



Universidade Católica do Salvador
Bacharelado em Engenharia de Software

Carlos Daniel Santana Cruz

Desenvolvimento de um Sistema IOT Voltado à Acessibilidade
na Mobilidade Urbana

Salvador

2019

Carlos Daniel Santana Cruz

**Desenvolvimento de um Sistema IOT Voltado à
Acessibilidade na Mobilidade Urbana**

Trabalho de Conclusão de Curso apresentado à Universidade Católica do Salvador como parte dos requisitos necessários para a obtenção do Título de Bacharelado em Engenharia de Software.
Orientador: Prof. Me. André Brasil Vieira Wyzykowski

Universidade Católica do Salvador
Bacharelado em Engenharia de Software

Salvador
2019

Carlos Daniel Santana Cruz

Desenvolvimento de um Sistema IOT Voltado à Acessibilidade na Mobilidade Urbana

Trabalho de Conclusão de Curso apresentado à Universidade Católica do Salvador como requisito parcial para a obtenção do título de Bacharelado em Engenharia de Software.

Comissão Examinadora

Prof. Me. André Brasil Vieira Wyzkowski
Universidade Católica do Salvador
Orientador

Prof. Me. Marcelo Índio dos Reis
Universidade Católica do Salvador

Prof. Mario Jorge Pereira
Universidade Católica do Salvador

Salvador, 1 de agosto de 2019

Agradecimentos

Agradeço, primeiramente a Deus, por me possibilitar concluir esta etapa da minha vida. Agradeço a minha família (meu pai e minha mãe (Carlos Humberto e Macleia), meus irmãos (Felipe e Samuel), meus tios (Charles e Debora)), por estarem comigo, por serem a base na qual confio e por me apoiarem em minhas decisões. Agradeço a minha avó Cleuza, pois, sem tudo o que ela já fez por mim, desde que nasci até hoje, eu não seria o que sou. Agradeço a meu orientador, André Brasil, por toda a ajuda, sugestões, por seus ensinamentos e por sua prontidão em me auxiliar neste projeto. Agradeço também a todos os colegas do grupo de orientandos e formandos: Hudson, Alan, Felipe, Gabryela, Lielson, pelas sugestões mútuas nos trabalhos, pelos momentos de distração, e por compartilharmos essa etapa do curso; e Danilo, por seu trabalho e ajuda com o template utilizado. Agradeço a meu amigo (e também formando) Murilo Guerreiro, por toda a ajuda durante o trabalho, pelas noites em claro codificando, escrevendo e trocando ideias sobre os respectivos trabalhos via vídeo conferência. Agradeço também a meus amigos Jeferson, Carlos e Lucas, por toda a parceria ao longo do curso. Agradeço também ao coordenador Osvaldo, e a todos os professores que me ensinaram ao longo do curso, os quais passei a ter como fonte de inspiração devido à suas histórias, seus ensinamentos e suas características individuais. Agradeço a todos os membros da comunidade evangélica CECUC, por todo o apoio e carinho durante estes últimos anos. Por fim, mas não menos importante, agradeço a todos aqueles que contribuíram direta ou indiretamente na concretização desta etapa de minha vida. Deixo aqui, a todos os que citei, e aos que eu eventualmente possa ter esquecido, os meus mais sinceros e profundos agradecimentos.

Resumo

Existiam no Brasil, até o ano de 2010, cerca de 11 milhões de pessoas com algum tipo de deficiência visual. No entanto, mesmo com essa quantidade expressiva de pessoas, e apesar da existência de algumas políticas de mobilidade urbana e leis voltadas à acessibilidade, alguns aspectos do cotidiano dessas pessoas ainda se constituem como um problema. Dentre esses problemas, encontra-se a falta de acessibilidade nos transportes públicos. Com base no problema mencionado, este trabalho destinou-se a propor um sistema composto por um dispositivo IOT embarcado, juntamente com uma aplicação web e um aplicativo para dispositivos móveis, todos eles com o propósito de contribuir para a melhoria na acessibilidade da mobilidade urbana, mais especificamente no que se refere aos transportes públicos, consequentemente contribuindo para uma vida mais autônoma e independente para pessoas com alguma deficiência. Para isto, primeiramente, realizou-se uma revisão bibliográfica com base nos temas relacionados ao trabalho; a seguir, fez-se o levantamento de requisitos para as funcionalidades iniciais que o sistema possuiria; por fim, são descritas as etapas percorridas no desenvolvimento das partes que compõem o sistema, juntamente com alguns problemas encontrados, e como esses problemas foram trabalhados. O sistema desenvolvido, mesmo tratando-se de uma primeira versão experimental, possibilitou a uma pessoa com deficiência visual o acesso a estimativas de distância entre sua localização e um veículo específico, e em um experimento controlado, possibilitou que o veículo pudesse ser identificado pela pessoa. Então, juntamente com a conclusão do trabalho, planejou-se para projetos futuros, dentre outras características, a adição de funcionalidades que possibilitem ao usuário uma gama maior de informações ao longo de uma viagem.

Palavras-Chave: Internet das Coisas. Acessibilidade. Mobilidade Urbana. Aplicativo Móvel.

Abstract

In Brazil, until the 2010 year, there were about 11 million people with some visual impairment. However, even with this significant quantity of people, and despite the existence of some urban mobility policies and laws related to accessibility, some aspects of their daily lives still constitute a problem. Among these problems is the lack of accessibility in public transport. Based on the mentioned problem, this work aimed to propose a system composed of an embedded IOT device, together with a web application and an application for mobile devices, all with the purpose of contributing to the improvement in accessibility and urban mobility, more specifically regarding to the public transport modality, consequently contributing to a more autonomous and independent life to people with some impairment. First, a bibliographic review was carried out based on the themes related to the work; then, a Requirements analysis was made for the initial functionalities that the system would have; finally, are described the steps taken in the development of the parts that compose the system, along with some problems encountered, and how these problems were worked out. The developed system, even being a first experimental version, made it possible for a visually impaired person to access estimated distances between their location and a specific vehicle's location, and in a controlled experiment, enabling the vehicle to be identified by the person. Due to this, with the conclusion of this work, it was planned for future projects, among other characteristics, the addition of functionalities, in order to offer to the user a greater range of information along a bus trip.

Keywords: Internet of Things. Accessibility. Urban Mobility. Mobile App.

Lista de Siglas e Abreviaturas

UCSAL *Universidade Católica do Salvador*

IOT *Internet of Things (Internet das Coisas)*

APP *Aplicativo (para dispositivos móveis)*

REST *Representational State Transfer*

API *Application Programming Interface (Interface de Programação de Aplicativos)*

GET *Método de requisição HTTP*

POST *Método de requisição HTTP*

HTTP *Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto)*

Sumário

1	INTRODUÇÃO	13
1.1	Problema e Motivação	15
1.2	Objetivos	15
1.3	Estrutura do Documento	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Acessibilidade e Mobilidade Urbana	17
2.1.1	Barreiras encontradas por deficientes visuais no âmbito de transporte urbano	18
2.1.2	Assistências a deficientes visuais já presentes em transportes públicos	18
2.1.3	Acessibilidade em aplicativos Móveis	20
2.2	Soluções Existentes Voltadas ao Transporte Público	22
2.3	Soluções IOT Embarcadas	22
2.3.1	Estado da Arte	23
2.4	Considerações Finais	24
3	DESENVOLVIMENTO	25
3.1	Descrição e Projeto do Sistema Proposto	25
3.2	Planejamento Inicial	26
3.3	Desenvolvimento do Aplicativo Móvel	27
3.3.1	Elaboração das Telas	28
3.3.2	Posição e Cálculo de Distância	30
3.3.3	Utilização de <i>Api's</i> e Integração com o Servidor	31
3.4	Desenvolvimento do Sistema a Rodar no Servidor	32
3.4.1	Comunicação com Clientes em Tempo Real	35
3.4.2	Cálculo de Distância	36
3.5	Desenvolvimento da Aplicação <i>IOT</i> Embarcada	36
3.5.1	Obtenção de Posição e Velocidade	37
3.5.2	Conectividade via GPRS	37
3.6	Considerações Finais	38
4	EXPERIMENTOS E RESULTADOS	39
4.1	Descrição do Experimento	40
4.2	Resultados dos Experimentos In Vitro	42

