FATEC

Santo André

Eletrônica Automotiva

Murilo Comitre Ceolin

PONTO DE VISTA: Um aplicativo móvel de assistência à mobilidade urbana para deficientes visuais

Santo André - SP

2019

Murilo Comitre Ceolin

PONTO DE VISTA: Um aplicativo móvel de assistência à mobilidade urbana para deficientes visuais

Trabalho de Conclusão de Curso apresentado ao Curso Superior em Eletrônica Automotiva da FATEC Santo André, orientado pelo Prof. Me. Murilo Zanini de Carvalho, como requisito parcial para obtenção do título de tecnólogo em Eletrônica Automotiva.

Santo André - SP

2019

C398p
Ceolin, Murilo Comitre
Ponto de vista: um aplicativo móvel de assistência à mobilidade urbana para deficientes visuais / Murilo Comitre Ceolin. - Santo André, 2019. – 83f: il.

Trabalho de Conclusão de Curso – FATEC Santo André.
Curso de Tecnologia em Eletrônica Automotiva, 2019.

Orientador: Prof. Murilo Zanini de Carvalho

1. Eletrônica. 2. Automóveis. 3. Mobilidade urbana. 4. Deficiente visual. 5. Aplicativo móvel. 6. Software. 7. Trânsito. 8. Programação. 9. Linguagem de programação. 10. Kotlin. 11. GPS. I. Ponto de vista: um aplicativo móvel de assistência à mobilidade urbana para deficientes visuais.

629.2

LISTA DE PRESENÇA

Santo André, 27 de Junho de 2019

LISTA DE PRESENÇA REFERENTE À APRESENTAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO COM O TEMA: **"PONTO DE VISTA: UM APLICATIVO MÓVEL DE ASSISTÊNCIA À MOBILIDADE URBANA PARA DEFICIENTES VISUAIS"** DO ALUNO DO 6º SEMESTRE DESTA U.E.

BANCA

PRESIDENTE:
PROF. MURILO ZANINI DE CARVALHO_____

MEMBROS:
PROF. WESLLEY MEDEIROS TORRES _____

PROF. FERNANDO GARUP DALBO_____

PROF. ROGÉRIO DANIEL DANTAS_____

ALUNO :

MURILO COMITRE CEOLIN_____

Gostaria de dedicar este trabalho a todas as pessoas que possuam algum tipo de deficiência, que hoje ainda sofrem muito, principalmente pela falta de estrutura e suporte para melhor agregá-las na sociedade. Acredito que todos merecem o mínimo de cidadania, respeito e igualdade, portanto, creio que quem possua condições de fazer algo para ajudar a vida de uma pessoa, mesmo que pouco, deva faze-la feliz pois estará ajudando a tornar esse mundo um lugar melhor para morar e ainda, colocando um sorriso no rosto de quem muitas vezes não tem condições de sorrir.

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus por me iluminar com ideias como essa que possam de alguma forma, mesmo que pouco, ajudar pessoas, e que continue me iluminando com novas ideias com a mesma finalidade pois acredito que todos deveríamos de alguma maneira fazer o que pudermos para tornar este mundo um lugar melhor e tudo começa com ajudar e amar o próximo e a Deus. Gostaria de agradecer também meus pais Marcos Antônio Ceolin e Rosana Candida Comitre Ceolin e a minha irmã Monique Comitre Ceolin por tudo que fizeram e ainda fazer por mim sem medir esforços, por terem me dado toda educação e respeito para me tornar a pessoa que sou hoje. Agradeço também a todos os professores da FATEC Santo André por todo suporte e ensinamentos que me deram durante esses três anos, e agradeço especialmente aos professores Murilo Zanini de Carvalho e Fernando Garup Dalbo por terem sido não só professores, mas amigos e que acreditaram na minha ideia desde o início e me incentivaram a continuar firme nesta caminhada, e principalmente o professor Murilo que sem ele hoje não teria conseguido chegar no nível que cheguei, pois além de se tornar um amigo, esteve desde o início comigo e me ajudou a moldar essa e muitas outras ideias apresentando sempre soluções e caminhos para alcançar o meu objetivo. Agradeço também a Isabela Lima de Oliveira, minha namorada, e ao Marcos Soares de Azevedo Cressoni, meu cunhado, por terem me ajudado na formatação desta monografia e no apoio moral durante a sua elaboração. Agradeço também aos meus amigos.

EPÍGRAFE

“A ciência é, portanto, uma perversão de si mesma, a menos que tenha como fim último, melhorar a humanidade. ”

NIKOLA TESLA

“Somente uma vida vivida para os outros é uma vida que vale a pena. ”

ALBERT EINSTEIN

“Seja a mudança que você quer ver no mundo. ”

MAHATMA GANDHI

“*É melhor merecer as honras sem recebê-las do que recebê-las sem merecê-las”*

MARK TWAIN

“Agora vejo que as circunstâncias do nascimento de alguém são irrelevantes, é o que você faz com o dom da vida que determina quem você é.”

MEWTWO

RESUMO

Este trabalho projeta apresentar elementos técnicos relacionados a assistência à mobilidade urbana para deficientes visuais, o projeto consiste no desenvolvimento de uma aplicação móvel voltada para o auxílio no trânsito urbano principalmente de pessoas com deficiência visual. O escopo do projeto visa a utilização somente de um aparelho celular “smartphone” como hardware, ao ponto que o GPS do próprio celular será empregado para direcionamento ao local desejado. O objetivo principal deste trabalho de graduação de curso tem o foco em ser um mecanismo auxiliar para beneficiar pessoas que possuem deficiência visual, com isso, tendo dificuldades amplas na locomoção de centros urbanos devido à má sinalização e a tantas movimentações nestes locais, trazendo um enfoque maior para este setor e resultando em maior autonomia para estas pessoas, também, desenvolver esta tecnologia para que possa ser utilizada para outros ramos e beneficiando novas áreas. A aplicação móvel para celulares será desenvolvida inteiramente em uma plataforma de desenvolvimento chamado *Android Studio* e utilizando uma plataforma a linguagem de programação Kotlin utilizada para o desenvolvimento de aplicações móveis para Android,

Palavras-chave: Deficiente visual. Tecnologia. Aplicativos. Mobilidade urbana. Kotlin. Android Studio. GPS. Locomoção

ABSTRACT

This project intends to present technical elements related to assistance to urban mobility for the visually impaired, the project consists of the development of a mobile application aimed at aiding in urban traffic mainly of people with visual impairment. The scope of the project aims to use only a mobile phone "smartphone" as hardware, to the point that the GPS of the mobile phone itself will be employed to target the desired location. The main objective of this course graduation work is to be an auxiliary mechanism to benefit people who are visually impaired, thus having wide difficulties in locomotion of urban centers due to bad signage and the many movements in these places, bringing a focus greater for this sector and resulting in greater autonomy for these people, also, to develop this technology so that it can be used for other branches and benefiting new areas. The mobile application for mobile phones will be developed entirely on a development platform called Android Studio and using a platform the Kotlin programming language used for the development of mobile applications for Android

Key words: Visually impaired. Technology. Applications. Urban mobility. Kotlin. Android Studio. GPS. Locomotion

Lista de Abreviaturas e Siglas

FAB	*Float Action Button*
API	*Application Programming Interface*
App	Aplicativo
Id	Identificador
LatLong	Latitude e Longitude
URL	Uniform Resource Locator
GPS	*Global Position System*
W-LAN	*Wireless Local Area Network*

Lista de Ilustrações

SUMÁRIO

1. INTRODUÇÃO

Situações cotidianas são deixadas de lado, dada sua trivialidade em vidas agitadas como a dos dias de hoje, contudo, isso nem sempre é realidade para todos. Quando alguém possui necessidade especiais ou algum tipo específico de deficiência, podem sofrer para realizar tarefas que as pessoas sem essa deficiência podem ignorar.

Estes aspectos demonstram a carência estrutural de assistência para pessoas deficientes, tornando estes, dependentes de alguém para que possam circular pelas cidades, tirando sua autonomia e dificultando algumas tarefas. Outro ponto dificultoso está no preço em que aparelhos de suporte possuem, até uma simples bengala pode ter um preço alto que muitos podem não ter a capacidade de pagar.

Com a situação exposta, este projeto está foi desenvolvido para apresentar uma solução para a mobilidade de pessoas com algum grau de deficiência visual.

1.2 Objetivo

O objetivo principal deste trabalho é desenvolver um aplicativo móvel para celulares, que terá a função de guiar o usuário do aplicativo ao local desejado notificado, informando ao motorista de ônibus próximo ou a estação de trem/metrô se há um usuário com deficiência visual e informando também, o usuário que seu ônibus está chegando caso ele se encontre já próximo ao ponto de ônibus ou à estação.

1.3 Justificativa

O tema do projeto foi escolhido com o propósito principal de ajudar as pessoas, proporcionando aos usuários maior facilidade e autonomia em sua mobilidade urbana, empregando a tecnologia a seu favor. Ficou estabelecido também, beneficiar-se de ferramentas tecnológicas comuns no dia a dia, como celulares e aplicativos.

O aplicativo proporcionará a oportunidade de deficientes visuais não precisarem mais de alguém para orientá-los pela cidade, devolvendo a eles sua autonomia de se locomover e reduzindo o custo, pois não é necessário gastar com aparelhos além de um smartphone.

1.4 Organização do trabalho

O trabalho está dividido em sete capítulos. Na Fundamentação Teórica há uma análise de toda a fundamentação utilizada no trabalho, contendo referências de artigos de assuntos correlatos ao tema de mobilidade urbana e aplicativos.

No capítulo Metodologia está é apresentada, contendo explicações dos métodos usados para o desenvolvimento do projeto. No capítulo do Desenvolvimento, todas as etapas para a elaboração do trabalho, incluindo os passos seguidos para a sua realização, foram apresentados.

Os Resultados e Discussões sobre o projeto encontram-se no quarto capítulo trazendo os resultados dos ensaios realizados com o aplicativo, suas funcionalidades e navegação. Por fim, no quinto capítulo há as Considerações Finais do trabalho de conclusão de curso e as propostas de trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo foi apresentada a fundamentação teórica utilizada. Mostrando conceitos analisados e problemas reais que diretamente ou indiretamente motivaram a escolha deste tema para a realização deste projeto, levando em conta alguns aspectos citados foi então definido o escopo do trabalho.

1.1 Mobilidade urbana

Segundo Gutierrez et al. (2016) na atualidade, a mobilidade urbana representa um dos problemas a serem enfrentadas nos grandes centros urbanizados, essencial para a produtividade econômica das cidades e para a garantia da qualidade de vida dos cidadãos que nela vivem.

Barbosa (2015) realizou uma pesquisa qualitativa e exploratória de postagens de 15 *blogs* de pessoas com deficiência e como categoria central deste estudo estava a mobilidade urbana como condição estratégica para a inclusão de pessoas com deficiência. No estudo as mais representativas das dificuldades que as pessoas com deficiência encontram para se locomover nas cidades estão na dificuldade de mobilidade:

> Ruas apertadas, calçadas cheias de obstáculos e que não comportam um cadeirante, transporte como barreira à locomoção; ausência de sinal sonoro dificulta mobilidade de pessoas com deficiência visual; problemas nas calçadas: calçada malconservada, com buracos, com entulho, inacabada, ou inexistência de calçada; falha na construção de rampas rebaixadas nas calçadas; rampas que são obstruídas por postes, buracos, que estão pela metade, ou ainda que não têm ligação com o outro lado da rua. (BARBOSA, 2015).

Segundo Bohusch (2013) a mobilidade pode ser interpretada como um processo de três níveis em um sistema sendo eles o global (políticas e ações internacionais), o regional (políticas e ações nacionais e estaduais) e o local (políticas e ações municipais e locais).

Catunda e Santana (1999) relatam que a produção do espaço urbano, juntamente com as novas relações trabalhistas, configurou as cidades a partir da chegada de novos equipamentos urbanos e do surgimento de novos serviços.

Associada à intensa migração e à falta de uma política de desenvolvimento urbano as cidades cresceram desenfreadamente, causando uma progressiva expansão da malha urbana.

Costa (2008) destaca que há uma preocupação com os padrões de desenvolvimento das cidades devido ao rápido crescimento e espalhamento urbano causados também por incapacidade do poder público de controlar o crescimento de território.

De acordo com uma pesquisa feita pelo Metrô de São Paulo (2013), mais de 50% dos deslocamentos das pessoas na região da Grande São Paulo são realizados pelo transporte público. O meio de transporte mais comum é o ônibus entre os meios de transporte disponíveis em São Paulo. A frota transporta mais gente do que todas as linhas do Metrô juntas. Somam cerca de 17 milhões de pessoas por dia, quase nunca em situação ideal.

> O Plano de Mobilidade Urbana é instrumento de incorporação das diretrizes, dos objetivos e dos princípios gerais da Política Nacional de Mobilidade Urbana. Tem como finalidade, por meio do planejamento de curto, médio e longo prazos, apresentar as melhorias da mobilidade urbana local em forma de metas, ações estratégicas e recursos materiais e humanos, possibilitando a efetiva transformação desejada e, assim, contribuindo com uma real promoção do desenvolvimento da cidade. (BRASIL, 2007)

(BRASIL, 2012) é citado por Gomes e Garcia (2017) afirmando que a acessibilidade é uma das formas de garantia de Direitos Humanos, logo ao notar-se a sua falta, isso ocasiona na exclusão de pessoas com deficiência das “trocas sociais, das práticas componentes e dos direitos de integração social e de identidade”, dando lugar à invisibilidade das pessoas com deficiência e a uma violência normativa que privilegiando algumas pessoas e outras não. Dessa forma dificultando vidas através da violação dos Direitos Humanos e gerando sofrimentos que diminuem o potencial de ação dos sujeitos que estão submetidos a ela, Carvalho (2017) diz que todos os dias, uma parte significativa da população encontra muitas barreiras nos espaços e transportes públicos.

De acordo com Balassiano (2012) é evidente a ligação existente entre a operação dos sistemas de transportes, a mobilidade urbana e os conceitos de Gerenciamento da Mobilidade e de sustentabilidade. Esse conjunto de conceitos e sistemas, é complicado e necessita de uma abordagem, de acordo com o autor “uma vez que promover mobilidade sustentável é o foco do planejador urbano ou de transportes”.

Para Campos (2006) levando em consideração as dimensões do desenvolvimento sustentável, é possível concluir que a mobilidade dentro da visão da sustentabilidade pode ser alcançada em relação a dois pontos, são eles uma relação entre a adequação da oferta de transporte e o contexto socio-econômico e outro ponto seria em relação com a qualidade ambiental. Para a autora Vania Campos “No primeiro se enquadram medidas que associam o transporte ao desenvolvimento urbano e a equidade social em relação aos deslocamentos e no segundo se enquadram a tecnologia e o modo de transporte a ser utilizado.”

> A maior cidade brasileira tem 25% da frota nacional, o que hoje representa perto de cinco milhões de veículos. Praticamente temos um carro para cada dois habitantes. A pesquisa Origem-Destino, realizada a cada dez anos desde 1967, abrangendo a área mais fortemente urbanizada da Região Metropolitana de São Paulo – que registra perto de seis milhões de veículos –, identificou em sua última versão 30 milhões de deslocamentos diários, sendo 10 milhões em transporte coletivo, 10 milhões em transporte individual e os restantes 10 milhões a pé. Nos últimos 40 anos tem surgido sempre o questionamento sobre o possível futuro colapso ou travamento total do trânsito. A hipótese de caos generalizado baseia-se em premissa falsa. Seria necessário que a cidade se verticalizasse indefinidamente ou que a taxa de motorização chegasse a níveis estratosféricos, o que nunca irá ocorrer. Entretanto, o que se verifica é o aumento do grau e da extensão da área de deterioração do trânsito na cidade, que acaba contribuindo para a degradação urbana. (SCARINGELLA, 2001).

Alexandre de Ávila Gomide (2006) relata que estudos e pesquisas realizados pelo autor concluíram que: “as populações de baixa renda das grandes metrópoles brasileiras1 estão sendo privadas do acesso aos serviços de transporte coletivo, um serviço público de caráter essencial, conforme a Constituição Federal de 1988. ” Segundo o autor, esta privação ajuda a reduzir oportunidades, pois impossibilita essas populações de ter acesso a equipamentos e serviços como escolas, hospitais, lazer, emprego etc.

> Historicamente, questões de planejamento urbano encontram-se associadas de forma intrínseca a aspectos de transporte, isto é, o crescimento das cidades influencia e é influenciado pelos meios de transporte disponíveis à sua população. Mais ainda, a forma como se dá o processo de circulação urbana interfere diretamente na demanda por transportes, nas áreas destinadas a estacionamento, nos congestionamentos, etc. O crescimento urbano desordenado, provocado pelo espalhamento espacial, o aumento

> excessivo no uso do automóvel, a falta de infraestrutura urbana, a poluição do meio ambiente, entre outras, são questões que interferem na qualidade de vida da população. (MAGAGNIN E SILVA, 2008).

Segundo Priscilla Alves (2014) a mobilidade urbana precisa ser repensada em aspectos no campo político e na prática, as ações precisam ser relacionadas de forma conjunta com a expansão urbana e precisam respeitar as necessidades reais de deslocamentos da população. Para a autora Priscilla Alves sobre a mobilidade: “A mobilidade precisa, ainda, aplicar o viés da sustentabilidade em suas ações, sustentabilidade essa, que consiste, de forma prioritária, em incentivo ao uso de modos de transporte mais sustentáveis, [...] conforto e segurança nos deslocamentos urbanos.”

1.2 Uso da tecnologia para promover a mobilidade urbana

Para Corrêa (1995), ter mobilidade é poder sair de casa e chegar ao seu destino no horário programado, seja no trabalho, escola ou qualquer outro destino, de ônibus, de metrô, de bicicleta, de cadeira de rodas, de carrinho de bebê, a pé, sem imprevistos nos horários previstos e que as ruas e avenidas disponibilizem fluidez mais satisfatória, dando assim condições de segurança para que se possa optar por um meio de transporte adequado e até mais saudável como bicicleta.

Queiroz (1999) relata que é inerente aos seres humanos, e se refere à atitude de se colocar no lugar do semelhante, visualizando a situação na perspectiva do outro. Em outras palavras, as máquinas reproduzem aquilo para o qual foram programadas, porém é preciso dotá-las de uma abordagem mais orgânica e voltada para as diferentes maneiras como as pessoas se comportam, evitando fadá-las a um comportamento igualmente programado. Assim, da mesma forma que as pessoas possuem diferentes aspectos (físicos, cognitivos, sociais, culturais), se faz necessário olhar o produto nas mais diferentes perspectivas de seu prisma.

Aquino et al. (2014) relatam que uma rede veicular é uma rede ad-hoc (Vehicular Ad-Hoc Networks) composta por veículos que possuem equipamentos de sensoriamento e comunicação sem o e, por isso, durante o seu deslocamento, são capazes de obter informações sobre o ambiente que os rodeiam e de trocar mensagens entre si e/ou entre pontos de acessos distribuídos pela cidade. Os dados

coletados pelas redes veiculares podem oferecer, dentre outros serviços, informações sobre as condições das estradas, do tráfego e do clima, sobre o comportamento dos veículos e dos motoristas.

Tais informações são uteis para uma grande variedade de aplicações: emissão de alertas de segurança, assistência ao motorista e roteamento de tráfego. Além disso, essas informações podem ser usadas para criar um sistema de tráfego inteligente capaz, dentre outras coisas, de atualizar os ciclos dos semáforos, de indicar prováveis zonas de pedágio, de medir a quantidade diária de veículos nas estradas ou de disseminar informações personalizadas aos motoristas.

Silva et al. (2016) diz que o Diagrama de Voronoi possui diversas aplicações em situações que se têm um plano, pontos nesse plano e deseja-se entender as relações de proximidade entre esses pontos. O teorema é aplicado e forma um polígono cujas distâncias entre os limites da célula e o núcleo que a originou sejam os menores possíveis. O diagrama é geralmente usado para análises territoriais e raios de influência, podendo incorporar parâmetros quantitativos e informações geográficas sobre o plano estudado. Neste caso, foi reconhecido o potencial para a aplicação do Diagrama de Voronoi sobre a malha viária existente com o objetivo de prever a distribuição de estruturas de apoio à rede ciclo viária, como estações de empréstimo e manutenção de bicicletas.

Silva et. al. (2016) relata que *Minimal Paths System* ou “sistema de caminhos mínimos” faz parte de uma lógica dos processos de conexões, diretamente relacionado a padrões de ocupação. Difere do sistema de caminhos diretos porque considera possíveis desvios que podem ser originados pelos fatores de atração entre os pontos desejados. O sistema de caminhos mínimos investigado por Frei Otto desde a década de 1960 se mostra aplicável no caso das rotas ciclaveis, pois permite ensaios sobre o uso da superfície de maneira eficaz e que também pode considerar variável como declividades, densidades e barreiras.

