário.

Pré-Condições: Aplicativo instalado no celular, conexão com a internet.

Pós-Condições: Cadastro realizado.

Descrição em alto nível: Cadastrar um novo usuário no sistema.

Fluxo Normal:

1. Usuário clica em criar conta
2. Visitante fornece dados de E-mail, senha e nome completo
3. Sistema cadastra novo usuário.
4. Encerra caso de uso

Fluxo Alternativo:

1. Visitante não fornece todos os dados necessários
 - a. Usuário não cadastrado.
 - b. Encerra caso de uso

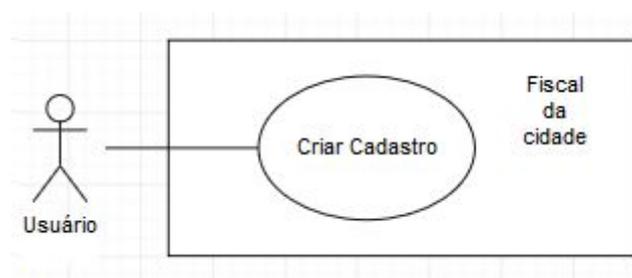


Figura 25 - Caso de uso: Criar Cadastro

5.2.2 Caso de Uso: Fazer Login

Atores do Caso de Uso: Usuário.

Grau de Abstração: Essencial.

Tipo do caso de uso: Primário.

Pré-Condições: Aplicativo instalado no celular, conexão com a internet.

Pós-Condições: Login efetuado

Descrição em alto nível: Usuário insere seus dados para fazer login.

Fluxo Principal:

1. Usuário solicita Logar no sistema
2. Usuário insere e-mail e senha
3. Sistema autoriza entrada do usuário no sistema
4. Usuário Loga no sistema
5. Encerra caso de uso

Fluxo Alternativo:

1. Usuário insere e-mail e/ou senha incorreto(s)
 - a. Sistema informa que não é possível fazer login com os dados fornecidos
 - b. Encerra caso de uso

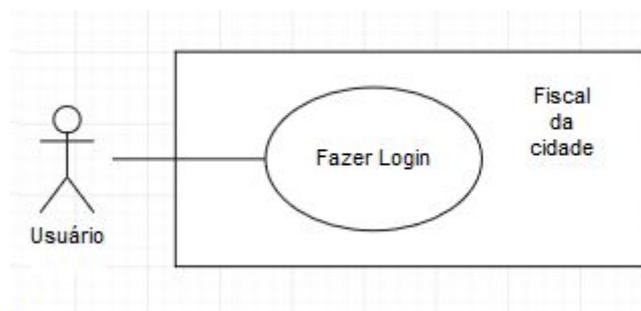


Figura 26 - Caso de uso: Fazer Login

5.2.3 Caso de Uso: Fazer Logout

Atores do Caso de Uso: Usuário.

Grau de Abstração: Essencial.

Tipo do caso de uso: Primário.

Pré-Condições: Aplicativo instalado no celular, conexão com a internet, Usuário logado.

Pós-Condições: Usuário não logado.

Descrição em alto nível:

Fluxo Principal:

1. Usuário clica no menu superior lateral esquerdo
2. Clicar no botão de logout
3. Encerra caso de uso

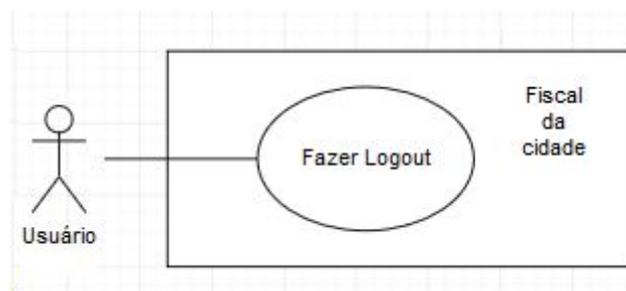


Figura 27 - Caso de uso: Fazer Logout

5.2.4 Caso de Uso: Navegar pelo Mapa

Atores do Caso de Uso: Usuário.

Grau de Abstração: Essencial.