5	CONCLUSÕES	45
5.1	Trabalhos Futuros	45
	REFERÊNCIAS BIBLIOGRÁFICAS	47
6	APÊNDICE	50

1 Introdução

De acordo com o último censo realizado pelo IBGE(ESTATÍSTICA, 2010a), no ano de 2010, existiam no Brasil cerca de 46 milhões de pessoas (aproximadamente 23% da população total), que declararam possuir algum tipo de deficiência(ESTATÍSTICA, 2010b). Desse grupo, cerca de 25% relataram algum tipo de deficiência relacionada à visão.

A Organização Mundial da Saúde (OMS) estima que, no mundo inteiro, aproximadamente 1,3 bilhões de pessoas vivem com algum tipo de deficiência visual((WHO), 2018). Dessas, cerca de 188,5 milhões apresentam leve perda da visão, 217 milhões apresentam grau de perda da visão entre moderada a severa, e 36 milhões de pessoas são cegas ((WHO), 2018).

A Organização das Nações Unidas (ONU), em 2006 adotou a convenção sobre os direitos da pessoa com deficiência, que estabelece vários aspectos, dentre eles,

“Os Estados Partes deverão tomar medidas efetivas para assegurar às pessoas com deficiência sua mobilidade pessoal com a máxima autonomia possível:

1. Facilitando a mobilidade pessoal das pessoas com deficiência, na forma e no momento em que elas quiserem, a um custo acessível;
2. Facilitando às pessoas com deficiência o acesso a tecnologias assistivas, dispositivos e ajudas técnicas de qualidade, e formas de assistência direta e intermediária, tornando-os disponíveis a um custo acessível;
3. Propiciando às pessoas com deficiência e ao pessoal especializado uma capacitação sobre habilidades de mobilidade; e
4. Incentivando entidades que produzem ajudas técnicas de mobilidade, dispositivos e tecnologias assistivas a levarem em conta todos os aspectos relativos à mobilidade de pessoas com deficiência.”.

No entanto, mesmo com essa quantidade expressiva de pessoas, e apesar da existência de algumas políticas de mobilidade urbana e leis voltadas à acessibilidade, alguns aspectos do cotidiano ainda se constituem como um problema. Dentre esses problemas, e sobre o qual este trabalho se destinou a investigar e contribuir para sua melhoria, encontra-se a falta de acessibilidade nos transportes públicos, juntamente com a ausência de um meio que possibilite a uma pessoa com alguma deficiência visual a identificar com precisão um ônibus no qual ela precise embarcar.

Nesse contexto, o crescimento no desenvolvimento de tecnologias assistivas se apresenta como um grande contribuidor no objetivo de proporcionar às pessoas com algum tipo de deficiência uma maior autonomia e melhor qualidade de vida.

Juntamente com a capacidade dessas tecnologias de influenciarem o cotidiano das pessoas, uma outra área que já desempenha este papel, e que tem apresentado um enorme crescimento, é a internet das coisas.

Para vários autores, como por exemplo (MULFARI et al., 2015), (HOLLIER; ABOU-ZAHRA, 2018), (BHOWMICK; HAZARIKA, 2017), (KUMARI, 2015), dispositivos desenvolvidos englobando características dessas duas áreas apresentam diversas vantagens para o usuário final, tendo o potencial de melhorar ainda mais o seu cotidiano. Isto se deve ao fato das características e funcionalidades presentes em produtos com tecnologia assistiva se beneficiarem diretamente dos avanços no que se refere à internet das coisas, como por exemplo na velocidade com que as informações podem ser processadas e retornadas ao usuário.

Em (KUMARI, 2015), é analisado os impactos da utilização de tecnologias assistivas por estudantes com alguma deficiência visual. Os resultados do trabalho são voltados ao âmbito acadêmico, e aponta um crescimento na quantidade de estudantes com alguma deficiência que estão buscando níveis maiores de escolaridade. É sugerido que isto tem acontecido

“due to high availability and accessibility of technologies by visually impaired children and especially in their academics”.

Já em *"Achieving Assistive Technology Systems Based on IoT Devices in Cloud Computing"*, é proposto pelos autores, a criação de um dispositivo de tecnologia assistiva capaz de interagir com quaisquer outros dispositivos em um ambiente.

“Specifically, we propose the usage of embedded systems to interface sensors and existing AT equipments to whichever computer-based devices[...]. Our solution allows a person with a disability to use his/her own personalized AT tool everywhere for interacting with many computer-based devices” (MULFARI et al., 2015).

Os dois trabalhos citados anteriormente demonstram o potencial dessas áreas no que se refere à sua atuação no cotidiano das pessoas, juntamente com a abrangência que essas tecnologias podem alcançar.

Hollier e Abou-Zahra (2018) apontam que o maior potencial da IOT para pessoas com alguma deficiência se relaciona à características como conectividade em tempo real, *“[...] which can ensure that people can quickly and easily obtain assistance and support. This can result in another avenue to achieving a good quality of life and facilitating participation, both socially and economically”* (HOLLIER; ABOU-ZAHRA, 2018).

Uma análise feita por Bhowmick e Hazarika (2017) revelou que existe um crescente interesse na área de dispositivos de tecnologias assistivas, principalmente no que se refere à publicações acadêmicas. Os autores apontam que houve um crescimento no volume de publicações nessa área, partindo de menos de 50 publicações por ano durante a década de XC, para cerca de 400 publicações por ano em 2014. *“Assistive Technology for persons with visually impairments is expected to grow at a swift pace and impact the lives of individuals and the elderly in ways not previously possible”* (BHOWMICK; HAZARIKA, 2017).

1.1 Problema e Motivação

De acordo com pesquisa publicada no Seminário Nacional NTU 2017 & Transpúblico, “O transporte figura entre os principais problemas urbanos percebidos nos municípios brasileiros. Assim como na pesquisa realizada pela NTU em 2006, o setor foi o quarto mais citado [...]”(URBANOS, 2017). Isso se dá devido a vários fatores, tais como superlotação, falta de informação, ou má qualidade na prestação de serviço.

Somado a estes fatores, uma pessoa com deficiência visual que precise recorrer aos meios de transporte público para se deslocar ainda encontra outros empecilhos, tais como a falta de calçadas adequadas, falta de sinalização nos pontos de ônibus, dentre outros. No entanto, um dos principais problemas se dá na falta de uma maneira que possibilite à pessoa com deficiência visual a identificação dos ônibus que possam atender ao seu trajeto.

Devido à utilização dos serviços de transporte público (ônibus, vans, etc) por parte de pessoas com deficiência visual ainda depender muito de alguém para reconhecer o veículo em questão, e nem sempre existir uma pessoa para prestar esse auxílio, o projeto busca utilizar a tecnologia para proporcionar mais um meio de independência para essas pessoas.

1.2 Objetivos

O objetivo deste trabalho visou o estudo, o desenvolvimento e a validação de um dispositivo que, embarcado em veículos de transporte público, e em conjunto com um aplicativo para dispositivos móveis, possibilite a pessoas portadoras de alguma deficiência visual, a identificação com maior precisão dos veículos específicos.