> A informática, a engenharia de tráfego, a eletrônica, a tecnologia comportamental e a democratização da informação são ferramentas essenciais e que modernamente compõem o que se chama de trânsito inteligente. A informática está cada vez mais potente, barata e amigável ao usuário. É inadmissível que São Paulo tenha apenas 25% de semáforos inteligentes. A eletrônica está aí para ser implantada. Painéis de mensagens

> variáveis e circuito de televisão são essenciais. O monitoramento eletrônico do trânsito em São Paulo deu os primeiros passos no pioneirismo dos semáforos coordenados e eletronicamente controlados a partir de 1982. O que falta é escala e atualização tecnológica. Afinal, em informática, um equipamento de 20 anos é quase pré-histórico. (SCARINGELLA, 2001)

1.3 Desenvolvimento de aplicativos móveis

Segundo Batista e Bazzo (2015) o desenvolvimento de aplicativos para dispositivos móveis tem fascinado muitos apreciadores aos cursos técnicos e superiores em tecnologia da informação. Esse interesse aparece pelo fato da crescente expansão deste mercado e por meio desses aplicativos, a possibilidade de remuneração, e que é notável o grande crescimento do mercado de desenvolvimento de aplicativos móveis. Em 2014, acumulou uma receita global de 13 bilhões de dólares, com um aumento de 100% ao ano, a Apple foi responsável por 67% do valor, e o Google pelos outros 33%, conforme relatório de AppAnnie (2015) citado por Batista e Bazzo.

Para Daniela Couto Carvalho Barra et al. (2017) apps são considerados como um conjunto de ferramentas utilizados para a realização de funções e trabalhos específicos e que os smartphones (dispositivos móveis), em especial, os aplicativos móveis, destinam-se a atender o acesso das pessoas à informação e ao conhecimento, sem restrição de tempo e espaço.

Segundo Tibes et al. (2014) em 2012, mais de 40 bilhões de aplicativos foram baixados nos smartphones e a previsão é de que em 2016 esse número chegue a 300 bilhões devido à facilidade com que os aplicativos podem ser acessados em suas respectivas lojas virtuais. Dessa forma, desenvolver soluções em forma de aplicativos móveis representa um meio eficaz de disponibilizar a ferramenta e atingir o público-alvo desejado.

Segundo Oliveira e da Costa (2012) a aquisição de aparelhos móveis por profissionais de saúde é muito alta, variando de 45-85% segundo uma revisão de Garritty e colaboradores citada. Apoiado nisso, ideias para fornecer conteúdo a este setor tem alcançado bastante popularidade. Na AppStore da Apple, versão americana, 8005 aplicativos na categoria Medicina são encontrados, número semelhante ao encontrado na plataforma de distribuição do sistema Android.

Segundo Melo e Carvalho (2014) o quadro dos aplicativos para smartphones pode ser facilmente associado à experiência com os desktops que vem ocorrendo, onde softwares nativos foram substituídos por softwares web, logo, é sensato supor a mudança de foco para o desenvolvimento multiplataforma.

Souza et al. (2013) desenvolveram um aplicativo na área de odontologia para pacientes com necessidades especiais chamado "OdontoPNE", para os autores “A construção do aplicativo por profissionais da área da saúde são possíveis desde que sejam aliados a profissionais específicos da área de desenvolvimento tecnológico, agregando ao produto final maior qualidade ao usuário.”

Segundo Mello e Sganzerla (2013) cada vez mais, aumenta a necessidade de aplicativos e softwares com o principal objetivo de inclusão, não apenas no meio digital, mas, também, que permitam e que ocasionem em uma independência destes indivíduos na vida real.

Segundo Wikitude (2012) citado por Galvão e Zorzal (2012) há um aplicativo chamado de Wikitude World Browser desenvolvido pela empresa Wikitude, que é capaz de encontrar, por exemplo, restaurantes, bancos, bares, casas de shows entre outros estabelecimentos apenas com a câmera do seu dispositivo móvel (smartphone), permite também mudar o que você está vendo, acrescentando novas imagens na tela, colocando imagens reais e objetos virtuais, dessa forma gerando uma imagem virtual.

Segundo Oliveira et al. (2014) para o desenvolvimento de aplicativos com reconhecimento de voz, é fundamental a utilização de um *“engine*” de voz (reconhecedores, sintetizadores ou ambos). Tendo um *engine* disponível para a língua alvo, uma API *(“application programming interface”)* torna o trabalho do programador menos trabalhoso durante o desenvolvimento.

De acordo com de Paula (2013) nos últimos tempos, com o aumento do número de smartphones sendo utilizados, alguns aplicativos feitos para sistema operacional Android tem sido desenvolvido também para a agricultura, sendo um deles, o “EpiCollect”, que é um aplicativo desenvolvido que dá a possibilidade de coletar dados, em sua maioria relacionados a epidemiologias, utilizando algumas funções como GPS e mapas (Google Maps).

Conforme Melo e Carvalho (2014) nos últimos cinco anos surgiram, no Brasil, algumas iniciativas que sugere a oportunizar e ampliar o uso pedagógico dos

dispositivos móveis em ambientes formais e informais de ensino e aprendizagem. Uma dessas iniciativas é o desenvolvimento de diversos aplicativos móveis para a educação livres para uso em dispositivos móveis *(smartphones).*

> Ao lado dos aparelhos móveis, crescentemente sofisticados e tecnologias de comunicação cada vez mais potentes em termos de capacidade de transmissão, permitindo tráfego de imagens e vídeos de alta resolução, tem-se tecnologias como o ambiente Java, cada vez mais presente no dia-a-dia seja em produtos fixos ou móveis e que apresentam entre outras características, alguma forma de comunicação e elevado grau de sofisticação em relação às tarefas que executam. [...] A expectativa de utilização de Java é grande. O usuário pode personalizar seu aparelho fazendo a cópia de aplicativos e com a possibilidade de atualização do software ser permanente, sem a necessidade de investir num novo aparelho a cada lançamento de nova aplicação. A plataforma Java é uma excelente escolha para o desenvolvimento sem fio por muitas razões. Uma delas é a segurança. [...] outra é o encorajamento à programação robusta com os 15 mecanismos da linguagem para o tratamento de exceções e o garbage collector que facilita a programação com o controle automático de objetos não mais referenciados na memória. (PALUDO, 2003).

Segundo Blois et al. (1999) a crescimento da tecnologia “Java” gerou o surgimento de novas formas de desenvolver aplicações para a Web. Estes aplicativos apresentam várias partes e podem ser organizadas na forma de componentes de *software*, aproveitando o potencial oferecido pela aplicação dos conceitos de orientação a objetos. Segundo o autor o *software* é um conjunto de componentes ligados que trabalham juntos para que seja possível o funcionamento de todo o ambiente.

Nas suas últimas versões a linguagem de programação Java introduziu um mecanismo de criação de classes Java que funcionam como programas CGI. Este mecanismo consegue ampliar das funções dos servidores Web que possuem a API dos servlets. De acordo com o autor “Com esta tecnologia, aplicações para a Web escritas em linguagens como o C e Perl podem ser portadas para a linguagem Java”.

Conforme Gomes e Melo (2013) há uma ferramenta chamada de *App Inventor for Android* que é um ambiente visual utilizado para programação em blocos, o App permite o desenvolvimento de aplicativos para dispositivos móveis com a plataforma Android de uma maneira mais simples, principalmente se comparada às linguagens

de programação tradicionais. De acordo com Campos (2011) citado por Gomes et al. (2014) o "'MIT App Inventor do Instituto de Tecnologia de Massachusetts é um *software* de plataforma web que permite aos seus usuários que desenvolvam, testem e exportem seus aplicativos para a plataforma *mobile* Android*."*

De acordo com Gomes e Melo ele possui um ambiente gráfico que possibilita o ensino de conceitos de lógica de programação de uma forma mais amigável e motivadora para estudantes do ensino médio e do ensino superior. De acordo Silva et al. (2014) na perspectiva da educação inclusiva, a tecnologia tem o foco em beneficiar a atuação do aluno com deficiência nas diversas atividades do cotidiano escolar. Por exemplo, um dos primeiros esforços de desenvolvimento de um conjunto de aplicativos móveis para este público alvo foi o GBraille Suite, que tem a função de auxiliar a prática da escrita braille e sua inserção nas práticas de escrita de português. Ao que estudos indicam esta linha de pesquisa tem um promissor potencial e que as experiências podem ser aumentadas, também, para mais participantes.

Segundo Nichele e Schlemmer (2014) os dispositivos móveis como *tablets e smartphones*, que possuem conexão sem fio e interface sensível ao toque junto com diferentes aplicativos têm proporcionado mudanças na forma de nos relacionarmos com a informação e produzir conhecimento. Devido ao aumento desses dispositivos desenvolvimento de novos aplicativos (Apps) tem aumentado, e muitos voltados para incrementar o aprendizado dentro e fora da sala de aula. Segundo os autores há vários tipos de Apps disponíveis já, dentre esses tipos estão jogos, mídias sociais, livros (entre eles dicionários, enciclopédias), revistas, aplicativos específicos para educação, para gerenciamento e organização de atividades e processo entre outros.

Nos desenvolvimentos de aplicativos é muito comum em alguns apps a utilização do GPS, entretanto, ele apenas pode ser utilizado para aplicações *outdoor*, segundo Rubinsztejn et al. (2004) nos últimos anos foram desenvolvidos diversos "sistemas de localização indoor baseados em crachás ativos ou passivos, bem como tecnologias que inferem a localização baseada em sinais de rádio frequência de pontos de acesso W-LAN. "

> Ao criar aplicativos que possam ser utilizados em seus smartphones e que incorporam funcionalidades específicas de dispositivos móveis como a mobilidade e o uso do sistema de geolocalização são desenhadas conexões reais entre o cotidiano do aprendiz e os conceitos aprendidos. Desta forma, os conceitos não apenas fazem sentido para os educandos, mas têm

> importância, pois a possibilidade de criar aplicativos reais os deixa mais motivados. [...] A premissa construcionista ressalta o aprendiz como participante ativo, condutor do seu processo de aprendizagem: experimentando, descobrindo, errando e depurando, mediante o ciclo de descrever- executar – refletir – depurar. Desta forma, ao invés de apenas fornecer o conhecimento ao aluno e levá-lo a fornecer respostas, o computador recebe as instruções e operações que devem ser executadas para produzir as respostas desejadas pelo aluno. (ALMEIDA, 2000).

Segundo Souza et al. (2013) uma possível dificuldade a ser encontrada no desenvolvimento de aplicações móveis é a necessidade de incorporar e adaptar os módulos do aplicativo às necessidades de pacientes e profissionais que sempre mudam. Logo o desenvolvimento de aplicativos, precisam manter a atualização e uma interface operacional com estas demandas, apesar de, segundo o autor “a diversificação conceitual e metodológica estar entre os motivos que fazem com que o desenvolvimento e a disponibilização desses instrumentos nem sempre apresente o mesmo ritmo que a sua necessidade. ”

> A adoção de aplicativos móveis (apps) no contexto educacional vem crescendo e abre espaço para o *mobile learning* (*m-learning*), que envolve o uso do dispositivo móvel, sozinho ou em combinação com outras tecnologias de informação e comunicação (TICs) para promover o aprendizado. Diante dessa crescente utilização e dos diferentes perfis de usuários, é relevante garantir que esses aplicativos contemplem requisitos de usabilidade, uma vez que eles possuem objetivos de aprendizado. (D’CARLO et al., 2016)

Caron, Biduski e De Marchi (2015) relatam que, sobre um aplicativo para ajudar pessoas com Alzheimer, ao projetar um jogo para pacientes com este tipo de doença, é necessário considerar alguns aspectos importantes, dentre eles estão a organização das informações, a facilidade de interação, o tamanho das letras e o uso de cores específicas. Segundo os autores, para desenvolver um aplicativo é preciso tomar alguns cuidados, tais aspectos além de produzirem efeitos de melhoria na facilidade de uso, estão relacionados com o estímulo maior da memória e compensam de habilidades que são afetadas pelo quadro patológico, como a compreensão das informações por exemplo.

1.4 Cidades inteligentes

Segundo Weiss, Bernardes e Consoni (2017) o ponto de entrada das cidades no universo das cidades inteligentes se diferencia pelas suas características: geográficas, culturais, políticas e socioeconômicas. Também se notou que sistemas de computação e comunicação de cidades como Rio de Janeiro, Porto Alegre e Curitiba estão cada vez presentes e inteligentes, conectando os cidadãos, as empresas e o próprio governo.

De acordo com uma adaptação de Wolfram (2014) citado por Benites (2016) levando em conta diversos fatores para definir, no cenário atual, o título de "*smart*" a uma cidade ou municio, a definição mais simples de cidade inteligente seria então: "uma cidade que aproveita, de forma inovadora, todo o potencial de sua infraestrutura digital moderna no compromisso com o desenvolvimento sustentável amplo e a resiliência".

Amorim (2016) relata que O conceito de *Smart City*, *SmarterCity* ou "cidade inteligente" é sutil e controverso, existindo uma grande variedade de termos para expressar conceitos semelhantes. No que pese diversos autores terem cunhado diferentes termos dando ênfase neste ou naquele aspecto, no fundo, esses conceitos possuem praticamente o mesmo significado.

> As cidades são consideradas inteligentes quando são identificadas contendo investimentos inteligentes ao longo dos eixos: economia, mobilidade, meio ambiente, recursos humanos e estilos de vida inteligentes. Os significativos avanços tecnológicos e das tecnologias da informação e comunicação (ICT) agora fazem das plataformas tecnológicas embarcadas um instrumento potencialmente significativo para sensorizar e monitorar a funcionalidade e o desempenho das cidades, permitindo ampliar sobremaneira suas capacidades de gerenciar recursos com mais eficiência e prover conectividade e informações de forma transparente aos seus cidadãos e visitantes. Estas estratégias permitem também que se compreendam melhor os custos financeiros e ambientais de seus próprios consumos. Torna-se assim possível que os gestores urbanos criem novos serviços e melhorem aqueles já existentes coletando e analisando informações sobre infraestruturas essenciais, como energia, água, transporte e saúde, entre outros de interesse da comunidade local. (C40 SÃO PAULO CLIMATE SUMMIT, 2011, p. 32)

De acordo com Lemos (2013) com o conhecimento acessível e mais próximo dos cidadãos em suas atividades do dia a dia, elas podem além de ter uma melhor percepção do espaço onde vivem, mas também pensar em alternativas criativas e inovadoras para as suas cidades. Como por exemplo, o uso da computação em nuvem, *Big Data e Internet das Coisas*, associado ao Open Data, pode ajudar no trânsito trocando informações em tempo real de pessoas, semáforos e etc, no controle da poluição ambiental como sensores de CO2 ou de ruído que conversam com aplicativos de celular, no uso mais eficiente da eletricidade com tecnologia *smart grid*, onde objetos são auto programados para poupar energia durante o seu funcionamento, entre outros.

> Mas o que é afinal uma cidade inteligente? Em uma definição ampla, trata-se de uma visão de desenvolvimento urbano que tenta integrar tecnologia de informação e comunicação e a chamada Internet das Coisas (conexão em rede de dispositivos e objetos do cotidiano entre si, com os usuários e com a internet para troca de informações em tempo real). Tudo isso para gerenciar de forma mais racional uma cidade, oferecendo uma melhor sustentabilidade e qualidade de vida a todos os seus cidadãos. Basicamente, conta-se com tecnologias que integram dados e informações dinâmicas, muitas vezes colhidas em tempo real, para planejar, gerir recursos e oferecer serviços aos cidadãos de forma mais eficiente. (PACHECO, 2017– Jornal da USP).

De acordo com Fábio Duarte (2005) algumas cidades buscam uma série de estratégias para serem as iniciadoras de inovações tecnológicas na sociedade de informação, trabalhando junto órgãos públicos e empresas privadas, até mesmo Faculdades e universidades. Dessa forma, é possível concluir que a criação de polos tecnológicos é um dos primeiros arranjos urbanos próprios da sociedade da informação. Antigamente a maior parte desses polos eram instalados nas periferias dessas cidades, se assemelhando muito a parques industriais, hoje em dia os polos de inovação, espontâneos ou induzidos, se concentram mais em áreas urbanas onde tem maior infraestrutura tecnológica, social, econômica, cultural e científica

Aquino et al. (2014) relatam que uma rede veicular é uma rede *ad-hoc (Vehicular Ad-Hoc Networks)* composta por veículos que possuem equipamentos de sensoriamento e comunicação sem o e, por isso, durante o seu deslocamento, são capazes de obter informações sobre o ambiente que os rodeiam e de trocar mensagens entre si e/ou entre pontos de acessos distribuídos pela cidade. Os dados

coletados pelas redes veiculares podem oferecer, dentre outros serviços, informações sobre as condições das estradas, do tráfego e do clima, sobre o comportamento dos veículos e dos motoristas.

Tais informações são uteis para uma grande variedade de aplicações: emissão de alertas de segurança, assistência ao motorista e roteamento de tráfego. Além disso, essas informações podem ser usadas para criar um sistema de tráfego inteligente capaz, dentre outras coisas, de atualizar os ciclos dos semáforos, de indicar prováveis zonas de pedágio, de medir a quantidade diária de veículos nas estradas ou de disseminar informações personalizadas aos motoristas.

Segundo Maria Alexandra Cunha et al. (2016) para as cidades chegarem a ser consideradas *smart city*, as cidades precisam superar diferentes barreiras. Cada uma terá desafios e obstáculos diferentes devido a diferença de sua identidade, história, geografia e cultura. Mas é possível identificar um conjunto de problemas comuns. De acordo com os autores os especialistas apontam seis frentes de trabalho para passar da estratégia à execução, dentre elas estão:

> 1. Assegurar uma liderança clara e uma gestão com capacidade de execução e com visão transversal das ações definidas, [...]”
> 2. Ter uma visão compartilhada e consensual de cidade e um plano de ação de longo prazo, [...].
> 3. Estabelecer um novo modelo de relações entre administração pública e empresas, [...].
> 4. Incorporar uma solução tecnológica aberta e padronizada, horizontal, inter operável e com escalabilidade, [...].
> 5. Promover modelos de financiamento com participação privada, [...].
> 6. Desenvolver modelos de negócios sustentáveis e com retorno para todos os agentes envolvidos, [...].

De acordo com Afonso et al. (2015) atualmente existem várias formas de medição e tipos indicadores em todo o mundo que tem a utilidade de categorizar Cidades Inteligentes, entretanto, não existe um número de estudo suficientes para fazer uma comparação destas cidades no Brasil. Segundo os autores, as bases de dados públicos dispõem de dados de diferentes indicadores e domínios, e esses dados precisam ser normalizados e agrupados para que possa ser possível fazer uma comparação entre as cidades inteligentes brasileiras. Fazer uma análise fundamentada no agrupamento de similaridade de indicadores de cidades inteligentes

pode acarretar aos gestores dos municípios uma compreensão das possibilidades estratégicas de otimização de recursos.

Conforme Guimarães e Xavier (2010) a tradução de cidade inteligente para a língua portuguesa é muito estudada hoje como um fenômeno contemporâneo do urbanismo, um modo de viver definido pela grande implantação cidadã dos indivíduos no espaço da cidade onde habita e realiza as suas atividades cotidianas, com a maior eficiência e qualidade de vida possível, atendendo satisfatoriamente as suas demandas de deslocar de um local ao outro, trabalhar, se comunicar e também de se relacionar com o meio ambiente.

Para Andrade e Galvão (2016) O desenvolvimento de uma cidade inteligente, ou *smart city*, leva em conta o fato de que a tecnologia é fator indispensável para que as cidades possam se modernizar e ter uma melhor infraestrutura para a população. Ademais, esse conceito tem se mostrado muito importante no processo de se tornar os centros urbanos mais eficientes e de oferecer uma qualidade de vida melhor e gestão de recursos naturais por meio de um processo participativo.

Conforme Weiss, Bernardes e Consoni (2013) é notável uma rede de conexões diretas com os sistemas eletrônicos e eletromecânicos que vem crescendo em edifícios, eletrodomésticos, máquinas de produção, plantas de processos industriais, sistemas de transporte, redes de geração e distribuição de energia elétrica, sistemas de fornecimento e tratamento de água, remoção de resíduos, sistemas de monitoramento e manutenção de funções vitais em hospitais, segurança pública e sistemas de gerenciamento para praticamente todas as atividades humanas.

Segundo Farias et al. (2011) a base par a construção de uma cidade inteligente é a existência de uma infraestrutura tecnológica que também é inteligente, isto é, a propagação pelo ambiente urbano de instrumentos eletrônicos para aquisição, tratamento e transmissão de dados. E tem como ponto mais alto disponibilizar serviços inteligentes para os usuários dos serviços urbanos como pessoas, administrações públicas, empresas, entre outros.