Tipo do caso de uso: Primário.

Pré-Condições: Aplicativo instalado no celular, conexão com a internet, Usuário logado.

Pós-Condições:

Descrição em alto nível: Usuário navega pelo mapa, visualizando problemas reportados e suas localizações.

Fluxo Principal:

1. Usuário desliza com um toque para navegar pelo mapa
2. Usuário visualiza os registros de problemas existentes localizados no mapa
3. Encerra caso de uso

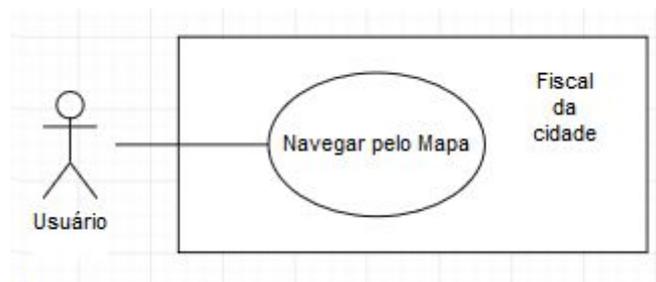


Figura 28 - Caso de uso: Navegar pelo Mapa

5.2.5 Caso de Uso: Visualizar registro

Atores do Caso de Uso: Usuário.

Grau de Abstração: Essencial.

Tipo do caso de uso: Primário.

Pré-Condições: Aplicativo instalado no celular, conexão com a internet, Usuário logado, Usuário com o mapa em tela

Pós-Condições: Registro aberto para visualização

Descrição em alto nível: Usuário ao navegar pelo mapa clica no marcador do registro

Fluxo Principal:

1. Usuário navegando pelo mapa
2. Usuário clica em marcador no mapa
3. Sistema exibe informações de título, descrição e imagem do problema. Assim como a quantidade de interações positivas e negativas
4. Sistema apresenta as possibilidades de interação positiva, negativa e fechar exibição do registro.
5. Encerra caso de uso

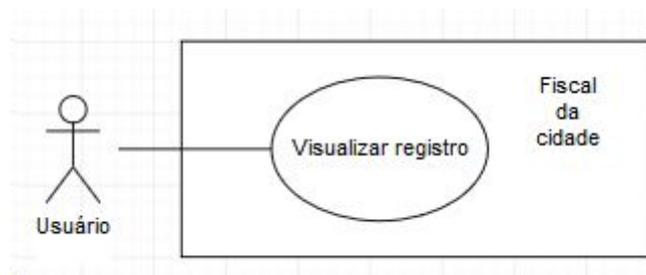


Figura 29 - Caso de uso: Visualizar registro

5.2.6 Caso de Uso: Reportar Problema

Atores do Caso de Uso: Usuário.

Grau de Abstração: Essencial.

Tipo do caso de uso: Primário.

Pré-Condições: Aplicativo instalado no celular, conexão com a internet, Usuário logado, Usuário com o mapa em tela

Pós-Condições: Problema reportado e disponível no mapa para visualização

Descrição em alto nível: Usuário visualiza suas contribuições e a situação de cada uma delas.

Fluxo Principal:

1. Usuário clica numa área livre no mapa
2. Sistema exibe formulário para registro de problema a reportar
4. Usuário seleciona tipo de problema
5. Usuário descreve o problema
6. Usuário submete imagem
7. Usuário clica em salvar para efetuar o registro do problema
8. Encerra caso de uso.

Fluxo Alternativo:

6. Usuário opta por não submeter imagem
7. Usuário clica em salvar para efetuar o registro do problema
8. Encerra caso de uso.

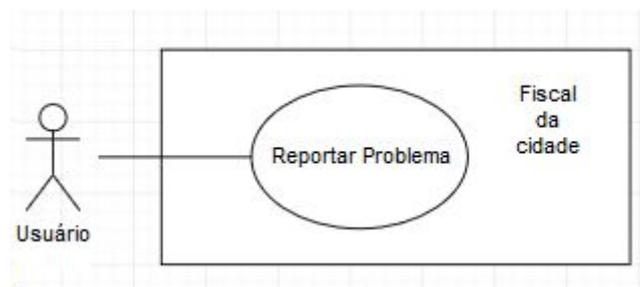


Figura 30 - Caso de uso: Reportar Problema

5.2.7 Caso de Uso: Excluir Registro de Problema

Atores do Caso de Uso: Usuário.