Para isso, os objetivos específicos do trabalho foram:

1. Pesquisar e conhecer soluções já existentes e/ou ideias e conceitos publicados, estejam elas disponíveis no mercado ou apenas documentadas em outros trabalhos acadêmicos;
2. confeccionar um dispositivo eletrônico a ser embarcado em veículos, capaz de atualizar sua localização junto a uma aplicação rodando em um servidor web;
3. Desenvolver uma solução de software, composta por um aplicativo para dispositivos móveis e uma aplicação para servidor, com o objetivo de interagir com o dispositivo confeccionado;
4. Testar e validar as funcionalidades dos sistemas elaborados;

1.3 Estrutura do Documento

O segundo capítulo deste documento se destina a uma abordagem dos principais temas que compõem esta pesquisa. Para isto, são utilizadas referências à publicações relevantes aos assuntos.

No terceiro capítulo, são descritas todas as etapas e procedimentos utilizados na elaboração do sistema proposto. No capítulo, também encontram-se presentes trechos de códigos relevantes, que possuem grande impacto no funcionamento das partes do sistema.

Já no quarto capítulo, são apresentados os testes que foram realizados com o sistema, juntamente com seus resultados.

Por fim, no último capítulo encontram-se as conclusões obtidas com o desenvolvimento e com os testes do aplicativo, juntamente com propostas de melhorias, extraídas a partir dos resultados alcançados até o momento da conclusão desta pesquisa.

2 Fundamentação Teórica

Para a realização deste trabalho, foram necessárias algumas etapas. Primeiramente, realizou-se uma revisão bibliográfica, tendo como fontes publicações de livros, manuais, mestrados e/ou doutorados, todos eles abrangendo todos os temas nos quais o trabalho se apoia, com o objetivo de construir uma sólida base teórica para o mesmo.

A abordagem da pesquisa é Quali-quantitativa, uma vez que almejou-se analisar os resultados obtidos estatisticamente e interpretar as opiniões emitidas pelos voluntários ao final dos experimentos.

A pesquisa tem o propósito de ser um estudo exploratório, com caráter experimental. De acordo com Gil no trabalho (GIL, 2002), a pesquisa “tem como objetivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a constituir hipóteses. Pode-se dizer que estas pesquisas têm como objetivo principal o aprimoramento de ideias ou a descoberta de intuições”.

Com relação a esta pesquisa, isso se dá devido a ela se destinar a desenvolver e testar uma solução para um problema vivenciado por deficientes visuais em seu dia a dia: a falta de acessibilidade nos serviços de transporte público, principalmente nos ônibus.

2.1 Acessibilidade e Mobilidade Urbana

No que diz respeito à acessibilidade, o Brasil possui a Lei Brasileira de Inclusão da Pessoa com Deficiência, Lei Nº 13.146, de 6 de julho de 2015, “destinada a assegurar e a promover, em condições de igualdade, o exercício dos direitos e das liberdades fundamentais por pessoa com deficiência, visando à sua inclusão social e cidadania” (BRASIL, 2015).

Ainda na referida lei, no artigo 3, é definido o conceito de acessibilidade como sendo a “possibilidade e condição de alcance para utilização, com segurança e autonomia, de espaços, mobiliários, equipamentos urbanos, edificações, transportes, informação e comunicação, inclusive seus sistemas e tecnologias, bem como de outros serviços e instalações abertos ao público, de uso público ou privados de uso coletivo, tanto na zona urbana como na rural, por pessoa com deficiência ou com mobilidade reduzida” (BRASIL, 2015).

Já o conceito de mobilidade urbana pode ser descrito como sendo a capacidade da população de se locomover de um ponto a outro, utilizando quaisquer modalidades de transporte (público, particular) ou até mesmo à pé.

Para (Carvalho, 2017)(CARVALHO, 2017), a mobilidade urbana está relacionada também à facilidade no deslocamento das pessoas, e (Magagnin e Silva 2008, p. 2)(SILVA,

2008) definem a mobilidade como “um atributo relacionado aos deslocamentos realizados por indivíduos nas suas atividades de estudo, trabalho, lazer e outras”.

2.1.1 Barreiras encontradas por deficientes visuais no âmbito de transporte urbano

O Brasil possui também, a lei federal N° 12.587, de 3 de janeiro de 2012, lei esta que “Institui as diretrizes da Política Nacional de Mobilidade Urbana”. Esta lei estabelece como sendo alguns de seus objetivos e princípios, dentre os quais pode-se destacar:

1. Integração entre os diferentes modos de transporte e a melhoria da acessibilidade e mobilidade das pessoas e cargas no território do Município
2. Reduzir as desigualdades e promover a inclusão social
3. Proporcionar melhoria nas condições urbanas da população no que se refere à acessibilidade e à mobilidade

Ela também apresenta, em seu artigo 14, como sendo direitos dos usuários, dentre outros, “ser informado nos pontos de embarque e desembarque de passageiros, de forma gratuita e acessível, sobre itinerários, horários, tarifas dos serviços e modos de interação com outros modais[...]” e “ter ambiente seguro e acessível para a utilização do Sistema Nacional de Mobilidade Urbana [...]”.

No entanto, apesar de dispor de leis como as citadas anteriormente, pessoas com algum tipo de deficiência ainda encontram empecilhos no que se refere ao transporte urbano, mais especificamente com relação aos ônibus. No caso de pessoas com deficiência visual, um dos principais dificultadores nesta modalidade de transporte é a falta de informação acessível, estando a maioria das informações disponíveis apenas de forma visual (DISCHINGER, 2000).

2.1.2 Assistências a deficientes visuais já presentes em transportes públicos

Conforme relatado por Silveira e Dischinger (DISCHINGER, 2000), no que se refere ao transporte público, algumas cidades possuem um ou mais tipos de assistência à deficientes visuais, sendo o mais comum, os avisos sonoros.

Esses avisos geralmente indicam “[...] a próxima parada, linha, destino e outras informações sobre o sistema” (DISCHINGER, 2000).

Porém, esse tipo de informação, na maioria das vezes encontra-se presente apenas em modalidades mais fixas (como metrô, trens e similares).

Existem no entanto, relatos de soluções que se propõe a contribuir com a melhora nesse aspecto da mobilidade urbana. Bianchi (2014)(BIANCHI, 2014), relata alguns projetos voltados à essa área:

“Na cidade de São Carlos, em São Paulo, foi implantado um sistema chamado BusAlert[...]. Este sistema possui um aplicativo para celulares que avisa por meio de mensagem de texto e de voz a hora que um ônibus está se aproximando do ponto de parada. [...] Outro sistema, chamado DPS 2000, foi desenvolvido pela Universidade Federal de Minas Gerais [...]. O sistema funciona com dois aparelhos: um transmissor, que fica com o usuário, e o receptor, instalado no ônibus[...].”

Em Salvador, estudantes do Instituto Federal da Bahia desenvolveram um sistema que alerta aos usuários quando os ônibus se aproximam dos pontos. Este sistema funciona com um transmissor (instalado no ônibus) e um receptor (instalado no ponto), que acoplado a um auto-falante, emite um sinal sonoro, informando sobre o ônibus que se aproxima.

Outros países também possuem soluções já implantadas ou estudos dedicados a promover uma melhor acessibilidade nos sistemas de transporte público. A maioria delas utiliza dispositivos especificamente desenvolvidos para esse propósito.