> Assim, para a existência de tais lugares, férteis em tecnologia, deve haver necessariamente um cruzamento de inúmeros fatores pertencentes a todas as esferas da sociedade criando uma condição mínima para que o espaço se torne produtor de tecnologias. No caso das cidades da inteligência onde há uma significativa concentração de espaços ligados diretamente ao processo produtivo, isto é, em que a vocação é a de produção tecnológica, existe uma

> maior demanda para a existência e um bom relacionamento de alguns elementos da produção científica e tecnológica, como universidades ou centros de pesquisa, empresas, e uma integração com os poderes políticos locais. Assim, são lugares onde notamos a presença marcante de espaços como incubadoras de empresas nascentes, PMEs e parques tecnológicos, responsáveis diretos pela produção de tecnologia. (FIRMINO E CAMARGO, 2014)

Segundo Strapazzon (2009) as cidades inteligentes, ou *smart cities* são a etapa mais avançada da relação entre convergência tecnológica, gestão de cidades, qualidade de vida e competitividade econômica. Depois que a cidade de Singapura ganhou o prêmio de cidade inteligente do ano, em 1999, o tema começou a ganhar mais destaque. Após o ocorrido, o tema começou a ser estudado com seriedade como alternativa viável para cidades de médio e pequeno porte

De acordo com Komninos (2007) A cidade inteligente é um sistema que pode ser divido e distribuído em três níveis, são eles:

“Nível I: Constitui o nível de base, compreendendo as atividades com elevada intensidade de conhecimentos da cidade. Trata-se de atividades de transformação e de prestação de serviços que se organizam em conglomerados e bairros (clusters). “

“Nível II: Compreende os mecanismos institucionais de colaboração social para a aprendizagem e a inovação: instituições e mecanismos de informação estratégica, de avaliação comparativa, financiamento do risco, transferência de tecnologia [...]”

“Nível III: O terceiro nível compreende as ferramentas digitais e aplicações de apoio à inovação, que criam um ambiente virtual de manipulação da informação e dos conhecimentos. Este nível refere-se ao sistema de inteligência artificial. ”

Para Rezende (2012) projetos de cidades inteligentes organizam e disponibilizam informações, sistemas e serviços para seus gestores e cidadãos. “O planejamento de informações municipais requer a participação dos servidores municipais e dos cidadãos, sejam trabalhadores, estudantes, aposentados, donas de casa, vereadores, empresários entre outros.”

De acordo com Wolfram (2012) citado por Prado e Santos (2014) há uma tendência de centralização de pesquisas relacionadas à inteligência das cidades, em sua maioria relacionadas aos conceitos de *smart* e *intelligent cities*, que são muitas vezes utilizados como sinônimos, entretanto, há confusões como no que diz respeito a conceito como de *digital cities*. Neste caso, não há uma definição muito clara do que

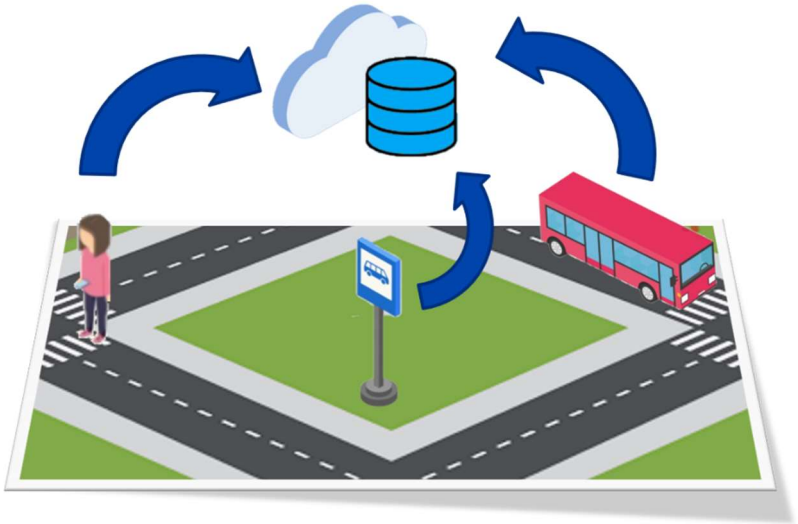
é uma *Smart City*, logo, há uma carência de um conceito único, dificultando o desenvolvimento de iniciativas concretas e específicas que visem alcançar este novo modelo de urbanização, também ocorre o fato de a maior parte da literatura disponível abordar apenas os aspectos tecnológicos causando a escassez de fontes de consulta relativas a seus demais componentes, tanto estruturais como institucionais.

3. DESENVOLVIMENTO

Decorrente do embasamento na fundamentação teórica, deu-se início o desenvolvimento do projeto seguindo algumas etapas pré-planejadas pelo orientador e o orientado. A proposta deste trabalho é desenvolver dois aplicativos diferentes, um deles para ser usado pelos motoristas de ônibus e outro para ser utilizado pelos usuários comuns com deficiência visual, também a plataforma *webservice* que fará a troca de dados fazendo a comunicação com um banco de dados.

O projeto funcionará da seguinte maneira: O motorista de ônibus enviará sua localização atual para o banco de dados a cada 50 metros automaticamente através de seu aplicativo, o usuário com deficiência visual enviará sua localização para o servidor a cada 5 metros aproximadamente por meio de seu aplicativo também e através de um site o ponto de ônibus terá sido cadastrado e então o servidor terá a localização de todos os elementos, ou seja, motorista de ônibus, usuário com deficiência visual e o ponto de ônibus, a Figura 1 representa este envio de informações para o servidor.

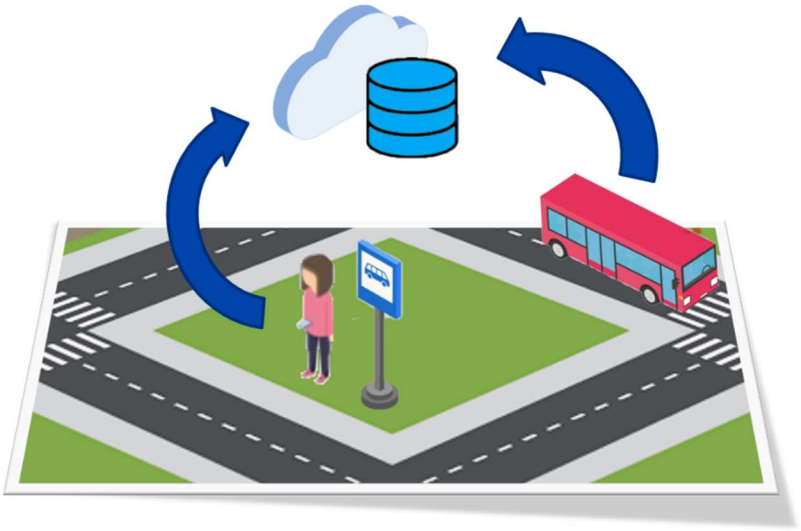
Figura 1 - Representação do envio das posições para o banco de dados



Fonte: Do próprio autor, 2019

O aplicativo do usuário deverá guiar o usuário até o ponto de ônibus para que então, apenas quando ele estiver próximo ao ponto como mostra a Figura 2, deverá ser feito uma busca pelos ônibus próximos a ele.

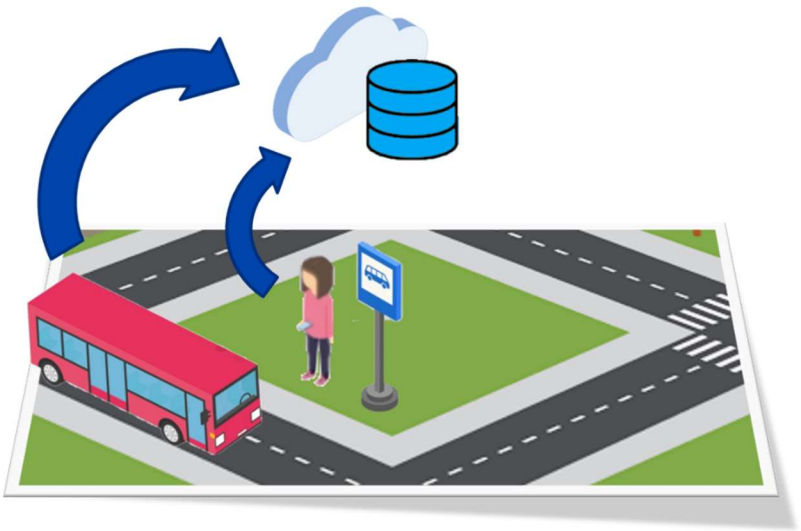
Figura 2 - Representação do usuário próximo ao ponto de ônibus



Fonte: Do próprio autor, 2019

Para evitar que o usuário perca o ônibus devido a algum atraso na resposta do servidor e então não seja notificado a tempo que o ônibus está chegando, foram estabelecidos alguns raios em volta dos pontos de ônibus maiores para a busca dos ônibus, dessa forma, será notado o ônibus se aproximando e mesmo que a resposta do servidor demore, o ônibus já terá sido encontrado antes. Caso o ônibus esteja próximo ao ponto de ônibus e o usuário esteja no ponto de ônibus também, como mostra a Figura 3, o servidor analisará o nome atual do ônibus.

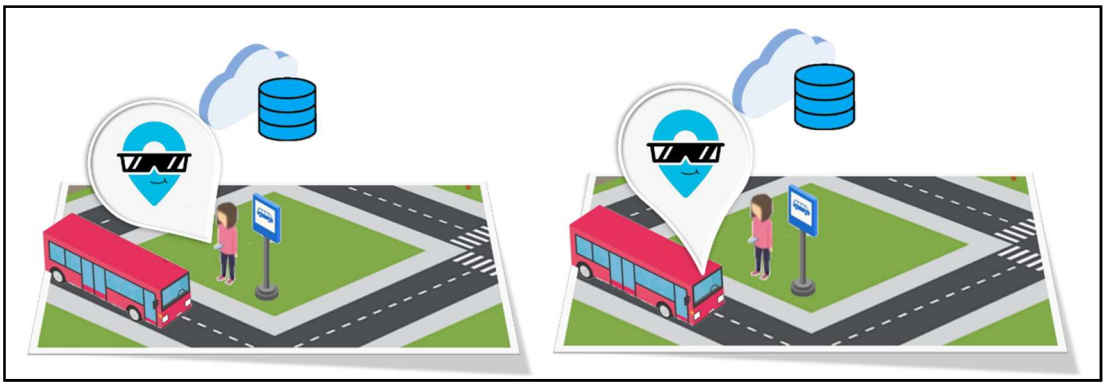
Figura 3 - Representação do ônibus próximo ao ponto de ônibus



Fonte: Do próprio autor, 2019

Caso seja verificado pelo servidor que o nome atual do ônibus é o que a pessoa precisa embarcar então será notificado o usuário que o ônibus está chegando e o motorista que há um usuário com deficiência visual no ponto de ônibus, como mostra a Figura 4, o mesmo deverá ser aplicado para o aplicativo da estação de trens e metrôs caso a pessoa esteja próximo à estação.

Figura 4 - Representação das notificações



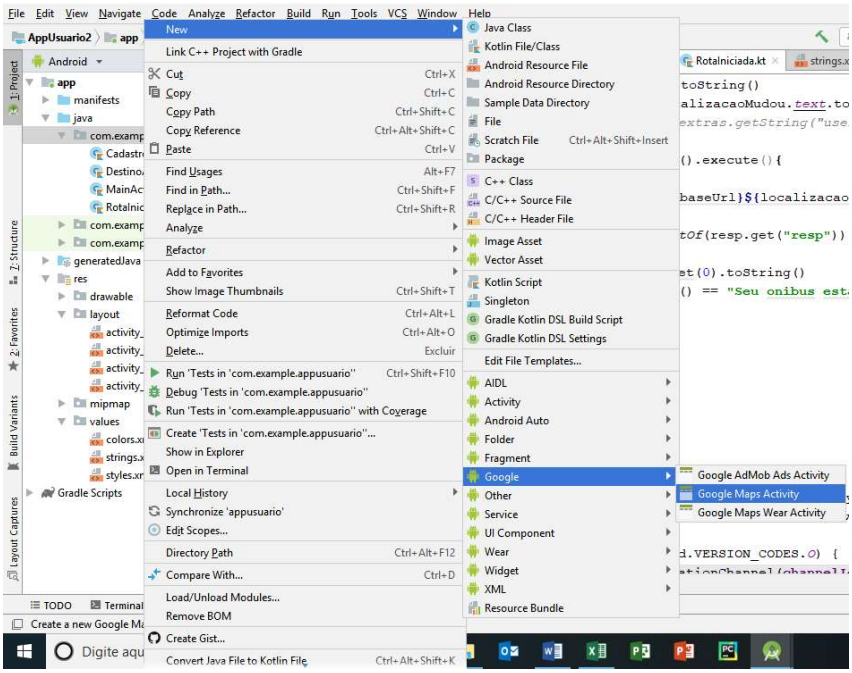
Fonte: Do próprio autor, 2019

Com isso, a primeira etapa do projeto consistiu na criação de uma primeira tela de um aplicativo simples para a aplicação do usuário, contendo apenas uma tela vazia e uma caixa de texto sem função específica.

Ao término, deu-se início à implementação dos mapas na tela principal do aplicativo, para isso, foi utilizado como base para as implementações de mapa o *Google Maps API.*

Para o uso desta API é necessário seguir algumas etapas. Existem diversas formas possíveis de se conseguir utilizá-la, neste trabalho será apresentado um desses caminhos. O primeiro passo foi clicar sobre a pasta onde se encontra a *Activity* do Aplicativo, depois selecionar a opção “*New”* e no segundo palete aberto clicou-se na aba descrita como “Google” e então foi selecionado a opção Google *Maps Activity*, conforme a Figura 5.

Figura 5- Demonstração do caminho para criar uma nova *Google Maps Activity*



Fonte: Do próprio autor, 2019

Após ter sido escrito o nome da *Activity* que seria criada, foi finalizado o processo de criação das *Activitys,* depois do Android Studio ter sincronizado e finalizado os carregamentos, selecionou-se o arquivo gerado ".xml" do *Google Maps API.* Dentro deste arquivo foram gerados automaticamente alguns comentários, dentre eles, há um link que foi copiado e pesquisado em um servidor da internet. A URL gerada é um caminho que leva até uma página do Google para registrar o aplicativo que estava sendo criado, sendo possível o acesso desta página.

Para o registro do aplicativo foi necessário fazer o login em uma conta de e-mail. Finalizado o login nesta página, gerou-se uma chave única para o aplicativo que é um conjunto de algarismos alfanuméricos. Esta chave foi copiada e então foi substituída dentro da aplicação, no mesmo arquivo ".xml" onde, por padrão vem escrito *"YOUR_KEY_HERE",* feito isso, foi sincronizado o projeto, após o sincronizado, o aplicativo já estava apto de usar mapas do Google em suas telas.

Foi compilado e testado o aplicativo em todo seu desenvolvimento em um smartphone Moto G 5S Plus, pois houve problemas com os emuladores do Android Studio. O Mapa apresentado apresentava uma localização fixa, porém, o aplicativo apresentado neste trabalho devia ter a função de acompanhar o usuário que estivesse utilizando-o durante seu deslocamento.

O próximo passo deste projeto foi iniciar a implementação da recuperação da localização do usuário em tempo real, para isso, foi utilizado com base do desenvolvimento os tutoriais disponíveis no site do Android, que contêm algumas etapas para a implementação de, como por exemplo, como recuperar a última localização do usuário etc.

No desenvolvimento da recuperação da localização do usuário foram encontrados problemas. Após ser implementado os métodos descritos no tutorial, ao iniciar o aplicativo, em sua tela, era mostrado apenas um mapa sem zoom com a Latitude zero e Longitude zero, como mostra a Figura 6, sem nenhum marcador de localização na tela, apenas após um período longo de tempo que variava de 10 a 40 minutos, que então a recuperação de latitude e longitude do smartphone utilizado era recuperada e mostrada na tela através de um marcador vermelho.

Figura 6 - Tela do aplicativo sem marcador de posição

Fonte: Do próprio autor, 2019

Tais problemas encontrados inviabilizaram a utilização deste método de recuperação de latitude e longitude atuais do usuário, portanto, após uma série de pesquisas foram encontrados outros métodos de implementações de recuperação de geolocalização de um usuário.

Estes métodos foram implementados e testados, porém ainda sem efeito, a recuperação do usuário continuava levando muito tempo para ser mostrada na tela, por causa deste problema, em conjunto, decidiu-se continuar a implementação das demais funcionalidades do aplicativo e retornar para solucionar esta problemática em uma etapa posterior do projeto para que não fosse atrasado o desenvolvimento do trabalho de conclusão de curso então, deu-se continuidade no projeto.

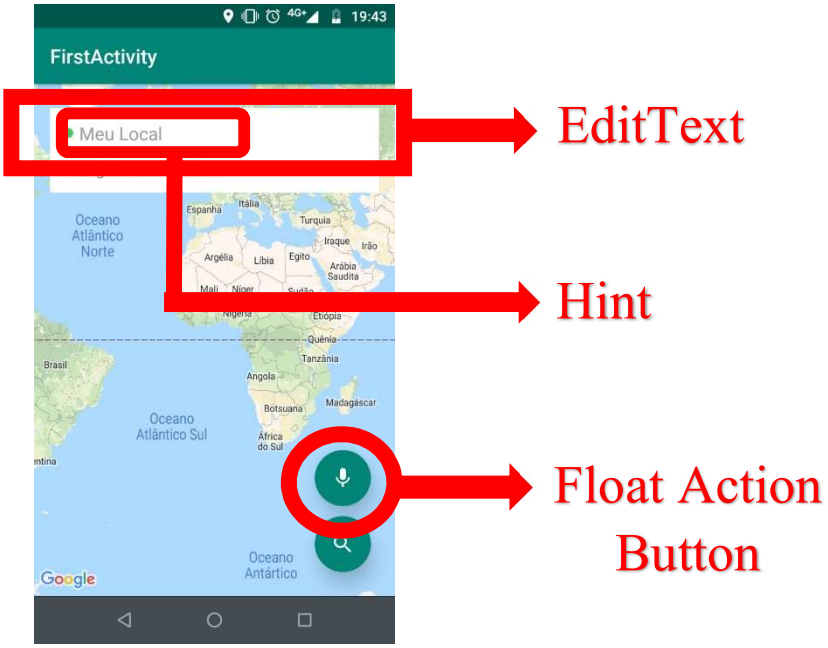
A etapa decorrente do trabalho foi a implementação dos componentes da tela manipuláveis pelo usuário, dentre eles estão a criação do campo de texto onde seria preenchido pelo destino escolhido pelo usuário, botões de pesquisa e o componente de botão responsável pela ativação do microfone e do reconhecimento da voz e da fala da pessoa que estaria utilizando o aplicativo para se locomover até algum destino desejável. O primeiro componente implementado foi a caixa de texto descrita anteriormente.

Devido ao fato da apresentação do mapa na tela do aplicativo visível pelo usuário, durante a criação das *Activitys* foram criadas automaticamente dois arquivos ".xml" sendo uma delas contendo um *fragment* com o mapa e outro arquivo onde foram inseridos os componentes de tela. Para o campo do local de destino do usuário foi utilizado um *EditText*.

Como o propósito principal do aplicativo é ajudar pessoas com deficiência visual a se locomover de um ponto a outro no mapa, há atualmente aplicativos específicos que fazem a leitura da tela para pessoas que tenham tal deficiência, por este motivo, para este projeto partiu-se do princípio que o usuário já tenha em seu *smarphone* algum aplicativo móvel que faça a leitura da tela para ele.