Grau de Abstração: Essencial.

Tipo do caso de uso: Primário.

Pré-Condições: Aplicativo instalado no celular, conexão com a internet, Usuário logado, Usuário com o mapa em tela

Pós-Condições: Registro excluído

Descrição em alto nível: Permite que o Usuário criador exclua seu registro.

Fluxo Principal:

1. Usuário navegando pelo mapa
2. Usuário clica em marcador no mapa
3. Sistema exibe informações de título, descrição e imagem do problema. Assim como a quantidade de interações positivas e negativas
4. Sistema apresenta as possibilidades de interação positiva, negativa e fechar exibição do registro.
5. Sistema apresenta a possibilidade de exclusão de registro para seu criador
6. Usuário clica no botão de exclusão de registro
7. Sistema pergunta ao Usuário se deseja realmente excluir
8. Usuário confirma exclusão
9. Encerra caso de uso.



Figura 31 - Caso de uso: Excluir Registro de Problema

6 Conclusão

6.1 Considerações finais

A percepção do crescimento da utilização de tecnologias móveis no âmbito das cidades e com o crescimento dos sistemas colaborativos, em que as pessoas podem compartilhar de diversos tipos de informação ou serviços através de diferentes sistemas, inspirou a ideia desse trabalho que teve por objetivo propor uma solução cooperativa e compartilhada para que o cidadão pudesse expor problemas do dia a dia do ambiente em que transita, como também, para os demais cidadãos que de alguma forma interseccionam com sua área geográfica.

Toda essa integração se torna ainda mais interessante, pois além de informar sobre eventos de interesse da administração pública, ainda existe o georreferenciamento implementado de forma prática, intuitiva e rápida se utilizando tanto do GPS nativos em aparelhos quanto da implementação do Mapbox. O que é um diferencial visto que localização espacial é um fator diferencial tanto para que o problema seja tratado rapidamente, como também para que possam ser feitas análises por sub regiões.

6.2 Trabalhos futuros

O intuito desse sistema é possibilitar que a entrada de dados e o compartilhamento dos dados inseridos fossem visualizados através da própria aplicação móvel. Portanto, ainda existe um conjunto de possibilidades a serem trabalhadas e desenvolvidas, dentre elas:

- O acúmulo de maior quantidade de informações disponibilizadas pelo usuário, como dados socioeconômicos como idade, renda, educação, gênero, etnia e outros, com o propósito de análise mais rica tanto sobre o alcance da aplicação, quanto da relação dos dados com esses fatores;
- Tratamento de dados sobre limites geográficos para que as informações possam ser tratadas de forma específica para divisões geopolíticas como municípios, estados e países;
- O tratamento mais profundo sobre questões relacionadas à privacidade e segurança da informação dentro da aplicação, que garantiria ao usuário a liberdade e poder de fazer publicações anônimas. Permitindo que questões mais graves como a de segurança pública ou outros casos específicos fossem tratados de forma devida;
- Tratamento mais profundo sobre as localidades marcadas para essas possam ter mais funcionalidades como notificação de fechamento de ocorrência por sucesso, improcedente, tempo expirado e possibilidades de comentários em cada uma das localidades;
- Padronização da exposição de dados para que as entidades governamentais pudessem consumir as informações sobre as reclamações (por exemplo, através de uma API), assim como a possibilidade de mensagens específicas para cada caso; e Estudos sobre a aceitação da ferramenta pelo usuário.