Na Índia, autores(RAO., 2015) realizaram testes supervisionados que provaram a eficiência de um sistema composto por dois módulos, um para o usuário e outro para o ônibus. O sistema funciona da seguinte maneira: quando o usuário escuta um ônibus se aproximando do ponto de ônibus, ele ativa o módulo do usuário, transmitindo um sinal de rádio frequência para todos os ônibus nas proximidades. Cada ônibus responde através de seu módulo para o módulo do usuário, enviando o código que identifica sua rota. Todas as identificações recebidas são lidas pelo módulo do usuário, que então pode selecionar a rota que deseja. Ao selecionar uma rota, o ônibus correspondente emite um sinal de voz, através de um auto-falante, localizado na entrada do ônibus, possibilitando que a pessoa se locomova até o ônibus específico.

Em (WANG et al., 2014), os autores propõem um sistema sem fio, dividido em dois módulos, um para o passageiro e outro para o motorista. O sistema utiliza rádio frequência para transmitir dados entre os módulos. Quando o módulo do motorista recebe um conjunto de informações corretas, enviadas pelo módulo do passageiro, ele informa ao motorista sobre a presença de alguém que deseja embarcar no ônibus.

Os autores ainda citam alguns outros exemplos: na República Tcheca, existe o TY-FLOSET(LTD, 2010), um sistema de informações com o objetivo de auxiliar pessoas com alguma deficiência visual. No Canadá, um outro sistema foi desenvolvido, chamado iBus(CANADA, 2014). O sistema possibilita aos usuários o rastreamento em tempo real dos ônibus.

Na Finlândia, pesquisadores em conjunto com o Ministério de Transportes e Comunicações do país desenvolveram um sistema chamado NOPPA(NOPPA, 2004b), com base em um projeto piloto de três anos. “*NOPPA is designed to provide public transport pas-*

senger information and pedestrian guidance for the users of public transport. The system has been designed especially for the needs of the visually impaired, but a sighted user can benefit from it as well”(NOPPA, 2004a).

“NOPPA includes a speech user interface, door-to-door guidance using public transport, the most common GPS functions and several information services”(NOPPA, 2004a).

A arquitetura deste sistema é baseada em um servidor de informações, no qual os celulares dos usuários se conectam através da internet. A transferência de dados no entanto, só é realizada quando o usuário faz uma requisição ao servidor, ou quando existe algum dado atualizado a ser enviado para o usuário. Toda a parte de busca e conversão de texto para voz dos dados é realizada pelo servidor, diminuindo assim, o tráfego de dados.

No entanto, vale ressaltar que o site oficial aparenta estar desatualizado, indicando, nas plataformas disponíveis para o sistema, apenas celulares não mais em produção por parte dos fabricantes.

No estado da Califórnia, pesquisadores desenvolveram um protótipo para um aplicativo que é capaz de prover informações aos usuários sobre ônibus próximos(FLORES; MANDUCHI, 2018). A comunicação é feita através de pontos de acesso wi-fi, instalados em ônibus e pontos de ônibus, o que elimina a necessidade do acesso à internet e da utilização de GPS. O aplicativo tem como objetivo ajudar passageiros com cegueira a chegar em seus destinos.

Os autores de *“Bus Identification System for Visually Impaired Person”*(ALAMY et al., 2012), propuseram um sistema com base em tecnologias como wi-fi e identificação por rádio frequência (RFID). De acordo com os autores, o sistema vai permitir aos usuários embarcarem em ônibus com segurança, através da ajuda de um dispositivo de áudio com interface tátil, que realiza a comunicação entre transmissor e receptor. Ainda de acordo com os autores, os usuários também terão acesso a informações como previsões de saída dos ônibus, dentre outras.

2.1.3 Acessibilidade em aplicativos Móveis

Com a evolução dos celulares, os softwares que os fazem funcionar também evoluíram. E com eles, melhorou-se consideravelmente a qualidade das funcionalidades relacionadas à acessibilidade nesses dispositivos que se tornam cada vez mais indispensáveis para o dia a dia das pessoas.

Dispositivos que tiveram seu início associado a uma completa inacessibilidade por parte de pessoas cegas ou com baixa visão, devido a estes não possuírem nenhuma característica tátil em sua usabilidade, hoje já saem de fábrica com funcionalidades que permitem a qualquer pessoa a sua utilização. Algumas das ferramentas voltadas à acessibilidade que a maioria dos celulares dispõem são:

1. Leitores de tela - softwares que utilizam síntese de voz para converter as informações presentes na tela para áudio;
2. Lupa ou zoom - ferramenta que permite ao usuário ampliar partes específicas da tela, a fim de visualizar melhor as informações contidas na região;
3. Contraste e fonte - funcionalidade que permite inverter o contraste da tela ou editar o tamanho e cor da fonte;
4. Comandos de voz - funcionalidade que permite a utilização do celular através da voz;
5. Áudio para texto - função que permite a conversão de voz para texto;

No entanto, mesmo com o avanço dessas tecnologias, ainda é necessário por parte dos desenvolvedores, a atenção para com os requisitos de acessibilidade no desenvolvimento de aplicativos destinados a dispositivos móveis.

Grandes companhias voltadas ao estabelecimento de padrões internacionais no desenvolvimento e na qualidade de aplicações web (desde sites até aplicativos móveis) abordam em suas recomendações e documentações tópicos voltados à acessibilidade.

A MDN, projeto que tem como missão “prover uma completa, exata e útil documentação para tudo sobre a *open* Web, sendo ele construído, suportado ou não pela Mozilla [...]” possui uma lista de requisitos de acessibilidade para desenvolvedores de aplicativos móveis, requisitos estes, que abrangem aspectos do desenvolvimento como cor e contraste, visibilidade de elementos, usabilidade dentre outros.

Já o W3C, consórcio que tem como missão “levar a *World Wide Web* ao seu potencial máximo, desenvolvendo protocolos e diretrizes que garantam seu crescimento a longo prazo[...]”, recomenda em sua Cartilha de Acessibilidade na Web algumas características semelhantes às indicadas pela MDN, tais como clareza em hiperlinks, áreas de toque com tamanhos que possibilite a facilidade na sua ativação, dentre outros.

Em sua publicação “*Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile*”, é descrito como uma outra lista de padrões de acessibilidade pode ser aplicada no desenvolvimento de conteúdo Web para dispositivos móveis, aplicativos Web para dispositivos móveis, aplicativos nativos e aplicativos híbridos, através da utilização de componentes originalmente destinados à web.

Existe também o “*BBC Mobile Accessibility Prototype*”, conjunto de diretrizes “*based on the requirements of bbc.co.uk content developed for UK audiences and for use with the technology commonly available in the UK*”.

Este último, apesar de não ser destinado primariamente à comunidade internacional, “[...] *can also be referenced by anyone involved in mobile development. The Guidelines are intended for use by anyone involved with the design, development or testing of mobile web and mobile web apps*[...]”.

2.2 Soluções Existentes Voltadas ao Transporte Público

Por todo o mundo, várias empresas se voltam à produção de soluções voltadas à mobilidade urbana. Todos esses serviços oferecem aos usuários uma vasta quantidade de informações sobre opções diversificadas de meios de transporte, contribuindo assim, para uma melhora no seu cotidiano.

Esses aplicativos auxiliam tanto motoristas (disponibilizando informações sobre melhor percurso no trânsito, congestionamentos, possibilitando o compartilhamento de veículos, dentre outras funcionalidades), quanto passageiros (possibilitando o aluguel de veículos, informações sobre rotas e horários de transportes públicos, contratação de serviços de taxi, dentre outros).

No entanto, uma pequena parte destes aplicativos apresenta aspectos voltados à acessibilidade. Geralmente, a maioria das informações presentes se encontram em gráficos, por sua vez, inacessíveis a leitores de tela, e o intervalo entre a atualização de informações importantes para o usuário não aparenta ser muito confiável.