Por isso, neste trabalho não foi implementado nenhuma função parecida pois não é o objetivo do projeto que foi estabelecido previamente, podendo ser um ponto a ser implementado em projetos futuros, dando continuidade a este projeto. A Figura 7 apresenta os componentes de *layout* utilizados na tela

Figura 7 – Identificação dos componentes da tela do aplicativo



Fonte: Do próprio autor, 2019

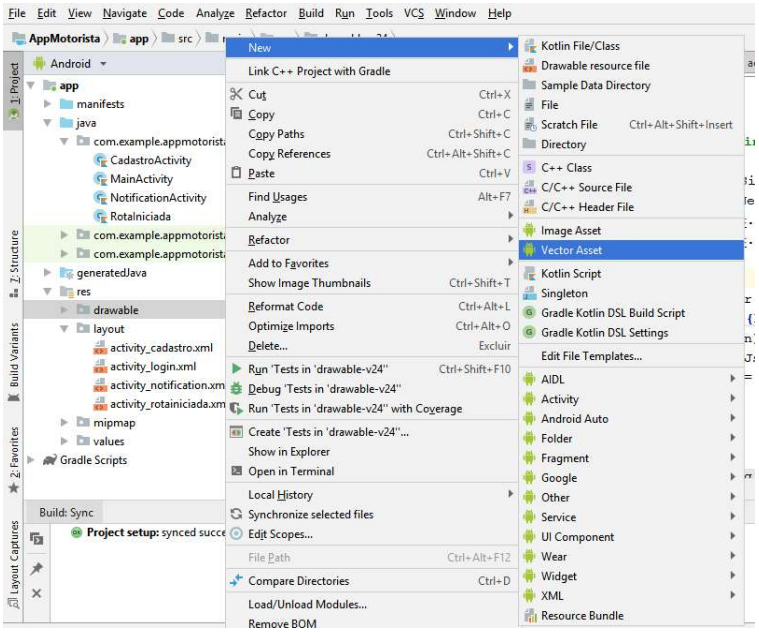
Então, para facilitar o entendimento, no *EditText* foi utilizado um *hint* (dica) com um texto de “Diga qual é o seu destino”. Para os botões, havia a possibilidade de criar um botão como um “*Button”* comum, porém, decidiu-se que os botões seriam um elemento de tela *“Float Action Button”*.

Foi criado um primeiro FAB par a função de pesquisa do destino escolhido pelo usuário, e depois um segundo botão para a ativação do microfone, após sua criação, deu-se início ao processo de atribuir ações para os botões.

Nos botões, para que fosse realizado alguma tarefa esperada foi adicionado um *Listener* para monitorar se há algum click no botão, como no projeto, a primeira tela o usuário possuía. Para adicionar um *listener* foi utilizado uma função “*setOnClickListener*” que, a partir da recuperação do *id* do componente da tela, foi chamado esta função, e dentro desta função foi onde adicionou-se as ações para eles, este processo foi feito para cada um dos botões.

O botão de pesquisa deveria ter a função de salvar o texto escrito no componente *EditText* e então enviar para um outra *Activity* secundária que seria aberta a partir do momento que o botão de pesquisa fosse acionado. Para criar o ícone do botão foi necessário criar e selecionar a pasta do *drawable* e com o botão direito selecionar *New Vector Asset* e então, foi escolhido o ícone de *search*, como mostra a Figura 8.

Figura 8 - Demonstração do caminho para criar um Vector Asset



Fonte: Do próprio autor, 2019

Para realizar a função de passar uma informação de uma Activity para outra utilizou-se a função *"Intent"* onde foi criado uma variável do tipo var que chamou a função Intent e então passado o contexto, que seria a *activity* de onde ela está e para onde ela deve ir. Depois, usando a função "*text*" que para o kotlin é o mesmo que as funções *"getText()"* e "*setText()"* em Java.

Ao término da criação desta intenção de troca de tela foi preciso adicionar a informação que ela iria levar, para isso utilizou-se a função "*putExtra*", esta função precisa de uma chave e a informação que ela vai enviar até a outra *Activity*, então, foi necessário através da variável da intenção criada chamar a função "*putExtra*" com a chave de "Destino" que será recebida na *Activity* descrita na função Intent que no caso foi a função "RotasActivity".

Com o botão da pesquisa funcionando deu-se início ao desenvolvimento do microfone, para isso, foi utilizado uma implementação do Google que já possui recursos "stt" (*Speech To Text*), foi utilizado como base de implementação um passo a passo do site do Google. Foi adicionado a ação de ligar o microfone e iniciar o reconhecimento da fala do usuário quando o botão fosse apertado que era monitorado através do *listener* criado.

Depois que o usuário falasse em voz alta o destino que ele desejava, a aplicação fazia a conversão da fala para texto e substituía o texto convertido no campo de destino criado anteriormente. Dessa forma a pessoa que utilizasse o aplicativo poderia tanto digitar o seu destino quanto apertar o botão do microfone e então falar em voz alta o local que ela desejava ir. Para o ícone do microfone foi feito o mesmo processo de criar um “Vector Asset”.

Ao término destas implementações, o próximo passo foi a conversão de um texto para uma latitude e longitude referente a um ponto no mapa equivalente ao destino desejado. Também foram utilizados recursos do Google retiradas principalmente do site do *Google Maps API* onde contém as implementações destas funções. A realização da conversão foi definida para acontecer quando o botão de pesquisa fosse acionado e após o usuário confirmar seu local de destino então aconteceria a troca de telas de uma *Activity* para outra usando a função “*Intent*” citada anteriormente.

A proposta do aplicativo é baseada na ideia de ajudar o usuário a se locomover pela cidade dando a ele maior autonomia e independência, por isso, foi definido pelo orientador e orientado que seria importante dar à pessoa que estivesse utilizando a opção de escolha de qual transporte ela gostaria de utilizar, público ou privado. Para o transporte de pessoas privado, foi realizado uma pesquisa entre dois aplicativos de taxi mais comuns utilizados hoje por brasileiros e após uma comparação chegou-se à conclusão de que a opção mais viável e rápida de ser implementada seria a criação de um botão que seria responsável por chamar uma viagem através do aplicativo de taxi Uber a partir do ponto onde o usuário estava até o ponto no mapa que o usuário confirmou após clicar no botão de pesquisa na tela anteriormente após ter confirmado seu local de destino.

A implementação do botão do UBER se baseou no próprio tutorial presente no site oficial da empresa, que contém um passo a passo para a criação de “*ride request button*” que é o botão feito pela empresa UBER para que seja realizado a requisição de uma viagem, e já contém funções prontas que precisam primeiramente, registrar o aplicativo no painel de desenvolvedor presente ainda no site oficial da empresa. Após registrado, já foi possível seguir o passo a passo que foi implementado em sua maioria, porém não foi possível atingir totalmente o que esperava devido a um problema decorrente do início do projeto.

A recuperação da localização atual do usuário ainda não estava implementada por esse motivo apenas foi concluído parte da implementação do botão, no caso, ao apertar o botão "*ride request button"* do Uber, criado, o aplicativo desenvolvido neste trabalho apenas redirecionava o usuário para a tela inicial do aplicativo Uber pois não era possível passar o parâmetro de posição atual do usuário. Em conjunto orientador e orientado, decidiu-se seguir para a segunda fase do projeto que consistia na lógica principal do projeto que foi a troca de dados entre usuário, motorista e estação de metrô/trem e login de usuários do aplicativo.

Iniciou-se então a segunda fase do projeto. O primeiro passo foi, em conjunto orientador e orientado, iniciar um esboço de como seria a troca de dados entre todos e como e onde seriam processados os dados. A primeira tentativa foi esboçar a estrutura da troca de dados utilizando o banco de dados do Google *Firebase*, entretanto, devido ao fato de haver uma limitação de recursos para se utilizar o *Firebase* sem a necessidade de pagar mensalmente pelo serviço, optou-se por não o utilizar para ser o banco de dados do aplicativo. Por isso, definiu-se que seria usado outro recurso, após debater algumas opções, decidiu-se que seria utilizado um *framework* escrito em python chamado *Flask* próprio para o desenvolvimento web.

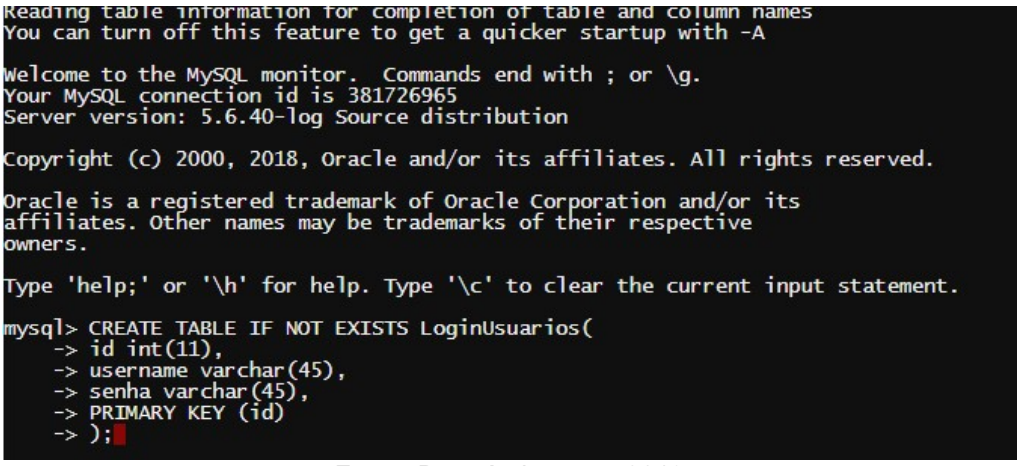
A princípio para a criação de um domínio que seria utilizado pelo aplicativo e/ou servidor em um computador ou notebook foi utilizado um software chamado Ngrok que é capaz de gerar uma URL para ser utilizada para ser o domínio para a troca de dados, porém utilizar este software inviabiliza a implementação do aplicativo em campo pois este *software* (Ngrok) apenas gera um domínio que é disponível enquanto o software estiver aberto, logo, no momento em que é fechado o domínio era necessário criar uma outra URL e adicionar novamente ao aplicativo.

Por este motivo, foi sugerido pelo orientador a utilização de um ambiente de desenvolvimento web feita em python chamado de PythonAnywhere, este ambiente permite que seja desenvolvido toda a estrutura e lógica do banco de dados. Há também consoles onde é possível digitar linhas de comando de forma mais rápida e simples, o PythonAnywhere gera uma URL que pode ser usada como domínio para troca de dados *Web* e diferentemente do Ngrok não varia a sua URL.

No PythonAnywhere, para estabelecer a comunicação entre o banco de dados criado no console e a lógica para realizar a troca de dados foi utilizado o mesmo framework *Flask.*

Afim de uma primeira familiarização com a estrutura e troca de dados entre o banco de dados, a primeira proposta foi implementar o login dos usuários, sejam eles motorista ou usuário comum, portando deu-se início, dentro do PythonAnywhere, a como estruturar a lógica de login e cadastro. Com isso, o passo inicial foi criar através do console do ambiente PythonAnywhere, uma tabela composta por *id*, "*username*" e senha chamada de LoginUsuarios. O *id* foi utilizado como a "*PRIMARY KEY*" e na sua criação estabeleceu-se que não seria nulo e iria incrementar automaticamente através da linha de comando "*NOT NULL AUTO_INCREMENT*" como mostra a Figura 9.

Figura 9 – Linhas de código para criar uma tabela no PythonAnywhere

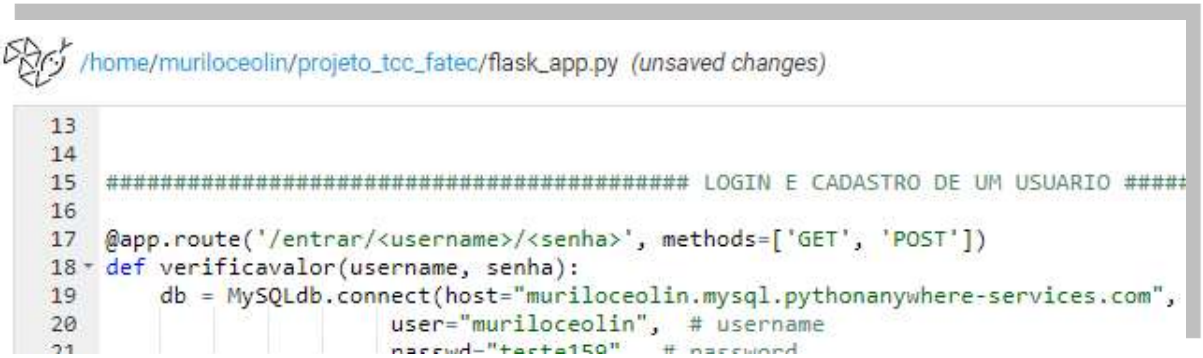


Fonte: Do próprio autor, 2019

Ao término da criação da tabela, dentro do arquivo python criado no PythonAnywhere para fazer a manipulação dos dados iniciou-se o algoritmo para a implementação do login e do cadastro.

A primeira etapa para a elaboração deste algoritmo é definir qual a rota do aplicativo, pois, basicamente, para realizar um determinado algoritmo é necessário um URL específico para que o aplicativo encontre no servidor está URL e então realize as linhas de comando descritas dentro daquela rota. A URL criada pode conter variáveis na sua extensão, para isso, é necessário definir o nome da variável que será passada através do caminho (entende-se caminho como o mesmo que URL) e deixa-la entre os sinais de menor e maior, depois, na função que será realizada quando aquela URL for lida deverão ser passados como parâmetros os mesmos nomes das variáveis que a URL contém, como mostra a Figura 10.

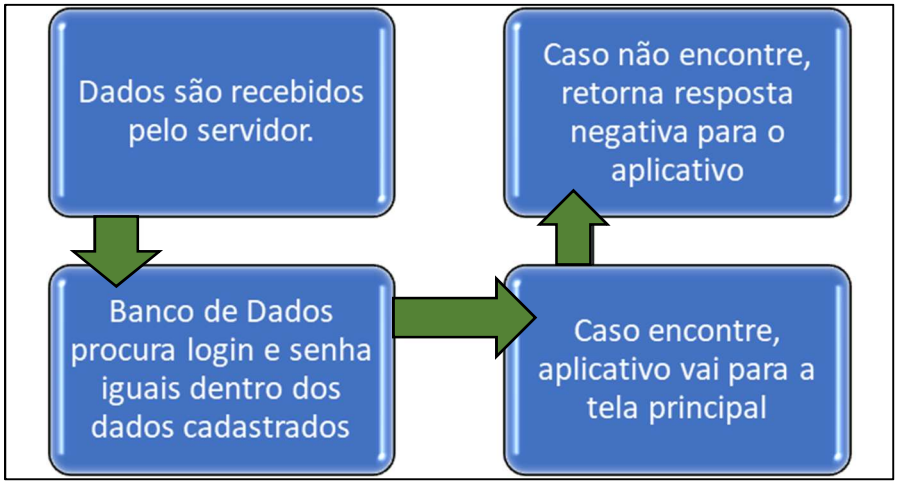
Figura 10 – Parte da rota no servidor que recebe os argumentos



Fonte: Do próprio autor, 2019

Para o *login*, foi realizado uma lógica simples para a verificação se há ou não um usuário com o nome digitado já existente no banco de dados conforme a Figura 11, foram passados através da URL duas variáveis, uma representando o nome do *username* eu o usuário digitar e a outra é equivalente à senha digitada.

Figura 11 – Diagrama de blocos da sequência lógica do login



Fonte: Do próprio autor, 2019

Primeiramente foi feito a comunicação com o domínio criado e então utilizando a função *cursor.execute* foi passado para esta função a linha de comando:

"SELECT username and senha FROM LoginUsuarios WHERE username = {} and senha = {};".format(user, psw)"

O comando seleciona o *username* e a senha da tabela criada anteriormente "LoginUsuarios" onde o "*username*" for igual ao nome da variável *user* que foi passada

através da URL do aplicativo para o servidor, que pela função “. format” substituiu as primeiras chaves pelo valor da variável “*user*”. O mesmo processo é feito com a senha, onde é substituída nas segundas chaves o valor da variável “psw” que foi passada através da URL, após isso é necessário escrever a linha de comando “db.commit()” onde “db” é variável que faz a conexão com o servidor, para que seja realizado a linha de comando descrita anteriormente.

Ao término, então, o servidor irá buscar na tabela criada LoginUsuarios o *username* e a senha que forem iguais às passadas para ele, então, é realizado um laço for com os resultados, onde são salvos em uma varável que guarda uma lista chamada de saída.

Quando for achado um resultado, ele será guardado nesta lista, após esta etapa estar concluída, é verificado se a lista está vazia, caso esteja então significa que não há no banco de dados um usuário e senha equivalentes, logo, o usuário escreveu a senha ou o *username* incorretos ou o usuário ainda não foi cadastrado e esta resposta é retornada para o aplicativo no formato de “jsonObject”, que precisa de uma chave e o valor que será passado, dessa forma, para receber um valor json é apenas necessário que seja buscado pela chave do valor que procura.

Caso a lista não esteja vazia então o usuário foi encontrado e senha é a mesma, caso isso ocorra, é retornado para o aplicativo, em formato json também que o usuário tem permissão de entrar no aplicativo.

Finalizado a lógica do banco de dados, deu-se início à implementação do aplicativo do motorista já com o *login* na rede, para isso, foi criado um projeto novo, onde no arquivo “.xml” foi criado uma interface simples de login do usuário contendo apenas um *EditText* para o campo *username* onde o usuário digitaria o seu *username* e outro *EditText* para a senha, neste campo o usuário deveria digitar a senha dele e um botão para fazer o *login*.

Todos estes componentes foram adicionados dentro de um *cardview* posicionado na parte superior da tela de *login*, foram alteradas as propriedades do *cardview* como a função “*app:cardCornerRadius = “20dp*” que cria um raio nos vértices do *cardiew* tornando ele com as bordas arredondadas, também definiu-se seu alpha para 0.9 dessa forma sua transparência é alterada para 90% de visibilidade. Finalizado a interface vista pelo usuário, iniciou-se então o desenvolvimento do

algoritmo para envio e recebimento dos dados fazendo a comunicação com o banco de dados criado.

Para a comunicação com o servidor, devido ao fato de o aplicativo estar sendo desenvolvido com a linguagem de programação kotlin, foram usados recursos próprios desta linguagem para realizar a conexão com o banco de dados. Primeiramente foi necessário adicionar dentro do arquivo "*manifests*" do aplicativo a permissão para a aplicação acessar a internet. Feito isso, outra linha de comando necessária para que haja uma comunicação com serviços web foi "*android:usersCleartextTraffic="true*", colocando esta variável para "*true*" foi possível enviar e receber dados do aplicativo com o servidor criado no PythonAnywhere.

Após realizar estes passos, foi aplicado um *listener* para o botão do login na Activity do login do usuário, onde quando ele fosse apertado fosse realizado um bloco de códigos definido posteriormente.

Para adicionar este *listener* apenas foi, através do id do botão, chamado a função *setOnClickListener*, e seguindo o que estava planejado, ao ser acionado, o aplicativo deveria enviar os dados de *username* e senha para o servidor e aguardar a resposta, caso a resposta seja positiva então o aplicativo deveria permitir que o usuário tivesse acesso ao conteúdo do aplicativo, caso não fosse, deveria mostrar na tela, para o usuário um texto informando a ele que a senha ou *username* estavam incorretos.

Para fazer o *login*, o aplicativo deveria acessar o mesmo caminho definido no servidor que contém a lógica para o *login* feita anteriormente, para isso, utilizou-se a função do kotlin "URL" onde é preciso enviar o caminho que deseja e posteriormente adicionar "*readText()"* para que seja realizado o acesso à URL descrita. No caso, devido ao fato de a URL do servidor receber dois parâmetros para fazer a análise do *login* e da senha. Estes parâmetros, no aplicativo foram recuperados através do *EditText* de cada um, ou seja, após o usuário digitar o *username* e a senha estes textos eram salvos em variáveis que seriam enviadas na URL para o servidor, como mostra a Figura 12.

Figura 12 – Variáveis que serão enviadas na URL

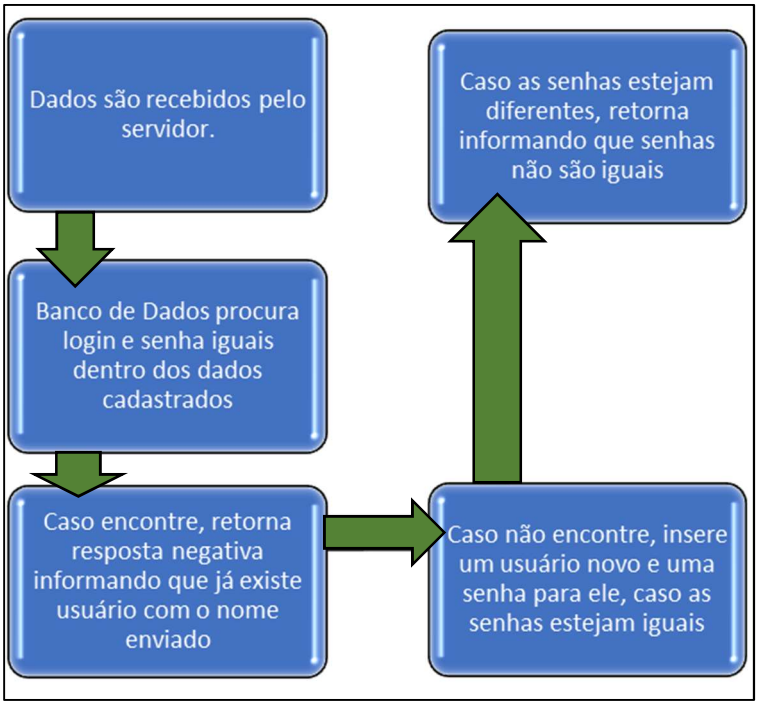
Fonte: Do próprio autor, 2019

A URL lida recebia a resposta do banco de dados por isso, foi salva em uma outra variável que posteriormente foi convertida em um objeto do tipo Json através da função “JSONObject”, ao termino deste processo, este objeto do tipo json foi convertido para uma lista onde pela função *“.get*” foi digitado a chave da resposta do banco de dados e então foi salvo em uma outra variável.