Referências Bibliográficas

1. LIMA, Mariana. *Brasil já tem mais de um smartphone ativo por habitante, diz estudo da FGV*. Disponível em:
<<https://link.estadao.com.br/noticias/geral,brasil-ja-tem-mais-de-um-smartphone-ativo-por-habitante-diz-estudo-da-fgv,70002275238>>. Acesso em: 25 nov. 2018.
2. BOM DIA RIO. *Canal entre prefeitura e cidadão, telefone 1746 é alvo de reclamações*. Disponível em:
<<https://g1.globo.com/rio-de-janeiro/noticia/canal-entre-prefeitura-e-cidadao-telefone-1746-e-alvo-de-reclamacoes.ghtml>>. Acesso em: 25 nov. 2018.
3. SARAIVA, Adriana. *Estudo inédito mostra moradores sujeitos a enchentes e deslizamentos*. Disponível em:
<<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/21566-estudo-inedito-mostra-moradores-sujeitos-a-enchentes-e-deslizamentos>>. Acesso em: 25 nov. 2018.
4. BUARQUE, Daniel. *Brasil é 2º país com menos noção da própria realidade, aponta pesquisa*. Disponível em:
<<https://www1.folha.uol.com.br/mundo/2017/12/1941021-brasil-e-2-pais-com-menos-nocao-da-propria-realidade-aponta-pesquisa.shtml>>. Acesso em: 25 nov. 2018.
5. OLIVEIRA, Carla de. *Sistemas Colaborativos: Conceito, Características e Funcionalidades*. Disponível em:
<https://docplayer.com.br/739510-Sistemas-colaborativos-conceito-caracteristicas-des-e-funcionalidades.html/>. Acessado em: 02 dez 2018
6. USABILITY FIRST. *Typical Collaborative Software Applications*. Disponível em:
<http://www.usabilityfirst.com/about-usability/web-application-design/collaborative-software-groupware/typical-collaborative-software-applications/>. Acesso em: 02 dez 2018.

7. COSTA, Matheus. *SUL: Sistema Útil de Localização*. 2017. Trabalho de Conclusão de Curso, Bacharelado em Sistemas de Informação, Universidade Federal do Estado do Rio de Janeiro(UNIRIO), Rio de Janeiro, 2017.
8. BERTOCHÉ, Uriel. *Sistema de Apoio à Mobilidade Urbana com Dados Abertos*. 2016. Trabalho de Conclusão de Curso, Bacharelado em Sistemas de Informação, Universidade Federal do Estado do Rio de Janeiro(UNIRIO), Rio de Janeiro, 2016.
9. TRAFI. Disponível em:
<<https://play.google.com/store/apps/details?id=com.trafi.android.tr/>>. Acesso em: 04 dez 2018.
10. Onde Tem Tiroteio. Disponível em:
<<https://play.google.com/store/apps/details?id=com.ondetemtiroteio.app/>>. Acesso em: 04 dez 2018.
11. Colab. Disponível em: <<https://www.colab.re/>> Acesso em: 04 dez 2018.
12. ONUBR. *ONU quer saber opinião dos brasileiros sobre a vida nas cidades*. Disponível em:
<<https://nacoesunidas.org/onu-quer-saber-opinia-o-dos-brasileiros-sobre-a-vida-nas-cidades/>>. Acesso em: 15 nov. 2018.
13. Backend na nuvem com Cloud Endpoints Disponível. em:
<<https://www.infoq.com/br/presentations/back-end-na-nuvem-com-cloud-endpoints>> Acesso em: 16 dez 2018.
14. React Native. Disponível em: <<https://github.com/facebook/react-native/>>
Acesso em: 15 nov. 2018.
15. Mapbox. Disponível em: <<https://www.mapbox.com/>> Acesso em: 15 nov 2018.
16. BIZ, Carson. *The Right Direction: How Mapbox Is Winning Over Developers To Challenge Google's Mapping Dominance*. Disponível em:
<<https://www.forbes.com/sites/bizcarson/2018/05/08/mapbox-maps-developers/#5eb83e67164d/>>. Acesso em: 15 nov. 2018.
17. NodeJS. Disponível em: <<https://nodejs.org/en/about/>> Acesso em: 15 nov 2018.
18. MongoDB. Disponível em: <<https://www.mongodb.com/what-is-mongodb/>>
Acesso em: 15 nov 2018.

19. Heroku. Disponível em: <<https://www.heroku.com/what/>> Acesso em: 15 nov 2018.