2.3 Soluções IOT Embarcadas

A compreensão da Internet das Coisas ou Internet *Of Things* (IOT) tem sua origem na interconexão de dispositivos na internet, bem como, diretamente entre si. Esses dispositivos que também podem interagir direta e indiretamente com os usuários, quando interconectados, formam grandes redes heterogêneas. Desse modo, pode-se dizer que o conceito de IOT está diretamente ligado a dispositivos associados a sensores "inteligentes" capazes de comunicar-se com outros sensores ou aplicações.

Doukas (DOUKAS, 2012) define a Internet das Coisas como sendo uma rede global que conecta dispositivos através das conexões locais, aplicações web e computação em nuvem. *“It allows devices to communicate with each other, access information on the Internet, store and retrieve data, and interact with users, creating smart, pervasive and always-connected environments”* (DOUKAS, 2012).

“The Internet of Things (IoT) is a system of physical objects that can be discovered, monitored, controlled or interacted with by electronic devices that communicate over various networking interfaces and eventually can be connected to the wider internet. “Things”, in IoT, can refer to a variety of devices such as automobiles with built-in sensors or even biochip transponders on farm animals. These devices collect useful data that can be processed by other systems, by demand or real-time” (MONTEIRO, 2018, p. 14).

Conforme se pode observar na citação acima, a expansão da IOT permitiu com que a variedade de possibilidades de sua aplicação crescesse exponencialmente. Dessa forma,

hoje a internet das coisas está presente nas mais diversas vertentes de aplicações, desde simples projetos de automatização residencial até o "Chão de Fábrica" propriamente dito, *“Examples of potential applications include monitoring vehicles to optimize driving routes, intelligent highways with warning messages and diversions according to climate conditions, for example”* (MONTEIRO, 2018, p. 14).

Na *IOT* os sensores são artefatos fundamentais para a interação entre o mundo real com um sistema *IOT*, tendo em vista que em grande parte dos sistemas embarcados, os dados de entrada são adquiridos por meio da atuação de sensores. Para Silva (SILVA, 2018), “sensores são entidades que tipicamente obtém dados oriundos do ambiente em que estão inseridos[...]”. Desse modo, a utilização correta de sensores é fundamental para o êxito de um projeto *IOT*.

2.3.1 Estado da Arte

Nos últimos anos, tem sido divulgados diversos produtos, estudos, teses, dentre outros materiais, voltados à internet das coisas. Apesar de ser um conceito relativamente novo, já existem várias empresas e produtos que aplicam conceitos e características próprias desta área.

Tatavarthy (SRI, 2016) revisa vários aspectos relacionados ao estado da arte na *IOT*. Com relação à arquitetura, ele descreve que

“[...] The architecture of IoT must be open, scalable, layered, to support heterogeneous future applications. IoT architecture consists of four layers viz. perception layer, network layer, middle ware layer and application layer [...]” (SRI, 2016).

Com relação à protocolos de comunicação, ele sugere que

“The protocols like HTTP and TCP/IP cannot be used for IoT due to the existence of smart devices and other constraints. For machine to machine communication, the protocols MQTT (Message Queue Telemetry Transport) and CoAP (Constraint Application Protocol) are used [...]. MQTT protocol includes the features of publish/subscribe message pattern, messaging transport. CoAP is a web transfer protocol for constrained nodes and constrained networks[...]” (SRI, 2016).

No entanto, para projetos menores, tais protocolos adicionariam um grau de complexidade indesejado aos projetos.

“The development of IoT is the major revolution in IT sector. IoT will have greater impact on daily activities and life style. The design of IoT involves many issues from hardware components to software versions. Data storage, medium of deployment, interoperability, communication mechanisms, data fusion mechanisms, sensor energy, resource and service discovery, bootstrapping and setup security, authentication, access control and accounting, connectivity, mobility and scalability, etc. are some of the several design challenges that exists and need to be addressed[...]” (SRI, 2016).

Ele também aponta que a segurança ainda é a maior vulnerabilidade de aplicações IOT, e que precisam ter a devida atenção. *“Due to the heterogeneity of IoT devices conventional security mechanisms does not directly suit IoT framework. [...]*

The IoT security should be provided at physical objects level, while acquiring information and during information transmission and the requirements such as availability, confidentiality, integrity authentication and authorization, access control, exception handling, resiliency, self organization, anonymity, non-repudiation, freshness need to be provided.” (SRI, 2016).

No que se refere à tecnologias disponíveis para a implementação de projetos IOT, em seu livro *“Building Internet of Things with the Arduino”*, Doukas (DOUKAS, 2012) aponta que:

“Despite the Internet of Things being a relatively new concept, there are already a few open platforms available that enable remote and seamless management and visualization of sensor data: Cosm, Nimbits, and ThingSpeak are just a few examples.[...] And Arduino works with all of them. [...] The Arduino is an incredibly flexible micro-controller and development environment that cannot only be used to control devices, but can also be used to read data from all kinds of sensors. Its simplicity and extensibility, in addition to its great success and adoption by users, has led to the development of a variety of hardware extensions and software libraries that enable wired and wireless communication with the Internet. [...] Arduino is the ideal open hardware platform for experimenting with the world of the Internet of Things [...]” (DOUKAS, 2012).

2.4 Considerações Finais

Ao longo deste capítulo foi feita uma breve abordagem sobre temas relevantes ao projeto como um todo, e que serviram como base para o mesmo. Estes temas foram desde aspectos sociais, até informações e conceitos mais técnicos.

Com relação à IOT, pode-se concluir que se trata de uma área relativamente nova, e em crescente expansão. Como tal, é uma área que apresenta diversos desafios a serem superados, vulnerabilidades a serem corrigidas, padrões a serem estabelecidos, dentre outros aspectos. Porém, a IOT também é uma área com grande impacto e potencial para proporcionar uma melhora significativa no dia a dia das pessoas.

A respeito de projetos voltados ao tema deste trabalho, observa-se que ao redor do mundo, existem várias pesquisas com o mesmo propósito, porém com ideias diferentes no que se refere à sua implementação. Infelizmente, também observa-se que boa parte desses projetos permaneceram apenas no âmbito acadêmico, e poucos produtos com este propósito encontram-se disponíveis à população.

No próximo capítulo, será descrito o processo percorrido para o desenvolvimento do sistema proposto, desde os planejamentos iniciais até os experimentos com uma primeira versão. Também serão abordados alguns problemas encontrados durante a elaboração do aplicativo e do dispositivo embarcado, juntamente com as decisões tomadas perante eles.

3 Desenvolvimento

Com base nos temas abordados no capítulo anterior, este trabalho se propõe a desenvolver um sistema destinado a auxiliar os usuários dos serviços de transporte público, através da utilização de alertas e da disponibilização de informações sobre os veículos que compõe as modalidades de transporte. O sistema se destina primariamente à contribuir com a mobilidade de pessoas com algum tipo de deficiência visual, ou de qualquer pessoa que tenha dificuldade para identificar linhas de ônibus enquanto às espera.

Neste capítulo são descritas as etapas percorridas para a elaboração do sistema proposto. Serão mencionadas ideias iniciais, e alguns problemas encontrados durante as etapas de desenvolvimento e confecção do dispositivo embarcado. Por fim, expõem-se os resultados obtidos com base em experimentos utilizando um primeiro protótipo do sistema.

3.1 Descrição e Projeto do Sistema Proposto

O sistema proposto é constituído por quatro outros sistemas: um aplicativo para dispositivos móveis, destinado aos passageiros; uma plataforma web, destinada aos administradores dos serviços de transporte; uma solução embarcada, a ser instalada nos veículos que integram as frotas de transporte público; e uma aplicação semelhante a um chat em tempo real, destinada a fazer a comunicação entre o aplicativo móvel e o dispositivo embarcado. Para que o sistema funcione corretamente, todas as partes mencionadas necessitam se comunicar entre si. Toda a comunicação é feita através da internet, utilizando quaisquer meios disponíveis para a conexão.