Depois, foi feito a análise desta resposta, caso a resposta fosse igual a “Entrou” então o usuário deveria entrar no aplicativo, para realizar esta troca de Activitys foi utilizado a função *Intent* que foi iniciada pela função “*startActiviy*” onde foi passado a função *intent* contendo a Activity que ela está e como segundo parâmetro a Activity que ela deveria ir, e após esta linha de comando foi chamado a função *finish()* para que a *activity* de login do usuário seja apagada para que não ocorra de o usuário estar no aplicativo, por algum motivo voltar para a tela de *login* e precisar fazer o *login* novamente.

A Figura 13 mostra a o diagrama de blocos da sequência lógica do cadastro do usuário do aplicativo. Caso a resposta fosse igual a “Usuario ou senha incorretos” então apenas era mostrado na tela de *login* esta resposta através da função “*text” (“setText()”* em java) para que o texto fosse substituído na tela um *TextView* contido entre o botão de *login* e o campo de senha do usuário, com isso o processo de *login* estava completo, deu-se início então ao cadastro do usuário.

Figura 13 - Diagrama de blocos da sequência lógica do cadastro



Fonte: Do próprio autor, 2019

O cadastro seguiu uma lógica similar à do *login* variando apenas alguns detalhes, para o cadastro, primeiramente, criou-se um botão com o texto de "CADASTRAR" fora do *cardview* na parte inferior da tela de *login*, pois caso o usuário não estivesse cadastrado ele teria a opção de se cadastrar e foi posicionado na parte inferior pois, geralmente, este botão será menos usado do que a do *login*, deixando o botão que será acionado mais vezes em um local mais acessível.

Finalizado este botão, dentro da *activity* do *login* foi necessário criar uma ação para este botão. No momento em que o botão fosse acionado então o aplicativo deveria direcionar o usuário para a página de cadastro, por isso adicionou-se para o botão criado um *listener* que ao ser acionado utilize a função *Intent* para mudar para a *Activity* de cadastro. Finalizado esta etapa, criou-se então dentro do projeto uma outra *Activity* chamada de CadastroActivity clicando com o botão direito sobre a pasta onde se encontrava as *Activitys* e selecionado a opção "new Kotlin file" e também foi criado um novo "*Layout resource file*" clicando sobre a pasta layout dentro da pasta "res", selecionado a opção "*new*" e "*layout resource file*" criando então o arquivo ".xml" como o nome definido de "activity_cadastro".

Dentro do arquivo ".xml" criado, foi desenvolvida uma interface semelhante à do login porém com um campo de senha a mais para que o usuário crie uma senha e

então confirme ela para evitar futuras falhas ao fazer o *login* e também foi adicionado um outro botão com o texto “CADASTRAR” para efetuar o cadastro do usuário, finalizado a interface foi necessário criar a ação para o botão, que, similar ao *login*, deveria, ao ser acionado, enviar os dados do *username* e os dois parâmetros de senha, a primeira digitada e a confirmação desta mesma senha, o servidor ficou encarregado de conferir os dados e retornar a resposta para o aplicativo.

Para essa ação, foi adicionado um outro *listener* para o este botão e ao ser acionado, utilizou assim como o *login*, a função URL do kotlin enviando como parâmetro o caminho com os parâmetros de *username* senha e a confirmação da senha, este caminho foi criado posteriormente no servidor.

Após a leitura da URL foi feito a conversão para um objeto do tipo json e então convertido novamente para uma variável do tipo “lista” e após isso foi feito a análise da resposta do usuário recuperando o resultado utilizando uma chave para receber o objeto json contendo a resposta do banco de dados.

Caso a resposta fosse “Cadastro Realizado com Sucesso” então o aplicativo deveria dar ao usuário o acesso ao aplicativo e encaminhá-lo até a *Activity* principal.

Para fazer a mudança de telas foi utilizado a função *Intent* que passado os parâmetros da *Activity* onde se encontrava e a que deveria ir foi utilizado a função *startAcitivty* passando como parâmetro a intenção criada anteriormente (*intent*) para que fosse possível a troca de telas, após isso, a função “*finish()”* foi utilizada para que o usuário não volte para a tela de cadastro novamente sem uma intenção real.

Se a resposta do banco de dados for igual a “Senhas diferentes” então esta resposta deveria ser mostrada na tela para o usuário através da função “*text*” sendo chamada a partir do id do *TextView* localizado entre o campo da confirmação da senha e o botão de cadastrar, enviando a resposta do banco de dados para ser mudada na tela.

Finalizado a interface do aplicativo, deu-se início então à implementação da lógica do servidor para fazer o processamento das informações e retornar a resposta correta para o aplicativo.

Dentro do arquivo python criado no PythonAnywhere que contém a implementação do login do usuário, abaixo desta implementação, foi criada outra rota do servidor, ou seja, outro caminho possível que poderia ser acessado tanto do aplicativo quanto em um servidor *Web*.

A rota criada recebia na extensão da URL três parâmetros, sendo eles: nome do usuário (chamada de *username*), senha e a confirmação da senha (chamada de senha2). Por este motivo na função chamada através deste caminho, foi necessário passar como argumentos os três parâmetros.

O primeiro passo foi estabelecer a conexão com o banco de dados criado, depois, foi criado uma condição onde foi verificado se a senha, passado como o primeiro parâmetro era igual à segunda senha, chamada de senha2, caso não forem iguais então o servidor retorna para o aplicativo em um formato de json que as senhas são diferentes, caso contrário é necessário que o servidor crie um novo usuário.

Para criar um novo usuário foi preciso enviar a linha de comando “INSERT INTO LoginUsuarios(*username*, senha) VALUES ({},{});".format(user, psw))“, esta linha de código é responsável por criar dentro da tabela LoginUsuarios um novo *username* e uma nova senha, sendo que estes valores são passados como “*user*” e “psw” que através da função *.format* substitui as chaves pelos conteúdos destas variáveis que são equivalentes ao nome do usuário e da senha digitados pela pessoa que teve a intenção de criar um cadastro.

Depois disso é retornado ao aplicativo em um formato json que o cadastro foi realizado com sucesso. Este mesmo processo foi feito tanto para o aplicativo dos motoristas quanto para o aplicativo dos usuários e o das estações de trem/metrô.

Finalizado este processo de *login* e cadastro, o próximo passo foi iniciar um esboço de como seria feito a lógica de troca de dados e em quais momentos seriam feitos os envios de dados etc.

O primeiro ponto que foi pensado foi em que momento seria feito o envio da posição atual do usuário e do motorista, após algumas ideias definiu-se que seria necessário enviar ao banco de dados a latitude e a longitude de cada um, e estas informações seriam enviadas no momento em que esses pontos se alterassem pois, na implementação onde o aplicativo recupera a localização atual do usuário há uma função em que no momento em que o ponto em que o usuário se encontra passa de um determinado raio definido na implementação, então é possível realizar alguma ação neste instante e no caso, a proposta foi que a ação fosse o envio dos pontos de latitude e longitude.

Depois foi estruturado como seria tratado as informações provenientes do aplicativo do motorista, do usuário e do aplicativo de empresas responsáveis por trens e metros.

O primeiro a ser modelado foram as informações do motorista pois, além do fato de ser necessário saber quais são os pontos de latitude e longitude atuais, também foi necessário saber o ponto de partida do ônibus, o ponto final, e qual rota ele estava fazendo.

Para solucionar estes problemas, foi criado uma nova tabela no banco de dados para fazer o armazenamento das informações, nesta tabela, há doze colunas, a primeira coluna é o "*id"* que incrementa automaticamente na medida em que é adicionado um usuário novo, em seguida vem a descrição, este campo foi adicionado para diferenciar os usuários, se o usuário é motorista, se é uma estação de trem/metro ou se é um usuário comum.

A terceira coluna é o *username* de cada um, que posteriormente será usada como chave para envio e recebimento de respostas dos objetos em formato "json".

As próximas duas colunas são a latitude e a longitude atuais do usuário, seja ele motorista usuário ou estação, no caso da estação esta latitude e longitude serão sempre fixas, já no caso de usuário e motorista as latitudes e longitudes atuais serão atualizados nestas colunas.

Depois, as demais colunas foram pensadas para solucionar o problema de como identificar se o motorista se encontra em uma determinada rota. Para isso, foi feito uma análise e notado que o mesmo ônibus se está em uma rota terá um determinado nome, porém caso chegue até seu ponto final e esteja retornando então seu nome será alterado para outro, e vice-versa caso chegue no seu ponto de início novamente.

Levando em conta este ponto, foi modelado a proposta de gerenciamento das informações da seguinte forma: o motorista ao iniciar a sua rota deverá digitar no aplicativo qual é o seu ponto final, qual será o nome do ônibus na ida e qual será o nome do ônibus na volta.

O aplicativo, a partir do ponto final deve converter o texto digitado neste campo em latitude e longitude e enviar para o banco de dados, depois enviar para os nomes também. Ao mesmo tempo, será preciso que o aplicativo recupere a latitude e

longitude no instante em que ele iniciar a rota, pois este ponto será o ponto de partida da rota, e envie também para o servidor.

A tabela criada ficará, depois da coluna de longitude, uma coluna para a latitude de saída, outra para longitude de saída que será o ponto de partida depois outra coluna com latitude do destino e uma coluna com a longitude de destino, que representarão o ponto final do motorista.

Após estas colunas a tabela terá as colunas de nome de ida, nome de volta e por último nome atual do ônibus. No caso, quando o motorista iniciar a rota o nome atual será igual o nome de ida, quando o motorista chegar no ponto final então o nome atual deverá ser alterado e ficar igual o nome de volta e no instante que o motorista chegar no ponto de partida da rota o nome atual deve retornar com o nome de ida, dessa forma, é possível saber a rota que o motorista está fazendo e para onde está indo.

No caso do usuário e da estação, os pontos de partida e do destino serão iguais a zero e os nomes referentes ao ônibus serão substituídos por um hífen. (Figura 14)

Figura 14 – Dados contidos na Tabela “Geolocalização06” do Banco de Dados

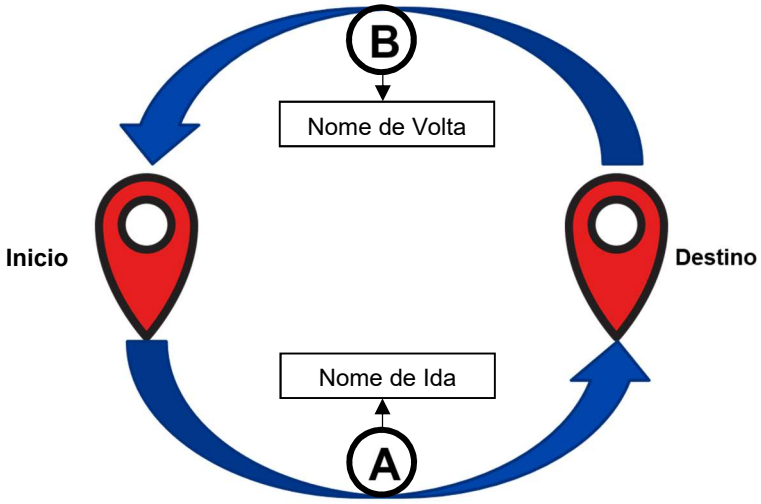
```
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Geolocalizacao06;
+----+-----------+----------+----------+-----------+-----------+-----------+-----------+-----------+--------+----------+-----------+
| id | descricao | username | latitude | longitude | latiSaida | longiSaida | latiDestino | longiDestino | nomeIda | nomeVolta | nomeAtual |
+----+-----------+----------+----------+-----------+-----------+-----------+-----------+-----------+--------+----------+-----------+
|  1 | motorista | primeiro |    -23.1 |      -46.1 |      -23.1 |      -46.1 |      -23.2 |      -46.2 | A      | B        | A         |
+----+-----------+----------+----------+-----------+-----------+-----------+-----------+-----------+--------+----------+-----------+
1 row in set (0.00 sec)

mysql> INSERT INTO Geolocalizacao06(descricao,username,latitude,longitude,latiSaida,longiSaida,latiDestino,longiDestino,nomeIda,nomeVolta,nomeAtual)
 VALUES('motorista','segundo',-23.1500,-46.1500,0,0,0,0,'-','-','-');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Geolocalizacao06;
+----+-----------+----------+----------+-----------+-----------+-----------+-----------+-----------+--------+----------+-----------+
| id | descricao | username | latitude | longitude | latiSaida | longiSaida | latiDestino | longiDestino | nomeIda | nomeVolta | nomeAtual |
+----+-----------+----------+----------+-----------+-----------+-----------+-----------+-----------+--------+----------+-----------+
|  1 | motorista | primeiro |    -23.1 |      -46.1 |      -23.1 |      -46.1 |      -23.2 |      -46.2 | A      | B        | A         |
|  2 | motorista | segundo  |   -23.15 |    -46.15 |         0 |         0 |         0 |         0 | -      | -        | -         |
+----+-----------+----------+----------+-----------+-----------+-----------+-----------+-----------+--------+----------+-----------+
2 rows in set (0.00 sec)

mysql> UPDATE Geolocalizacao06 SET descricao = 'usuario' WHERE id = 2;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM Geolocalizacao06;
+----+-----------+----------+----------+-----------+-----------+-----------+-----------+-----------+--------+----------+-----------+
| id | descricao | username | latitude | longitude | latiSaida | longiSaida | latiDestino | longiDestino | nomeIda | nomeVolta | nomeAtual |
+----+-----------+----------+----------+-----------+-----------+-----------+-----------+-----------+--------+----------+-----------+
|  1 | motorista | primeiro |    -23.1 |      -46.1 |      -23.1 |      -46.1 |      -23.2 |      -46.2 | A      | B        | A         |
|  2 | usuario   | segundo  |   -23.15 |    -46.15 |         0 |         0 |         0 |         0 | -      | -        | -         |
+----+-----------+----------+----------+-----------+-----------+-----------+-----------+-----------+--------+----------+-----------+
2 rows in set (0.00 sec)

mysql>
```

Fonte: Do próprio autor, 2019

Após modelado a tabela de armazenamento das informações iniciou-se o processo de implementação do gerenciamento das informações no servidor. Foi criado uma rota nova para o envio das informações do motorista onde, na *URL*, continha todos os parâmetros como o *username* que é o mesmo que foi utilizado para entrar no aplicativo, a “LatLong” atual, de saída e de destino, nome de ida, volta e nome atual.

A lógica de gerenciamento dos ônibus foi simplificada e é feita de forma automática, primeiro o aplicativo deveria salvar a latitude e longitude de início, depois, através da informação de ponto final que deveria ser convertida em latitude e longitude, é gravada seu ponto de destino e então cria-se um raio em torno desses pontos. Dependendo do sentido do ônibus ele possuí um nome diferente, por exemplo, do ponto de inicio para o ponto de destino um ônibus tem o nome de “A”, porém, do ponto de destino para o ponto de início ele possuí o nome de “B”. A Figura 15 representa este fluxo.

Figura 15 - Fluxo de atualização do nome do ônibus



Fonte: Do próprio autor, 2019

Caso o servidor verifique que o motorista está próximo ao seu ponto de destino, seu nome atual, no servidor, é atualizado para o nome de volta, se for analisado que ele está próximo ao ponto de início então o nome atual é atualizado para nome de ida, se sua localização estiver fora do raio destes dois pontos então seu nome atual é mantido o mesmo.

O primeiro passo foi fazer a conexão com o banco de dados, depois analisou-se se motorista já se encontra no ponto final para isso é selecionado a latitude do motorista e salvado em uma variável, após isso, o servidor seleciona a sua latitude de destino e então faz uma subtração e guarda em uma variável multiplicando ela por 10000 (dez mil).

O mesmo processo é feito para a longitude e através de uma formula matemática foi possível descobrir a distância entre o ponto atual e o final, caso esta

distância estivesse dentro de um raio de cem metros então é feito uma atualização do nome atual pelo nome de volta, se não, é realizado o mesmo processo inicial porem ao invés de o servidor selecionar a “LatiLng” de destino é selecionado a latitude e longitude de saída, o mesmo processo matemático é feito e se caso a distância do ponto atual estiver dentro de um raio de cem metros então o nome atual é atualizado pelo nome de ida, estas atualizações são feitas através da linha de comando utilizando o “*UPDATE*”.

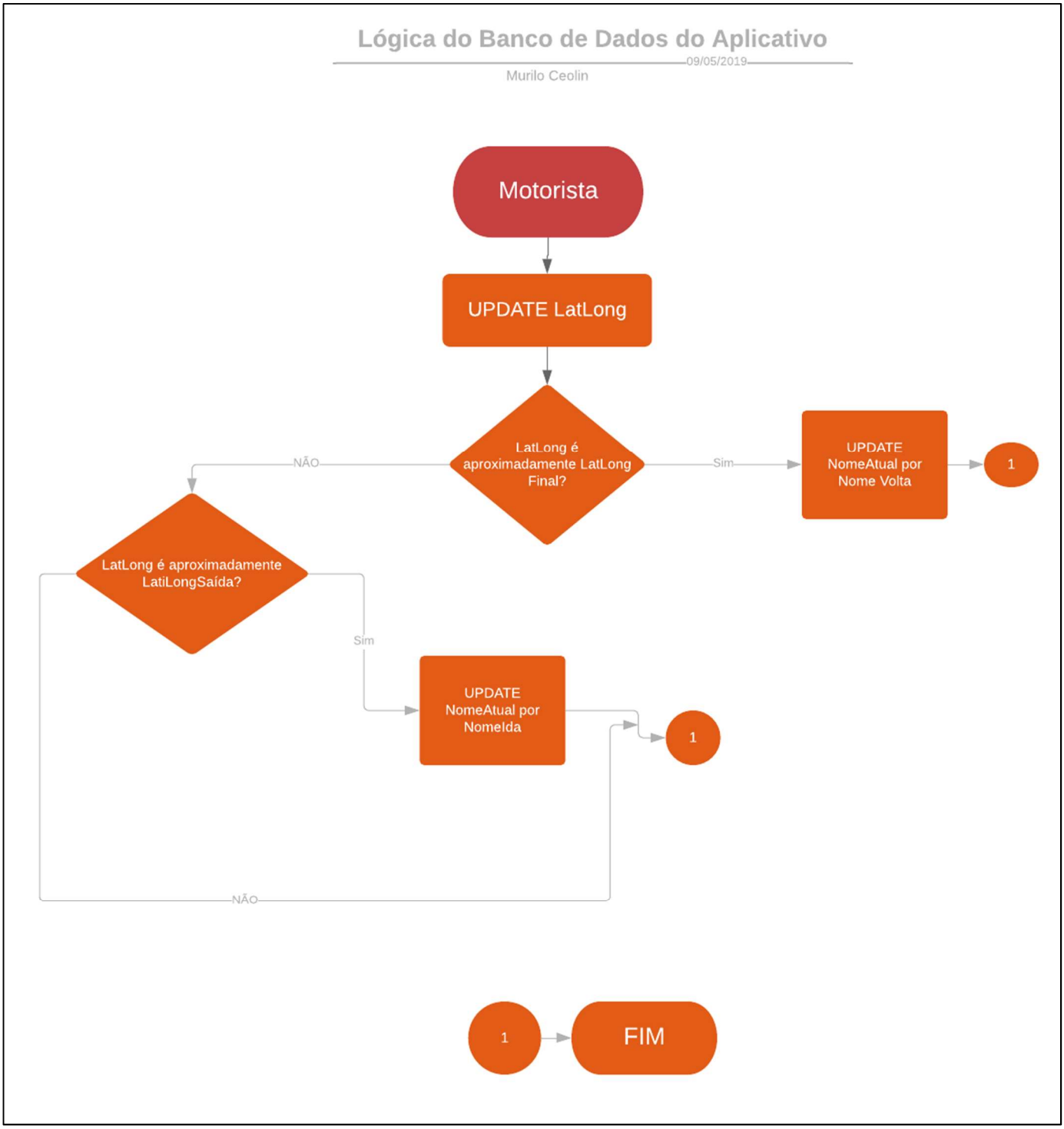
Depois, deu-se início ao processo de gerenciamento das informações vindas do aplicativo do usuário. Criou-se uma rota para os usuários que apenas eram passados como parâmetros o *username*, que é o mesmo utilizado para entrar no aplicativo, a latitude e longitude atuais e o ônibus que o usuário deve embarcar.

O primeiro passo realizado na função realizada para esta rota foi atualizar, toda vez que é recebido parâmetros novos, a latitude e longitude do usuário, depois, é feito o mesmo processo descrito na função da *URL* do motorista para selecionar a latitude atual do usuário porém como segundo valor de latitude será selecionado a latitude de todos os usuários em que a descrição for igual a “estação” e o mesmo processo é realizado com a longitude, depois é calculado novamente a distância entre os dois, caso a distância seja menor que trezentos metros, então é necessário notificar a estação perto que um usuário com deficiência visual está entrando nesta estação.

Caso a distância não esteja dentro deste raio, então é feito outra análise de latitude e longitude comparando a LatiLong atual do usuário, porém foi comparado com todos os usuários que tiverem descrição igual a “ponto” caso a distância for menor que cinquenta metros então significa que o usuário está próximo ao ponto de ônibus, logo, isso indica que se caso tiver algum ônibus que o leve até o seu destino então este motorista deve ser notificado.

Por isso, caso a distância for maior que este raio o usuário está no trajeto até o ponto, por isso o motorista não deve ser notificado. O diagrama de blocos da Figura 16 ilustra o procedimento.

Figura 16 – Fluxograma do tratamento dos dados do Motorista



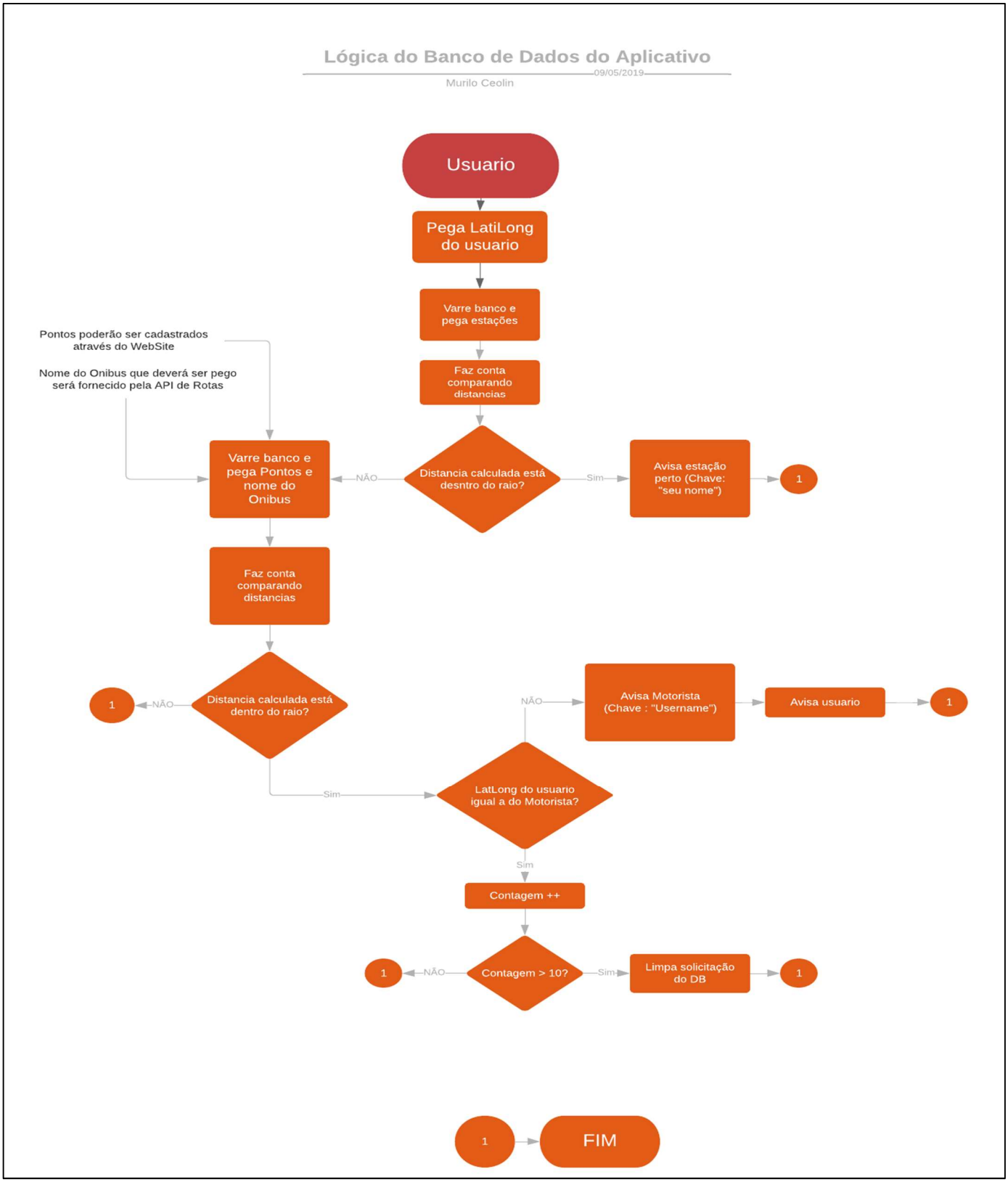
Fonte: Do próprio autor, 2019

No caso de o usuário estar próximo ao ponto então é feito novamente a mesma análise de distância entre latitude e longitude atuais do usuário, entretanto será comparado com usuários em que a descrição for igual a “motorista” e o nome atual for igual ao nome do ônibus que foi passado através da URL.

O nome do ônibus que o passageiro deve embarcar é proveniente dos dados provenientes da *API* do Google que traça rotas através do transporte público. Se a distância entre o usuário e este ônibus selecionado naquele instante for menor que dois mil metros então significa que o usuário e o ônibus que deve ser pego estão próximos então é retornado que o ônibus está chegando, caso contrário, é retornado que não há nenhum ônibus perto.

Se por acaso a distância dor menor que quinze metros, uma variável estática iniciada em zero é incrementada, se em dez atualizações de localização a distância entre os dois estiver dentro desta faixa então a solicitação do usuário é retirada do banco de dados atualizando sua “LatLong” para zero em ambas. A Figura 17 mostra o fluxograma da lógica do tratamento dos dados do Usuário no banco de dados

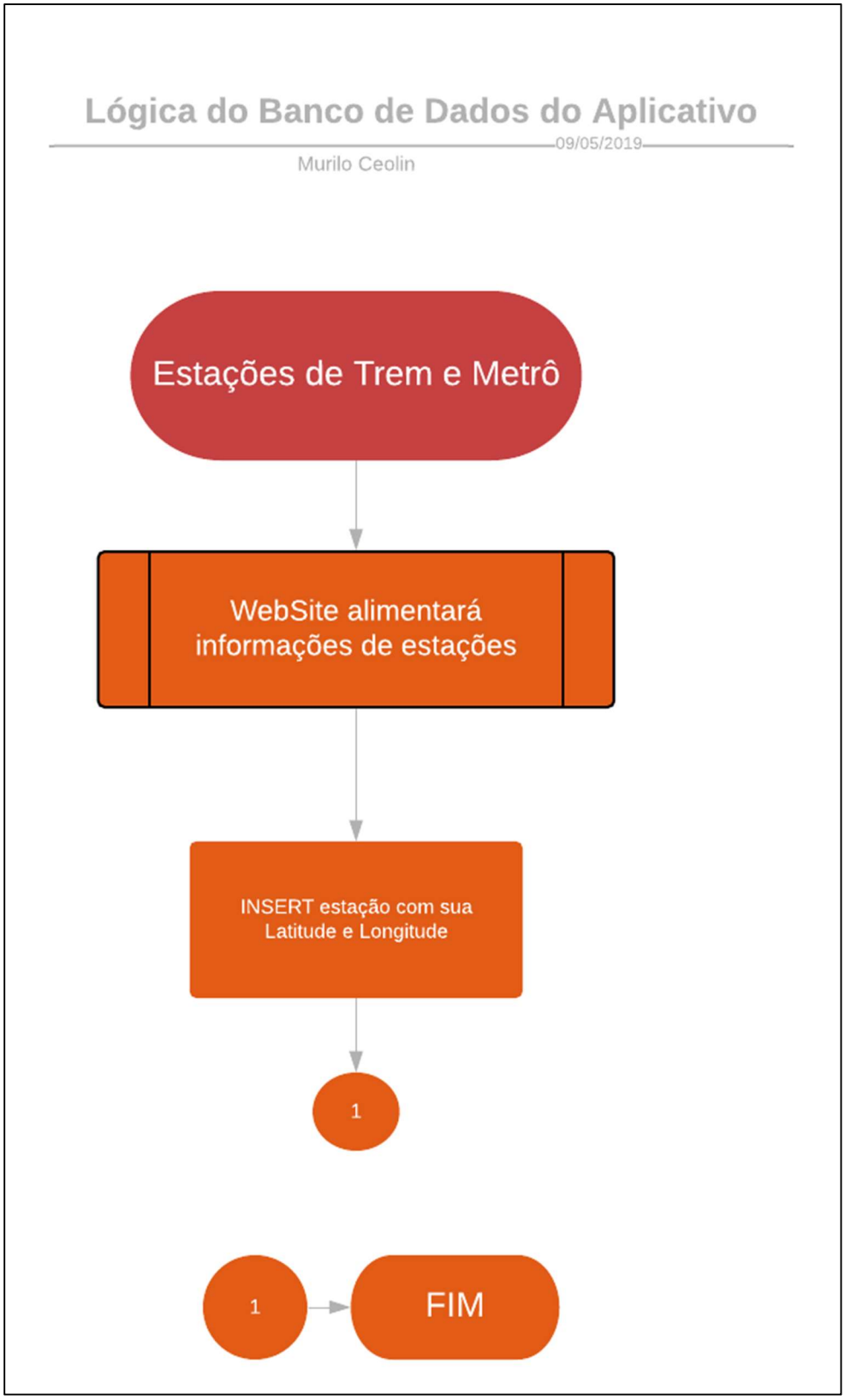
Figura 17 – Fluxograma do tratamento dos dados do Usuário



Fonte: Do próprio autor, 2019

No caso das estações e pontos de ônibus, foi criada uma rota que apenas insere na tabela do banco de dados um novo ponto ou uma nova estação caso não exista, como mostra o fluxograma da Figura 18, pois a proposta planejada por orientador e orientado é que seja um sistema que seja útil e utilizado por todos afim de ajudar as pessoas principalmente com deficiência visual a se locomoverem nas cidades com maior autonomia, dessa forma, as estações e pontos de ônibus poderão ser adicionadas ao sistema através de um Website que será implementado posteriormente e ficará aberto e os usuários poderão alimentar o banco de dados com as informações, dessa forma, qualquer pessoa do brasil poderá instalar o aplicativo e adicionar os pontos e estações da sua cidade podendo assim usufruir do aplicativo e do sistema em qualquer região contanto que os motoristas dos ônibus e as estações utilizem também o aplicativo de cada um.

Figura 18 – Fluxograma do tratamento dos dados das estações de trem e metrô



Fonte: Do próprio autor, 2019

Devido ao fato da recuperação atual do usuário estar com problemas, para os testes dos aplicativos foi simulado as latitudes e longitudes de cada um através de *EditTexts* e a função de enviar os dados no momento em que a localização variasse foi simulada por um botão simples, e as respostas eram mostradas em um *TextView* simples localizada acima do primeiro *EditText.*

Por fim a última implementação feita foi a notificação para o usuário em que no instante que a resposta do banco indicasse que usuário e motorista do ônibus que o usuário precisa estão próximos então fosse enviado uma notificação informando o status. A implementação da notificação utilizou recursos próprios do *android* e utilizou-se textos fixos para cada situação e o ícone do aplicativo.

4. RESULTADOS OBTIDOS

Aqui serão abordados os resultados que foram alcançados em cada etapa do projeto, apresentando o comportamento do aplicativo nas diversas possibilidades de situações que podem ocorrer enquanto usado pelos usuários, dentre elas serão apresentados as telas e o comportamento do aplicativo do usuário e do motorista de ônibus.

4.1 Aplicativo do Motorista de Ônibus:

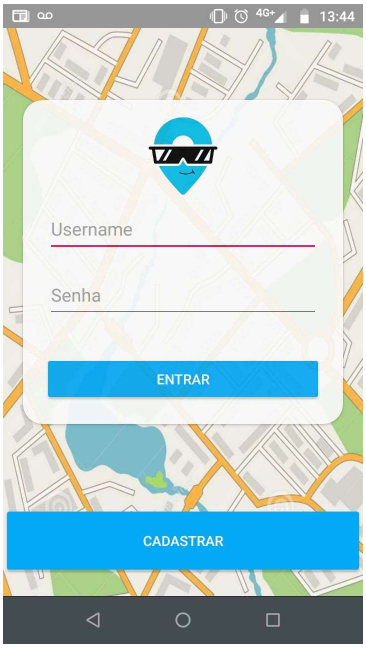
Neste capítulo será apresentado o comportamento do aplicativo do motorista de ônibus em algumas possibilidades como por exemplo, um *login* com senha ou nome de usuário incorreto, o *login* de um usuário já cadastrado e etc.

4.1.1 *Login* do motorista cadastrado

A Figura 19 é a tela inicial do aplicativo, para fazer a realização do *login* o usuário primeiro deve digitar o seu "*username*", ao clicar sobre a caixa de texto referente ao campo do nome do usuário então o teclado irá aparecer para que possa ser digitado o nome desejável.

Após escrito seu *username* o usuário deverá digitar sua senha clicando sobre o campo de texto referente à senha, a senha para entrar no aplicativo deverá ser numérica. Depois de digitado a senha, caso o usuário tenha digitado a senha e o usuário corretamente e estiver cadastrado então ele terá acesso ao aplicativo indo então para a tela principal representada pela Figura 24.

Figura 19 - Tela inicial do aplicativo do motorista



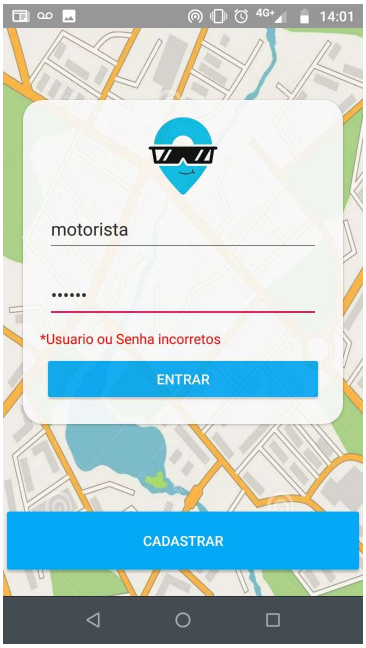
Fonte: Do próprio autor, 2019

4.1.2 *Login* do motorista com usuário ou senha incorretos ou não cadastrado

Há a possibilidade de o usuário digitar a senha errado ou até mesmo o seu nome de usuário, caso isso ocorra é necessário que o aplicativo não deixe que o usuário entre no aplicativo por motivos de segurança e é preciso também, informá-lo do que está ocorrendo.

No caso, após fazer o processo de *login*, no momento em que, após digitado o nome de usuário ou senha errados então, apertar o botão escrito “ENTRAR” então, no aplicativo irá aparecer a mensagem informando ao usuário que houve um problema conforme mostra a Figura 20, no momento em que essa mensagem aparecer a aplicativo não irá direciona-lo para a tela principal até que seja digitado um usuário ou senha validos e for apertado o botão de entrar novamente.

Figura 20 - Mensagem de erro em caso de dados incorretos



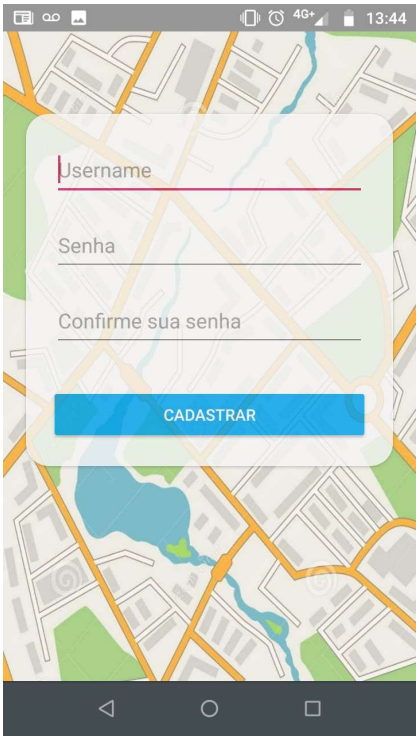
Fonte: Do próprio autor, 2019

4.1.3 Cadastro do motorista

Caso seja a primeira vez que o usuário entra no aplicativo é necessário que ele realize um cadastro, neste cadastro apenas é preciso o *username*, ou seja, o nome de usuário que ele queira escolher, e a senha numérica que deseja, depois é preciso confirmar a senha, caso as senhas estejam iguais e não haja outro usuário com o nome escolhido no momento do cadastro então após apertar o botão “CADASTRAR” o aplicativo deverá fazer o cadastro e redireciona-lo para a tela principal dando-lhe permissão para entrar no aplicativo.

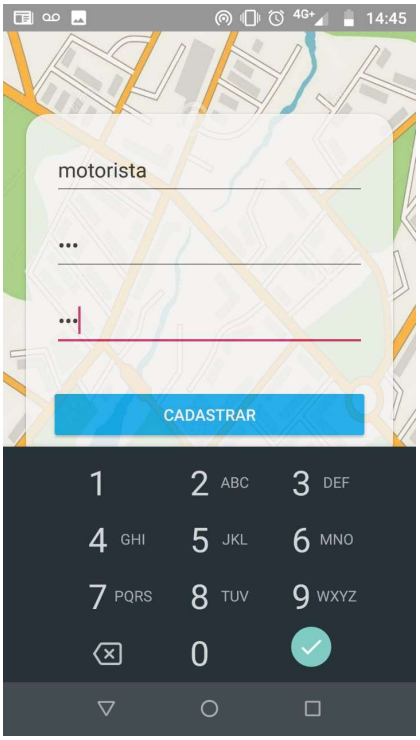
O fluxo para que isso aconteça, está representado pela Figura 27, onde a Figura 19 é a tela inicial, para realizar o cadastro é preciso que o usuário aperte o botão inferior da página principal com o texto de “CADASTRAR”, ao fazer isso, o usuário será redirecionado para outra tela como mostra a Figura 21, em que deverá digitar o *username* que deseja e uma senha, caso as senhas estejam iguais e não haver nenhum *username* igual ao que foi digitado no servidor então o aplicativo redirecionará o motorista para que na tela principal (Figura 24) o motorista digite qual é o ponto final da sua rota, qual o nome do ônibus no trajeto de ida e qual o nome do ônibus no trajeto de volta após isso, ele deverá apertar o botão “INICIAR ROTA”.

Figura 21 - Tela de cadastro



Fonte: Do próprio autor, 2019

Figura 22 – Tela sendo preenchida



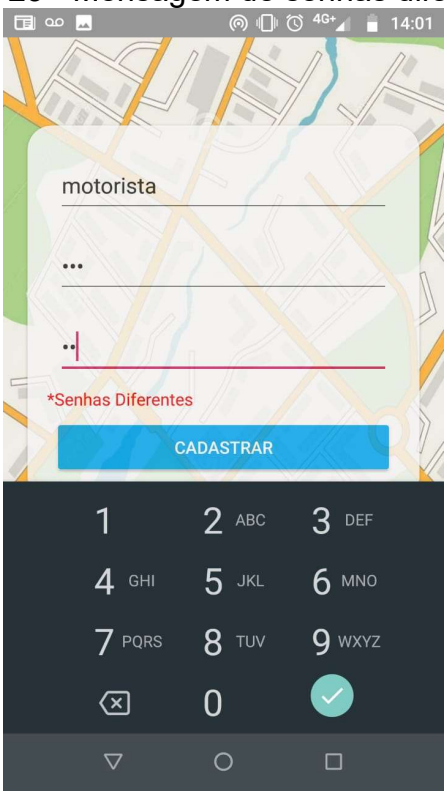
Fonte: Do próprio autor, 2019

4.1.4 Cadastro do motorista com senhas diferentes

Caso o motorista que estiver fazendo seu cadastro digitar as senhas diferentes o aplicativo deverá informar que as senhas estão diferentes e não permitir que o motorista faça o cadastro para que não haja erros no *login* posteriormente.

Portanto, após passar pela tela como mostra a Figuras 21 em seguida pela tela representada pela Figura 22 e digitar o usuário e as senhas diferentes então irá aparecer na tela a mensagem informando que as senhas estão diferentes como mostra a Figura 23. Após isso, o aplicativo não poderá efetivar o cadastro enquanto as senhas estiverem iguais e não haver nenhum outro usuário igual no banco de dados criado.