Através da plataforma web, os administradores podem cadastrar informações sobre as opções do serviço de transporte: ônibus, linhas, pontos e endereços. Ela também disponibiliza uma Interface de Programação de Aplicação (*API*), que permite a recuperação dos dados inseridos nela, mediante a links de acesso.

Cada dispositivo embarcado assume uma identificação única, previamente configurada, associada a um ônibus cadastrado através da plataforma web. O dispositivo então passa a ser responsável por se conectar à aplicação em tempo real e atualizar sua posição constantemente, através de informações obtidas por componentes como GPS, acelerômetro, dentre outros.

O aplicativo para dispositivos móveis consulta dados presentes na plataforma web (através da *API*) e também interage com a aplicação em tempo real, recebendo dados da mesma e notificando ao usuário, sempre que houver alguma atualização.

Por fim, a aplicação em tempo real é responsável por receber as informações dos

dispositivos embarcados e enviar essas informações atualizadas para o aplicativo móvel conectado a ela.

Os próximos capítulos descrevem tudo o que foi necessário para a confecção do sistema, desde as primeiras ideias até o sistema construído. Também são feitas as descrições detalhadas das funcionalidades de cada parte mencionada a cima.

3.2 Planejamento Inicial

A maneira como as notificações são utilizadas, juntamente com a qualidade na precisão das informações acessíveis, são alguns dos principais problemas identificados nos aplicativos destinados à mobilidade urbana disponíveis até o momento da escrita deste trabalho. Esses problemas dificultam a utilização eficiente de alguns desses aplicativos por usuários com alguma deficiência visual. Com base nisto, os principais objetivos do sistema são o de proporcionar uma maior precisão nesse tipo de informação, e notificar ao usuário sobre a aproximação dos ônibus que ele deseja de uma maneira mais eficiente e conveniente.

Com relação à maneira como o usuário é avisado sobre os ônibus, o objetivo é que o aplicativo móvel tenha seu funcionamento semelhante ao de um simples detector de metais, ou sensor de colisões presente em alguns veículos. O aplicativo deve sinalizar ao usuário, através de vibração, de sinal sonoro ou de informações textuais, sobre os ônibus desejados à medida que estes se aproximem da localização atual. O intervalo ou a intensidade da vibração ou do sinal sonoro devem se intensificar à medida que a distância entre o ônibus e o usuário diminuam.

Já as informações sobre localizações, pontos, distâncias, dentre outras foram planejadas para serem exibidas através de textos, uma vez que estes são mais acessíveis a leitores de tela.

Para a identificação da distância entre o usuário e o ônibus, a princípio, pensou-se em utilizar o sinal wi-fi, eliminando a necessidade da conexão com a internet. O sinal seria emitido por um módulo transmissor, instalado no ônibus, e monitorado constantemente pelo aplicativo sendo executado em um celular. Com base na intensidade do sinal wi-fi detectado, o aplicativo notificaria ao usuário, utilizando a ideia de notificação descrita anteriormente. No entanto, as desenvolvedoras dos dois sistemas operacionais mais comuns para celulares possuem limitações quanto ao escaneamento por sinais de redes wi-fi.

“[...] The following limitations apply to the frequency of scans using `WifiManager.startScan()`. Android 8.0 and Android 8.1: Each background app can scan one time in a 30-minute period. Android 9 and later: Each foreground app can scan four times in a 2-minute period. This allows for a burst of scans in a short time.

All background apps combined can scan one time in a 30-minute period.[...]” (DEVELOPERS, 2019). Devido a estas limitações, a primeira ideia foi descartada.

Então, optou-se pela criação de um sistema a ser executado em um servidor. Esse sistema atua realizando a comunicação entre o módulo presente no ônibus e o aplicativo móvel. Nesse modelo, o transmissor wi-fi do módulo foi substituído por um receptor de GPS e por um módulo de conexão GSM/GPRS, o que permite, respectivamente, a obtenção da localização do módulo e a conexão com a internet.

A seguir, são detalhadas cada uma das partes desse segundo modelo escolhido.

3.3 Desenvolvimento do Aplicativo Móvel

Como mencionado na introdução deste capítulo, a principal interface entre o sistema proposto e o usuário dos serviços de transporte público é um aplicativo para dispositivos móveis. Essa escolha se deu devido à comodidade que os dispositivos móveis proporcionam, e ao fato deles estarem se fazendo cada vez mais presentes no cotidiano das pessoas.

Devido ao fato de, a princípio, o sistema ser destinado a auxiliar usuários que possuam alguma deficiência visual, optou-se pela apresentação das informações de uma maneira mais textual. Informações como localização atual, localização de ônibus, distâncias, dentre outras estão presentes primariamente através de textos, ao invés de imagens. Essa escolha se deve ao fato da conversão de informações realizadas pelos leitores de tela atualmente presentes nos dispositivos móveis serem mais eficientes, se comparadas informações textuais com informações gráficas.

Para o desenvolvimento do aplicativo móvel, foi escolhido o *framework Ionic* na versão 4. Trata-se de um conjunto de ferramentas para o desenvolvimento de aplicações móveis híbridas, com base em tecnologias para o desenvolvimento web. Ele é constituído por tecnologias amplamente utilizadas no desenvolvimento web, tais como Linguagem de Marcação de Hipertexto (HTML), Folha de Estilos em Cascada (CSS) e Java Script. Ele também é capaz de ser integrado a outras bibliotecas de desenvolvimento, como Bootstrap e Angular.

De acordo com a empresa responsável pelo seu desenvolvimento, o framework é voltado à experiência de usuário, focado em aspectos como controles, navegação, gestos, dentre outros.

O framework também é multi-plataforma, ou seja, ele gera aplicativos que funcionam nos principais sistemas operacionais presentes nos dispositivos móveis atuais, além de ser possível a sua utilização como um aplicativo web. De acordo com a desenvolvedora, *“Ionic Framework is built with simplicity in mind, so that creating Ionic apps is enjoyable, easy to learn, and accessible to just about anyone with web development skills”*.

Para o levantamento de requisitos do aplicativo, seguiu-se as seguintes etapas. Primeiro, listou-se as ações mais comuns e necessárias que uma pessoa tomaria em seu cotidiano para completar uma viagem de ônibus. Então, fez-se uma pesquisa e listagem de funcionalidades já presentes em outros aplicativos disponíveis no mercado. A seguir,

com base nessas duas listas, separou-se as funcionalidades mínimas necessárias para que o aplicativo cumprisse seu objetivo. Por fim, as funções foram moldadas com base em normas e boas práticas de desenvolvimento. As tabelas disponíveis no apêndice deste trabalho exibem os requisitos resultantes das etapas descritas anteriormente.

Com base nesses requisitos, selecionou-se os componentes do *framework* de desenvolvimento que possibilitaram o cumprimento das funcionalidades desejadas. Nas subseções a seguir, são descritos alguns aspectos sobre os componentes utilizados no desenvolvimento das funcionalidades do aplicativo.

3.3.1 Elaboração das Telas

A usabilidade do aplicativo foi pensada para ser bastante simples. Sendo assim, ele possui poucas telas. Devido ao *framework*, essas telas se assemelham a páginas HTML em sua elaboração, com o acréscimo de componentes que utilizam recursos específicos do celular.

As telas utilizam o sistema de layout baseado em grade disponível no *framework*. Ou seja, todos os componentes presentes são organizados com base em linhas e colunas, as quais o *framework* se encarrega de alinhar e posicionar de acordo com o tamanho da tela do dispositivo no qual o aplicativo estará sendo executado.