Figura 23 - Mensagem de senhas diferentes



Fonte: Do próprio autor, 2019

4.2 Aplicativo do Usuário:

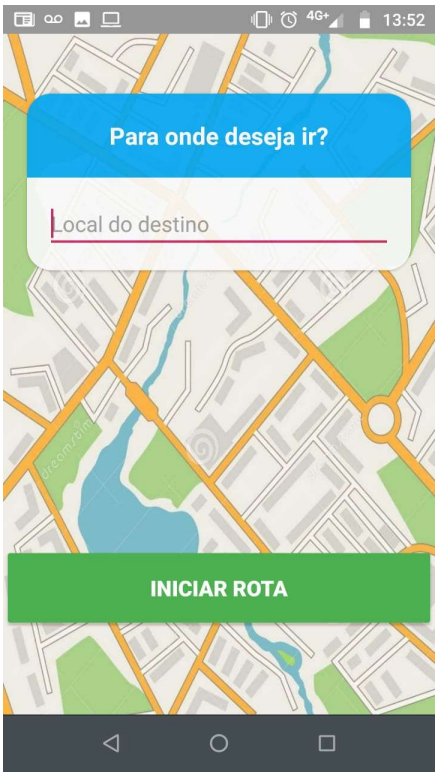
As telas de *login* e cadastro do usuário são similares às do aplicativo do motorista, pois foi optado por manter um padrão de aplicativo apenas mudando funções e informações que devem ser passadas por cada um.

4.2.1 *Login* do usuário cadastrado

O *login* do usuário comum é semelhante ao do motorista, deve digitar o *username* e a senha já cadastrados no aplicativo, no instante que forem digitados o usuário irá apertar o botão “ENTRAR” e então o aplicativo irá enviar os dados para o servidor para serem verificados se as informações estão certas, caso a resposta do servidor seja positiva então a aplicativo irá para a tela principal do usuário como mostra a Figura 24, que deverá ser digitado apenas o local de destino.

Todas as informações de qual ônibus ele deverá embarcar deveriam vir da API do Google que cria as rotas.

Figura 24 - Tela principal do aplicativo

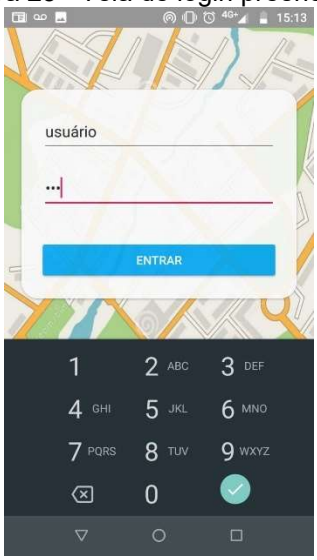


Fonte: Do próprio autor, 2019

4.2.2 *Login* do usuário com usuário ou senha incorretos ou não cadastrado

Caso o usuário digite a senha ou o nome de usuário errados quando for fazer o login, então, deverá ser mostrado ao usuário que o usuário ou a senha digitada estão incorretos, como mostra a Figura 26. As sequências de Figuras acima demonstram telas exemplificando um usuário que tentou fazer um *login* com informações erradas ou com um *username* não cadastrado, ao clicar no botão “ENTRAR” na tela aparece a mensagem “*Usuário ou Senha incorretos”.

Figura 25 - Tela de login preenchida



Fonte: Do próprio autor, 2019

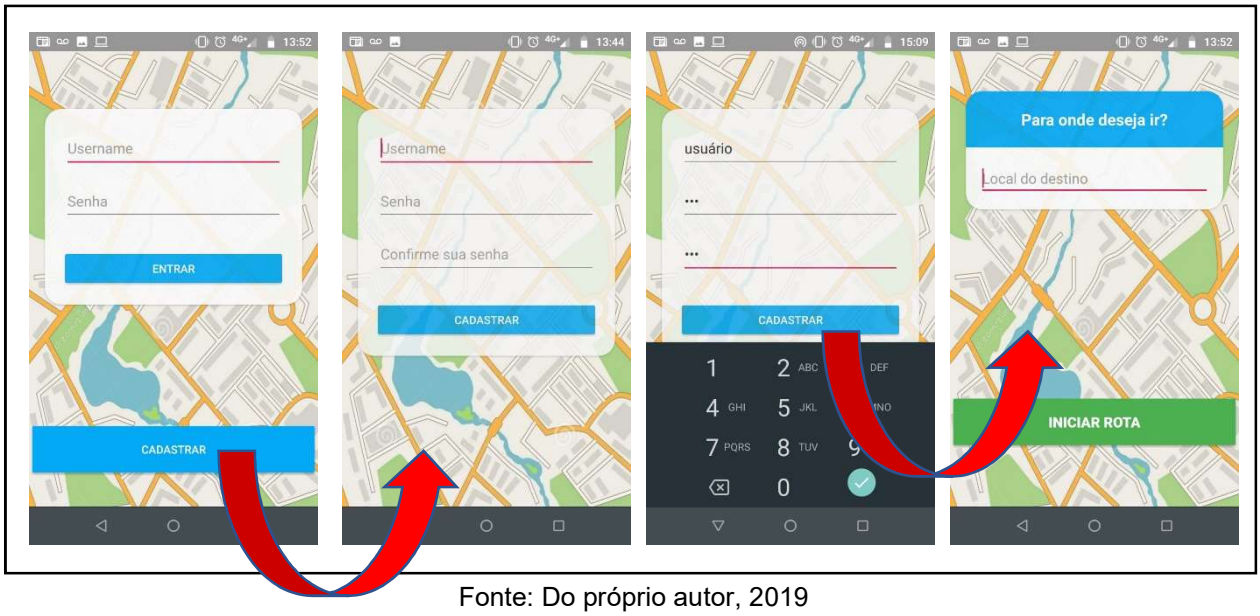
Figura 26 - Mensagem de erro na tela no *login*

Fonte: Do próprio autor, 2019

4.2.3 Cadastro do usuário

O processo de cadastro de um usuário é semelhante ao do motorista já apresentado no trabalho, onde ao apertar o botão “CADASTRAR” o usuário é redirecionado à tela de cadastro que contém três campos para serem digitados, sendo eles o nome de usuário, a senha e a confirmação da senha, neste momento caso não haja nenhum usuário cadastrado já com o nome que foi digitado, então o aplicativo irá para a tela principal representada pela Figura 27 em que deverá ser digitado o seu destino.

Figura 27 – Sequência de telas em um cadastro realizado



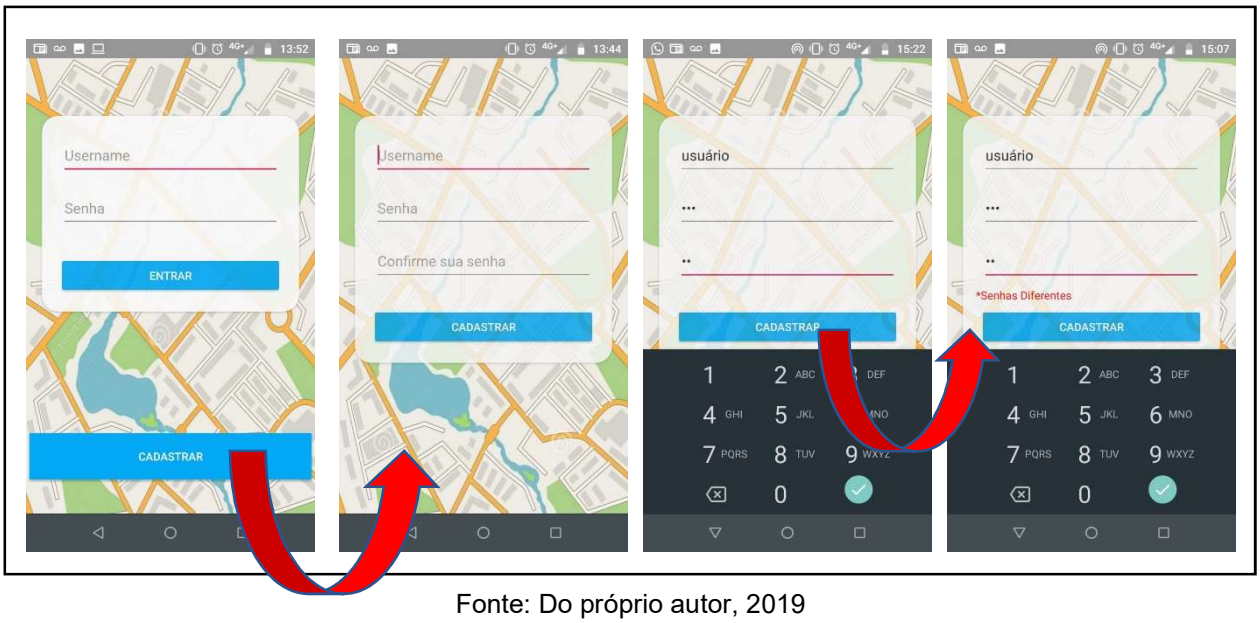
Fonte: Do próprio autor, 2019

4.2.4 Cadastro do usuário com senhas diferentes

No momento em que o usuário tentar se cadastrar no aplicativo com senha ou usuário inválido, seja pelo motivo das senhas estarem incorretas, ou seja pelo motivo de já haver cadastrado no aplicativo um “*username*” igual, então o aplicativo deverá informar o usuário, através de uma mensagem na tela, que no caso de um usuário deficiente visual poderá ser lida a mensagem através de um aplicativo próprio que transforma textos na tela de um aplicativo em fala, ou seja, o usuário poderá entender o que está escrito na tela e ter a possibilidade de mudar as informações para efetuar o cadastro.

A mensagem que informa o usuário está representada na Figura 28 e permanecerá na tela até que o usuário digite informações validas ou saia da tela de cadastro e entre novamente.

Figura 28 - Sequência de um cadastro com senhas diferentes



Fonte: Do próprio autor, 2019

4.3 Simulações

Devido ao fato de não ter sido possível a recuperação da posição atual do usuário (entenda usuário deste caso como qualquer pessoa que esteja utilizando o aplicativo, seja ela motorista ou não) e afim de facilitar os testes durante a implementação, foi utilizada este tela onde ao ser apertado o botão “LOCALIZACAO” então são enviados para o banco de dados a latitude e a longitude simuladas para o banco de dados criado para que seja feito as análises que seriam feitas com latitudes e longitudes reais.

4.3.1 Aplicativo do motorista de Ônibus

O botão simula a função que realiza o envio da latitude e longitude do motorista no instante em que elas mudariam, dessa forma, é possível alterar dinamicamente seu posicionamento testando todas as condições possíveis de situações reais sem ter a necessidade de se locomover com o aplicativo. A mensagem é apresentada na tela apenas para que o desenvolvedor tenha o feedback da resposta do banco de dados.

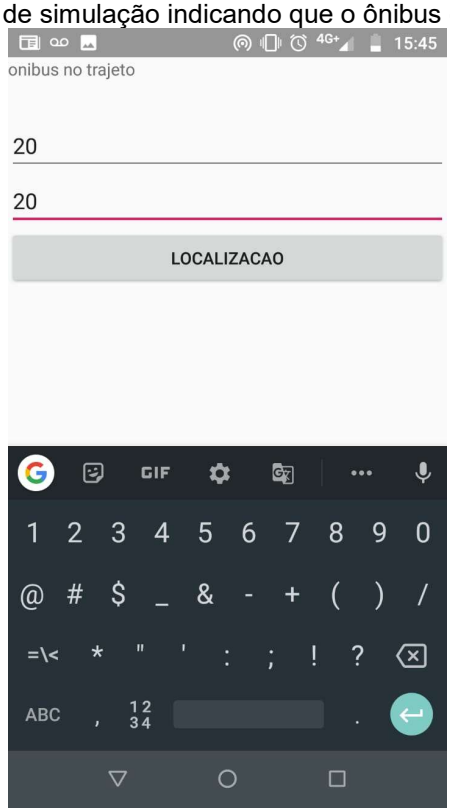
4.3.1.1 Ônibus no trajeto

A Figura 29 representa a tela utilizada para gerar as simulações de mudança de posicionamento, onde o primeiro campo onde é possível digitar, simula a latitude do motorista e o campo de texto abaixo representa a longitude.

No caso, as latitudes e longitudes do ponto de partida e do ponto final (Destino), o nome do ônibus de ida e o nome do ônibus de volta já foram enviados para o banco de dados e gravados através da tela principal como mostra a Figura 28. Através do campo “Ponto final da sua Rota” o texto digitado é convertido em latitude e longitude e enviados nas colunas de “Destino” (“latiDestino” e “longiDestino” são os nomes das colunas de latitude e longitude do destino do ônibus).

O servidor irá então, a cada vez que for apertado o botão simulando a mudança do ponto de latitude e longitude, analisar onde o ônibus se encontra e retornar a resposta na tela de simulação, como a LatiLgn enviada não estava próxima nem do ponto de partida nem do ponto final então apenas foi retornado que ele estava no trajeto como mostra a Figura 29.

Figura 29 - Tela de simulação indicando que o ônibus está em sua rota



Fonte: Do próprio autor, 2019

4.3.1.2 Ônibus chegando no ponto final da Rota

Nesta condição apresentada pela Figura 30 foi simulado a situação de o ônibus estar bem próximo do ponto final da sua rota, comparando com os dados enviados pelo motorista antes de iniciar a rota, no instante em que isso acontece, o servidor deve alterar o nome atual do ônibus pelo nome enviado de Volta, assim, será possível separar os ônibus pelas rotas que estão fazendo e buscar, quando necessário, apenas os ônibus uteis para cada situação.

Foi simulado a latitude e longitude iguais às do ponto final, nesse momento a reposta retornada pelo servidor foi que o nome atual foi atualizado pelo nome de volta como mostra a Figura 30.

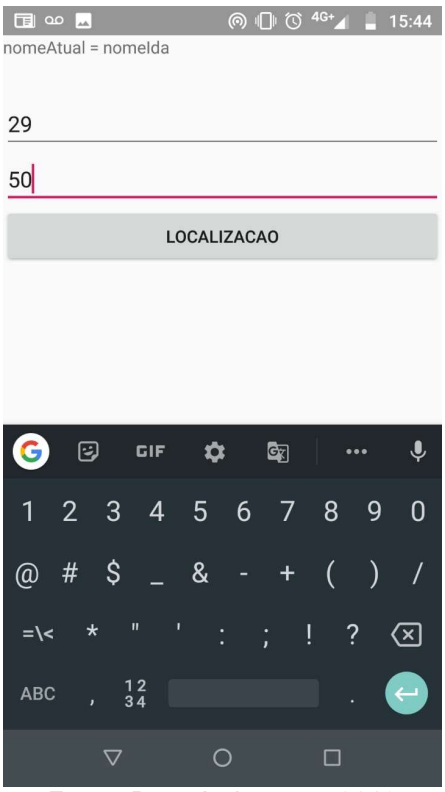
Figura 30 - Tela de simulação indicando que o ônibus chegou em seu ponto final

Fonte: Do próprio autor, 2019

4.3.1.3 Ônibus chegando no ponto de Partida

A Figura 31 representa a situação oposta à mostrada na Figura 30, porém, neste caso, o teste foi feito simulado a situação do ônibus ter completado todo o trajeto e chegado novamente ao seu ponto de partida, dessa forma, o ônibus deverá recomeçar a rota, por isso o servidor, ao notar esta situação deve atualizar o nome do ônibus no aplicativo pelo nome de ida novamente, para que não ocorra erros ao buscar os ônibus para os usuários.

Figura 31 - Tela de simulação indicando que o ônibus chegou em seu ponto inicial



Fonte: Do próprio autor, 2019

4.3.2 Aplicativo do usuário com deficiência visual

Para realizar os testes do aplicativo destino aos usuários foi criado uma tela para simulação de latitude e longitude similar à do aplicativo do motorista de ônibus, porém com uma informação a mais, o nome do ônibus que o usuário busca. No aplicativo final, o nome do ônibus deveria ser selecionado automaticamente e enviado para o servidor a partir do momento que o usuário digitasse o destino desejado ou através do microfone, falar qual local deseja ir e apertasse o botão na tela de iniciar rota.

Foram adicionados ao banco de dados latitudes e longitudes de pontos de ônibus fictícios para que fossem feitos as análises e comparações.

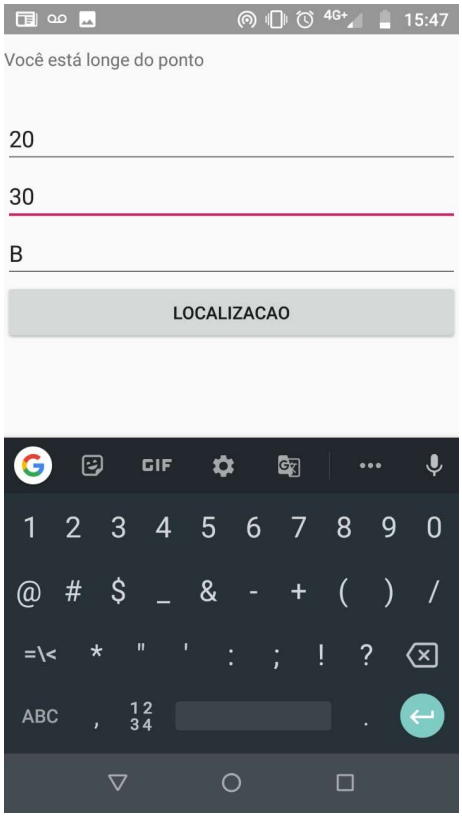
4.3.2.1 Usuário longe do ponto

A Figura 32 mostra a simulação de uma latitude, uma longitude e um nome para o ônibus que o usuário busca fictícios, sendo que o primeiro valor representa a latitude do usuário, o segundo valor a longitude e o último campo para o texto representa o nome do ônibus que o usuário busca.

O botão simula a mudança de posicionamento do usuário, e no instante que fosse mudado sua localização os novos dados de latitude e longitude deveriam ser enviados para o servidor para que fossem atualizados e analisados.

O momento em que o usuário se movimenta é simulado pela tela representada pela Figura 32, ou seja, sua latitude e longitude se alteram, porém, ele não se encontra próximo ao ponto de ônibus, onde na parte superior da tela é mostrada a resposta vinda do servidor.

Figura 32 - Tela de simulação indicando que o usuário está longe do ponto

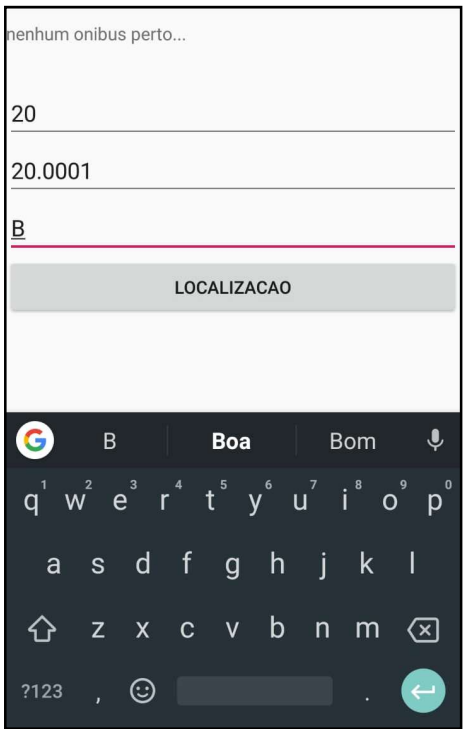


Fonte: Do próprio autor, 2019

4.3.2.2 Nenhum ônibus perto

O aplicativo do usuário só deveria analisar se há ônibus que ele precise embarcar caso o usuário estivesse próximo a um ponto de ônibus. A Figura 33 representa o momento em que o usuário se encontra próximo ao ponto de ônibus, porém não é encontrado nenhum ônibus com o nome de “B” próximo ao usuário, a resposta do servidor é apresentada na parte superior da tela.

Figura 33 - Tela de simulação indicando que não há nenhum ônibus útil próximo



Fonte: Do próprio autor, 2019

4.3.2.3 Ônibus chegando

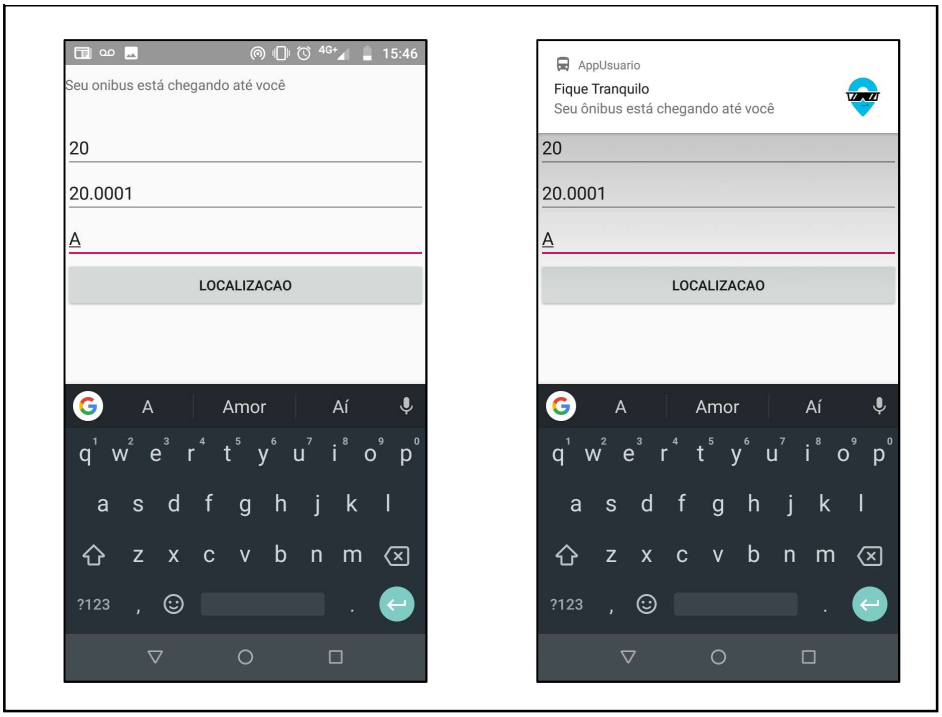
No momento em que o usuário estivesse dentro de um raio de 15 metros do ponto de ônibus e fosse identificado que há um ônibus que tenha o mesmo nome que ele procura e esteja dentro de um raio de dois quilômetros então, o aplicativo deveria avisar o usuário a situação por meio de uma notificação.

A Figura 34 simula uma latitude e uma longitude de um usuário que está dentro de um raio de 15 quilômetros do ponto de ônibus e procura pelo ônibus com nome “A”, no caso, foi modificado a latitude e longitude de um ônibus com o nome atual de “A” para próximo ao ponto de ônibus e ao usuário, dessa forma, a reposta proveniente

do banco de dados é positiva e informa que o ônibus que o usuário precisa embarcar está próximo.

A Figura 34 mostra a notificação que é enviada quando é notado que que há um ônibus próximo ao usuário que está no ponto de ônibus.

Figura 34 – Notificação recebida ao ter a localização próxima ao ponto e o ônibus



Fonte: Do próprio autor, 2019

5. CONSIDERAÇÕES FINAIS

Baseado no desenvolvimento e nos resultados obtidos do presente trabalho, podemos concluir que a maior parte dos principais objetivos estabelecidos no início do projeto foram concluídos por meio de etapas gradativas.

O primeiro ponto de atenção concluído foi a ativação do microfone através de um *Float Action Button* e a realização do reconhecimento da fala o usuário e então convertido sua fala em um texto que posteriormente seria convertido para um ponto com valores de latitude e de longitude. Dessa forma, é possível aplicar este reconhecimento de voz em várias etapas do aplicativo para facilitar a usabilidade para usuários com deficiência visual, não sendo necessário a ações como digitar textos ou escolher opções na maior parte do aplicativo.

Outra etapa importante para a continuidade do projeto foi a possibilidade de converter um texto em coordenadas de latitude e longitude, pois, para selecionar o local de destino, o usuário precisará digitar descrições deste local como por exemplo o nome da rua, ou do bairro e etc, dificilmente o usuário saberá as coordenadas de latitude e longitude. Como para a geração de possíveis rotas e para a geolocalização são necessários estes pontos com latitude e longitude, converter um texto nestas coordenadas se tornou uma etapa importante do projeto que foi concluída.

Como a proposta do aplicativo é fornecer alternativas de deslocamento de usuários principalmente com deficiência visual foi necessário dar a ele a opção de escolha, se ele gostaria de se deslocar até seu destino com transporte público ou privado. Por isso, a integração do aplicativo com o aplicativo de Taxis UBER foi importante e foi parcialmente concluída, porém através do aplicativo é possível a transição de aplicativo para o outro, o que facilita o processo de requisição de uma corrida pelo UBER caso ele esteja utilizando o aplicativo desenvolvido no presente trabalho. O aplicativo é um protótipo e por esse motivo foi implementado apenas a interface com o aplicativo UBER apenas para validar a proposta do projeto.

A principal etapa concluída do projeto foi a criação do banco de dados que realizou a interface entre os aplicativos e a comunicação através do envio de dados e respostas vindas do servidor. Por meio deste banco de dados criado, foi possível que fosse reconhecido as posições dos usuários no espaço, a rota que os motoristas estavam fazendo para que fossem avisados quando um usuário, que busca aquela linha de ônibus, estivesse no ponto e ao mesmo tempo, avisar apenas que o ônibus

que fosse necessário embarcar estivesse chegando próximo ao usuário. Por meio deste servidor criado no PythonAnywhere foi possível o cálculo da distância de cada um para fossem tratados os dados e se completasse o principal objetivo do projeto.

Um dos pontos essenciais que foram concluídos foi o reconhecimento das respostas vindas do banco de dados para que pudessem ser tratadas e ser realizado tarefas específicas para cada resposta, para isso foi necessário a implementação de conversões de respostas para tipos específicos que pudessem ser reconhecidos.

Após finalizado a estruturação do *software* só foi possível a validação do projeto por meio das simulações que foram realizadas, por isso, a etapa das telas de simulação que foram concluídas trouxeram para o projeto a possibilidade de legitimar o funcionamento das logicas implementadas em cada etapa do projeto.

Um ponto de atenção que foi finalizado foi o desenvolvimento da notificação que deveria ser enviada para o celular em casos específicos. Esta etapa foi estabelecida no início do projeto como um ponto importante pois de nada adiantaria o reconhecimento de ônibus próximos ao usuário que fosse necessário embarcar para chegar ao seu destino se o usuário tiver alguma deficiência visual, assim ele precisaria ser avisado através da notificação, caso chegasse notificação referente ao aplicativo então o usuário saberia que o seu ônibus se aproxima, por isso esta etapa concluída foi de extrema importância para o fechamento da proposta principal do projeto.

Além de tudo, a motivação deste presente trabalho foi criar uma ferramenta sem custo que possa ajudar as pessoas que sofrem com algum tipo de deficiência, por isso, desenvolver esta ferramenta com um banco de dados que seja alimentado apenas pelo desenvolvedor deste projeto é limitar a extensão desta ferramenta não atingindo seu objetivo principal.

Por isso, criou-se a possibilidade do desenvolvimento de um Website em que pessoas de todo o mundo possam adicionar pontos de ônibus e estações de trem ou metrô das suas regiões para que possam usufruir do aplicativo nas suas cidades em todo mundo, dessa forma, a motivação principal do projeto que é ajudar a todos, poderá, finalmente, ser atingida.

Como proposta para trabalhos futuros ficam alguns tópicos como por exemplo, a recuperação da localização do usuário no instante em que ele entra no aplicativo. Outro ponto que poderá ser desenvolvido em projetos futuros para complementar este projeto é a implementação foi integrar o aplicativo à API de rotas do Google que geram

automaticamente as rotas a partir das duas coordenadas, e por isso, não foi possível também a recuperação das informações provenientes desta API como o nome do ônibus que o usuário precisa embarcar.

Também, devido ao fato da possibilidade de monitorar os ônibus, como proposta de trabalho futuro, é possível que seja feito uma análise dos trajetos e um servidor monitorar o momento em que deve ser feito a manutenção no ônibus com base nos quilômetros percorridos. Outra possibilidade que este projeto cria é a gamificação no aplicativo do motorista pois, com isso, é possível monitorar em muitos ônibus diversos parâmetros em suas viagens.

6. REFERÊNCIAS BIBLIOGRÁFICAS

Afonso, Ricardo. et al. SmartCluster: Utilizando Dados Públicos para Agrupar Cidades Inteligentes por Domínios Alternative Title: SmartCluster: Using Public Data to Group Smart Cities by Domains. 2015.

BATISTA, André Luiz França; BAZZO, Walter Antônio. Questões contemporâneas e desenvolvimento de aplicativos móveis: onde está a conexão? R. B. E. C. T., v. 8, n. 4, set-dez.2015 ISSN - 1982-873X

BARBOSA, Adriana Silva. **Mobilidade urbana para pessoas com deficiência no Brasil: um estudo em blogs.** 2015. 13 f. Tese (Doutorado) - Curso de Política Científica e Tecnológica, Universidade Estadual de Campinas, Campinas, 2015.

BARRA, Daniela Couto Carvalho et al. **MÉTODOS PARA DESENVOLVIMENTO DE APLICATIVOS MÓVEIS EM SAÚDE: REVISÃO INTEGRATIVA DA LITERATURA.** Florianópolis. 2017.

BENITES, Ana Jane. **ANÁLISE DAS CIDADES INTELIGENTES SOB A PERSPECTIVA DA SUSTENTABILIDADE: O CASO DO CENTRO DE OPERAÇÕES DO RIO DE JANEIRO.** 2016. 224 f. Dissertação (Mestrado) - Curso de Política Científica e Tecnológica, Universidade Estadual de Campinas, Campinas, 2016.

BLOIS, M., Choren, R., Laufer, C., Ferraz, F. & Fuks, H. (1999) Desenvolvendo Aplicativos para a Web ' com o Scriba, Anais do XXVI SEMISH - Seminário Integrado de Software e Hardware, Sociedade ' Brasileira de Computação (SBC), Rio de Janeiro, julho 1999, pp. 119-133

BOHUSCH, Graziela. **MOBILIDADE URBANA SUSTENTÁVEL: UMA PROPOSTA DE VISÃO AMPLIADA DO CONCEITO.**2013. 178 f. Tese (Doutorado) - Curso de Geografia, Universidade Federal de Santa Catarina, Florianópolis, 2013.

CAMARGO, Azael; BOTELHO, Carlos. ESPAÇOS INTELIGENTES, CIDADES DA INTELIGÊNCIA E REGIÕES DINÂMICAS EM INOVAÇÃO: AS NOVAS TECNOLOGIAS E A CONFIGURAÇÃO URBANA E REGIONAL. 2018

CAMPOS, Vânia Barcellos Gouvêa. **UMA VISÃO DA MOBILIDADE URBANA SUSTENTÁVEL.** 2006.

CARON, Julie; BIDUSKI, Daiana; DE MARCHI, Ana Carolina Bertoletti. Alz Memory – um aplicativo móvel para treino de memória em pacientes com Alzheimer. RECIIS – Rev Eletron de Comun Inf Inov Saúde, Passo Fundo. 2015 abr.-jun. ISSN 1981-6278

CARVALHO, Inês Rodrigues Urbano Fernandes de. **Análise à Mobilidade Inteligente Urbana de Pessoas:** Caso da cidade do Porto. 2017. 132 f. Dissertação (Mestrado) - Curso de Especialização em Serviços, Universidade Católica Portuguesa, Porto, 2017.

CATUNDA, Luciana de Andrade; SANTANA, Antônia Neide Costa. MOBILIDADE URBANA NA CIDADE DE SOBRAL/CE: DISCUSSÃO DE CONCEITOS E CONSTATAÇÕES PRELIMINARES. Revista Casa da Geografia de Sobral, Sobral v.17, n.1, p.160-177, 2015. ISSN 1516-7712

CONSTRUINDO CIDADES SUSTENTÁVEIS: Síntese do C40 São Paulo Climate Summit 2011. São Paulo: Secretaria Municipal de Desenvolvimento Urbano e Universidade de São Paulo, jun. 2012.

COSTA, Marcela da Silva. **UM ÍNDICE DE MOBILIDADE URBANA SUSTENTÁVEL.** 2008. 274 f. Tese (Doutorado) - Curso de Engenharia Civil, Universidade de São Paulo, São Carlos, 2008

COSTA, Marcela da Silva. **MOBILIDADE URBANA SUSTENTÁVEL: UM ESTUDO COMPARATIVO E AS BASES DE UM SISTEMA DE GESTÃO PARA BRASIL E PORTUGAL.** 2003. 196 f. Dissertação (Mestrado) - Curso de Engenharia, Universidade de São Paulo, São Carlos, 2003.

CUNHA, Maria Alexandra et al. **SMART CITIES: TRANSFORMAÇÃO DIGITAL DE CIDADES.** São Paulo: Eaesp, 2016. 164 p.

D'CARLO, Deborah; BARBOSA, Glívia Angélica Rodrigues; OLIVEIRA, Erica Rodrigues. Usabilidade em Aplicativos Móveis Educacionais: Um Conjunto de Heurísticas para Avaliação. In: V CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 5., 2016, Uberlândia. Minas Gerais. 2016. p. 777 - 786.

DUARTE, Fábio. CIDADES INTELIGENTES: inovação tecnológica no meio urbano. **São Paulo em Perspectiva,,** São Paulo, v. 19, n. 1, p.122-131, mar. 2005.

EMMEL, Francini Priscila. **INDICADORES DE MOBILIDADE URBANA SUSTENTÁVEL: UM ESTUDO DE CASO NO CENTRO DE SANTA CRUZ DO SUL, RS.** 2017. 85 f. TCC (Graduação) - Curso de Engenharia Civíl, Universidade de Santa Cruz do Sul, Santa Cruz do Sul, 2017.

GALVÃO, Marco Aurélio; ZORZAL, Ezequiel Roberto. **Aplicações Móveis com Realidade Aumentada para Potencializar Livros.** 2012. 10 f. Universidade de São Paulo, São Paulo, 2012.

GUIMARÃES, Patrícia Borba Vilar; XAVIER, Yanko Marcius de Alencar. SMART CITIES E DIREITO: CONCEITOS E PARÂMETROS DE INVESTIGAÇÃO DA GOVERNANÇA URBANA CONTEMPORÂNEA. **Revista de Direito da Cidade**, [s.l.], v. 8, n. 4, p.1362-1380, 27 nov. 2016. Universidade de Estado do Rio de Janeiro. http://dx.doi.org/10.12957/rdc.2016.23685.

GOMES, Tancicleide C. S.; MELO, Jeane C. B. de. App Inventor for Android: Uma Nova Possibilidade para o Ensino de Lógica de Programação. In: II CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (CBIE 2013), 2., 2013, Recife. Limeira. 2013. p. 620 - 629.

GOMES, Ruthie Bonan; GARCIA, Ana Luiza Casasanta. A FALTA DE ACESSIBILIDADE URBANA PARA PESSOAS COM DEFICIÊNCIA E SUAS IMPLICAÇÕES EM SAÚDE MENTAL E GARANTIA DE DIRETOS HUMANOS. Cadernos Brasileiros de Saúde Mental, Florianópolis, v.9, n.24, p.230-253, 2017. ISSN 1984-2147

GOMIDE, Alexandre de Ávila. **MOBILIDADE URBANA, INIQÜIDADE E POLÍTICAS SOCIAIS.** 2006.

GUTIERREZ, Andrea et al. **Mobilidade Urbana:** Desafios e Sustentabilidade. São Paulo: Mack Pesquisa, 2016.

KOMNINOS, Nicos. CIDADES INTELIGENTES. Sistemas de Inovação e Tecnologias da Informação ao serviço do Desenvolvimento das Cidades. 2007

LEMOS, André. Cidades inteligentes: De que forma as novas tecnologias — como a computação em nuvem, o Big Data e a Internet das Coisas — podem melhorar a condição de vida nos espaços urbanos? **Espaços Urbanos,** Bahia, v. 12, n. 2, p.46-49, dez. 2013.

MAGAGNIN, Renata Cardoso; SILVA, Antônio Nélson Rodrigues. A percepção do especialista sobre o tema mobilidade urbana. TRANSPORTES, v. XVI, n. 1, n. 1, p. 25-35, 2008. ISSN: 1415-7713.

MELO, Rafaela da Silva; CARVALHO, Marie Jane Soares. **APLICATIVOS EDUCACIONAIS LIVRES PARA MOBILE LEARNING. Universidade do Rio Grande do Sul.** 2014.

MELLO, Cleusimari M. Colombo; SGANZERLA, Maria Adelina R.. **APLICATIVO ANDROID PARA AUXILIAR NO DESENVOLVIMENTO DA COMUNICAÇÃO DE AUTISTAS.** 2013. Universidade Luterana do Brasil, Gravataí, 2013.

NICHELE, Aline Grunewald; SCHLEMMER, Eliane. Aplicativos para o ensino e aprendizagem de Química. In: CINTED- NOVAS TECNOLOGIAS NA EDUCAÇÃO, 2., 2014, Porto Alegre. 2014. v. 12, p. 1 - 9.

OLIVEIRA, Rafael et al**. Recursos para Desenvolvimento de Aplicativos com Suporte a Reconhecimento de Voz para Desktop e Sistemas Embarcados.** 2018. Disponível em: <https://www.researchgate.net/publication/267709371_Recursos_para_Desenvolvimento_de_Aplicativos_com_Suporte_a_Reconhecimento_de_Voz_para_Desktop_e_Sistemas_Embarcados>. Acesso em: 11 set. 2018.

OLIVEIRA, Thiago Robis de; COSTA, Francielly Morais Rodrigues da**. Desenvolvimento de aplicativo móvel de referência sobre vacinação no Brasil.** 2012.

PALUDO, Lauriana. **UM ESTUDO SOBRE AS TECNOLOGIAS JAVA DE DESENVOLVIMENTO DE APLICAÇÕES MÓVEIS.** 2003. 118 f. Monografia

(Especialização) - Curso de Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2003.

PAULA, Leonam João Leal de. **Desenvolvimento de aplicativo para dispositivos móveis para coleta de dados georreferenciados através de reconhecimento de voz.** 2013. 82 f. Dissertação (Mestrado) - Curso de Engenharia de Sistemas Agrícolas, Universidade de São Paulo, Piracicaba, 2013.

REZENDE, Denis Alcides. Cidade Digital Estratégica: Modelo e Aplicação em um Município Paulista. In: VII SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO, 7., 2012, São Paulo. São Paulo. 2012. p. 1 - 6.

SCARINGELLA, Roberto Salvador**. A CRISE DA MOBILIDADE URBANA EM SÃO PAULO.** São Paulo. 2001.

SCHMITT, Bruna Bianc; BADALOTTI, Claudine Machado. Acessibilidade para portadores de necessidades especiais no centro histórico da cidade de Itá – SC. **Revista de Arquitetura IMED,** Passo Fundo, v. 6, n. 2, p. 45-70, dez. 2017. ISSN 2318-1109.

SILVA, Antonio Rodrigo S. et al. Especificação e desenvolvimento de um ambiente educativo móvel para a prática da escrita Braille. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 3., 2014, Dourados. Fortaleza. 2014. p. 431 - 440.

SILVA, Marcelo Moro da; SANTOS, Marilde Terezinha Prado. Os Paradigmas de Desenvolvimento de Aplicativos para Aparelhos Celulares. T.I.S, São Carlos, v. 3, n. 2, p 1-9, ago. 2014. ISSN 2316-2872.

SOUZA, Rafael Celestino de et al. **Processo de criação de um aplicativo móvel na área de odontologia para pacientes com necessidades especiais.** 2014. Disponível em: <https://docs.google.com/document/d/1QactHI7ITZ_nO8Ka4XdqfFcTK1Ef7KayDcvDLc9LY9Q/edit>. Acesso em: 11 set. 2018

STRAPAZZON, Carlos Luiz. CONVERGÊNCIA TECNOLÓGICA NAS POLÍTICAS URBANAS: PEQUENAS E MÉDIAS “CIDADES INTELIGENTES”. 2009

TIBES, Chris Mayara dos Santos; DIAS, Jessica David; ZEM-MASCARENHAS, Silvia Helena. Mobile applications developed for the health sector in Brazil: an integrative literature review. **Reme**: Revista Mineira de Enfermagem, [s.l.], v. 18, n. 2, p.471-478, 2014. GN1 Genesis Network.

Weiss, Marcos Cesar; Bernardes, Roberto Carlos; Consoni, Flavia Luciane. Cidades inteligentes: a aplicação das tecnologias de informação e comunicação para a gestão de centros urbanos. Tecnologia e Sociedade, vol. 9, núm. 18, 2013.

Weiss, Marcos Cesar; Bernardes, Roberto Carlos; Consoni, Flavia Luciane. Cidades inteligentes: casos e perspectivas para as cidades brasileiras. Revista Tecnológica da Fatec Americana, vol. 05, n. 01, 2017.

GOOGLE. **Android Developers**. [*S. l.*], ca. 2016. Disponível em: https://developer.android.com/?hl=en. Acesso em: 1 fev. 2019.

7. Apêndice A – Banco de Dados e Aplicativo

https://github.com/Muriloceolin/AplicativoPontoDeVista/tree/master