Na tela inicial e também principal, é onde se concentram as informações mais relevantes para a utilização do aplicativo. Nela estão presentes botões que permitem ao usuário pesquisar e selecionar um ponto de destino, e obter informações sobre algum ônibus ou ponto de ônibus, juntamente com informações como o endereço atual do usuário, o ponto mais próximo de sua localidade, a lista dos ônibus que possuem em seu roteiro um ponto próximo a ele e o ponto de destino selecionado atualmente (caso ele selecione um).

A tela de busca apresenta em sua parte superior um campo para a entrada de texto, juntamente com botões selecionáveis, que permitem ao usuário procurar pontos de ônibus com base em uma característica específica. Na parte inferior dessa mesma tela, encontra-se uma lista, a ser preenchida com os resultados da busca que o usuário realizar. Os itens dessa lista são clicáveis, e ao serem acionados, selecionam o ponto específico, retornando para a tela principal em seguida.

Figura 1 – Imagem mostrando uma das telas do aplicativo para dispositivos móveis

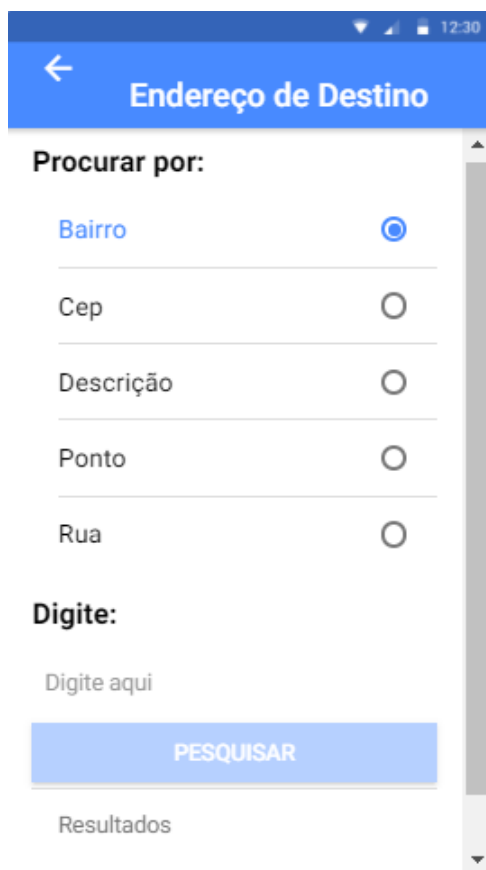
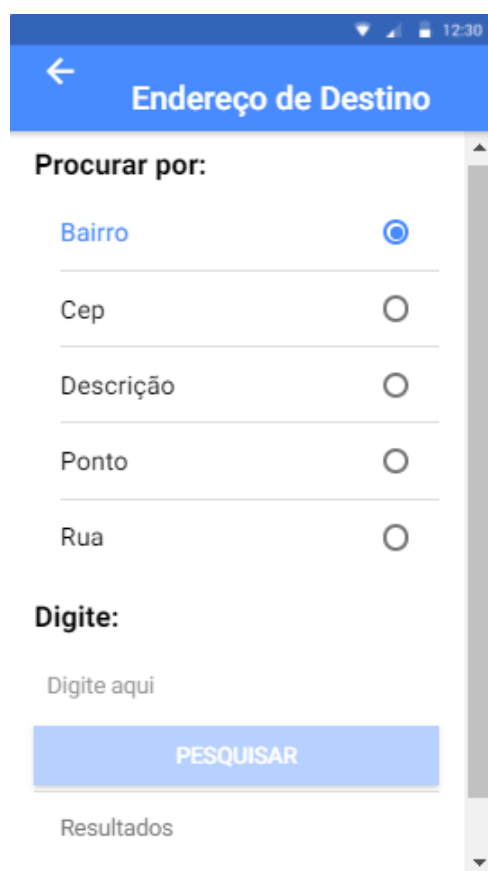


Figura 2 – Imagem mostrando uma das telas do aplicativo para dispositivos móveis



As imagens a cima mostram cada uma, uma tela diferente do aplicativo destinado ao usuário. Em uma é possível ver os campos necessários para a realização de buscas, e em outra, vários campos contendo informações como "destino atual", "ponto mais próximo", dentre outros.

3.3.2 Posição e Cálculo de Distância

Para obter a localização do usuário, o aplicativo utiliza três componentes existentes no *framework*: o *Native Geocoder*, o *Background Geolocation* e o *Geolocation*. Esses componentes fazem parte da biblioteca *Ionic Native*, um conjunto de *plugins* e integrações, que possibilitam o aplicativo híbrido executar funções nativas do dispositivo.

O *Geolocation* proporciona a obtenção de informações sobre a localização do dispositivo, tais como latitude e longitude. Essas informações são provenientes de fontes como o sistema de GPS, e deduções através de sinais de rede, como endereço IP, wi-fi, tecnologias utilizadas por operadoras de telefonia, como GSM/CDMA, dentre outras. Já o *Background Geolocation* permite a obtenção da localização atual enquanto o aplicativo se encontra em primeiro plano ou em plano de fundo, com algumas opções que, de acordo com o desenvolvedor, proporcionam uma economia de bateria.

Essas informações são obtidas em formato numérico, o que não as torna muito significativas para um usuário comum, uma vez que essas podem ser mais difíceis de serem interpretadas. Então, utiliza-se o *Native Geocoder*, que permite a conversão dos valores obtidos pelos componentes em notações textuais. Isso possibilita com que o usuário visualize o endereço de uma maneira mais comum, constituído pelo nome da rua, bairro, cidade e estado.

Referente ao cálculo de distância, foi elaborada uma função baseada na equação conhecida como fórmula de Haversine, uma equação utilizada na navegação, que tem como embasamento matemático a Lei dos Cossenos, e que utiliza o valor aproximado de 6371 km para a curvatura da Terra, devido ao seu formato elíptico.

Esta função foi utilizada para estimar a distância em metros entre a posição do usuário e a posição do ponto de ônibus mais próximo. Sendo que, primeiramente, o aplicativo solicita à *api* uma lista de pontos, com base na localização do usuário, então, o cálculo é feito, comparando as coordenadas do usuário com cada uma das coordenadas dos pontos retornados pela *api* (se ela retornar algum).

É importante ressaltar que, normalmente, coordenadas geográficas são expressas em formato sexagesimal (composto por graus, minutos e segundos), e que para a realização do cálculo, converte-se essa medida para sua correspondente em formato decimal. Essa conversão já é realizada pelo componente utilizado no framework.

3.3.3 Utilização de *Api's* e Integração com o Servidor

Como mencionado em vários trechos anteriormente, o aplicativo utiliza dados de uma aplicação externa. Para essa interação, utiliza-se módulos do *framework* que permitem a elaboração, o envio e o tratamento de respostas para requisições http, encaminhadas a um servidor.

Estas requisições seguem o padrão *REST*, definido por Roy Thomas Fielding (FIELDING, 2000) em sua tese de doutorado, e utilizado por toda a web. O padrão “*REST provides a set of architectural constraints that, when applied as a whole, emphasizes scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems*” (FIELDING, 2000).

Para a transmissão e recepção dos dados utilizados no cálculo da estimativa das distâncias, o aplicativo precisa estar conectado em tempo real com a aplicação sendo executada no servidor. Para isto, utilizou-se uma biblioteca que implementa web *sockets*, chamada de *Socket.IO*.

3.4 Desenvolvimento do Sistema a Rodar no Servidor

Para que o aplicativo descrito no capítulo anterior funcione corretamente, ele precisa interagir com uma aplicação externa. Essa aplicação é responsável por gerenciar os dados disponíveis a respeito dos ônibus, pontos e endereços. No apêndice deste trabalho, também encontra-se a lista de requisitos funcionais e não funcionais levantados para a aplicação.

Como indicado nas tabelas e listas de requisitos funcionais e não funcionais, os objetos modelados para os quais as informações do sistema são destinadas, são os ônibus e os pontos de ônibus. Os pontos, por sua vez possuem endereços associados a eles. Cada endereço possui um par de coordenadas geográficas, obtidas com base em sistemas externos de geolocalização. O cadastro dessas informações no sistema é feito através do acesso web.

Devido à natureza e às especificações do sistema, optou-se por dividir o desenvolvimento do mesmo em duas partes. Uma parte dedicada ao gerenciamento e disponibilização das informações, e outra parte responsável pela interação em tempo real entre os dispositivos embarcados nos ônibus e o aplicativo móvel.

Para a parte de gerenciamento dos dados relacionados aos ônibus, pontos e endereços, utilizou-se o *framework* de desenvolvimento Grails, em sua versão 33.9. O *framework* é voltado ao desenvolvimento web, utilizando a linguagem de programação Groovy. Ele também tem foco em desempenho e alta produtividade, e é capaz de ser executado a partir da Jvm (*Java Virtual Machine*).

As tecnologias que compõem o framework são amplamente difundidas, testadas e bem avaliadas no mercado. Dentre elas, se incluem o sistema avançado de automatização de compilação e gerenciamento de dependências (Gradle), o *framework* Java baseado na inversão de controle e injeção de dependências (Spring), o *framework* de mapeamento entre objetos e banco de dados (Hibernate), o servidor de aplicação Tomcat, o *framework* para o design de interfaces (Bootstrap), dentre outros *frameworks* e bibliotecas. Ele também possui funcionalidades que facilitam a criação de *apis rest* (*Representational State Transfer*).

O banco de dados utilizado para a persistência das informações escolhido foi o PostgreSQL. Isso se deu ao fato de o banco de dados em questão estar entre os sistemas de gerenciamento de banco de dados mais utilizados no mercado, além de sua fácil integração com o *framework* Grails, através da utilização de uma biblioteca própria para isso.

De acordo com as tabelas 3 e 4, o sistema permite o cadastro de endereços (composto por campos como rua, bairro, cep, cidade, dentre outros), o cadastro de pontos de ônibus (composto por um nome, um par de coordenadas geográficas e uma ligação com um endereço), e o cadastro de ônibus (contendo informações como um código que identifique a linha e todos os pontos de seu itinerário).

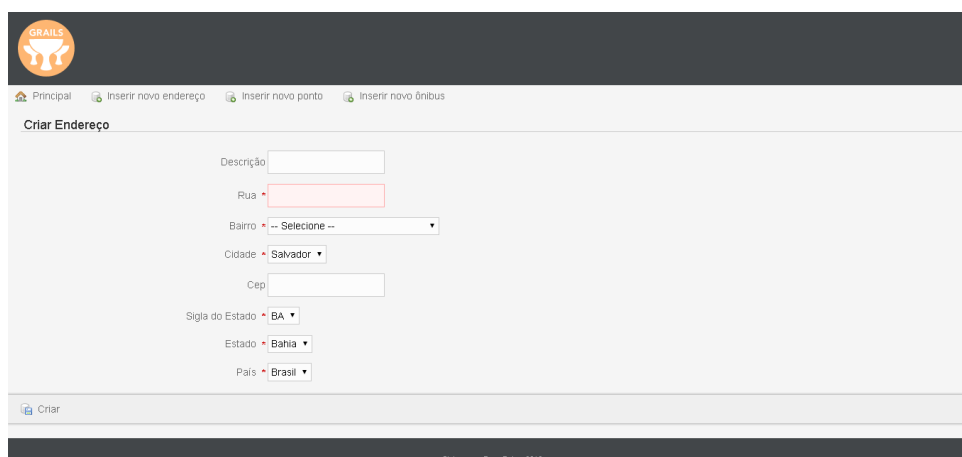
As telas foram modeladas utilizando o *framework* Bootstrap, juntamente com a tecno-

logia de renderização de visualizações Groovy *Server Pages* (GSP). Um GSP é geralmente a junção entre marcações HTML e *tags* próprias do GSP, o que auxiliam na renderização da camada *View* da arquitetura *Model View Controller* (MVC).

Um outro aspecto relevante sobre o cadastro das informações se refere à geolocalização dos pontos de ônibus. Para esse cadastro, foi utilizado um serviço de mapas chamado MapBox. O MapBox é uma plataforma que provê informações geográficas, e permite a extração de dados como latitude e longitude através da renderização de mapas.

No entanto, como a parte gráfica não é acessível a softwares de leitura de telas, duas outras funções foram implementadas como alternativa: a geocodificação e a geocodificação reversa. Esses processos permitem a obtenção de dados como latitude e longitude a partir de endereços escritos por extenso, e o processo contrário, verificar um endereço através de um par de coordenadas geográficas.

Figura 3 – Imagem mostrando a tela para o cadastro de endereços na aplicação web do projeto



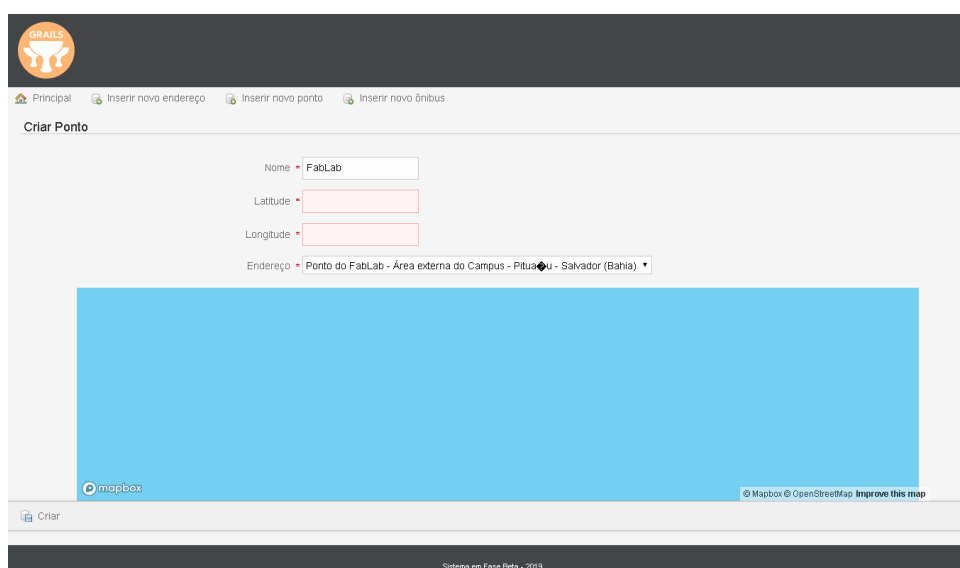
A imagem mostra a interface de usuário para o cadastro de endereços em uma aplicação web. No topo, há um cabeçalho com o logotipo do Grails e um menu de navegação com opções: Principal, Inserir novo endereço, Inserir novo ponto e Inserir novo ônibus. O formulário principal, intitulado "Criar Endereço", contém os seguintes campos:

- Descrição: campo de texto.
- Rua: campo de texto com um ícone de localização.
- Bairro: menu suspenso com o texto "-- Seleccione --".
- Cidade: menu suspenso com o texto "Salvador".
- Cep: campo de texto.
- Sigla do Estado: menu suspenso com o texto "BA".
- Estado: menu suspenso com o texto "Bahia".
- País: menu suspenso com o texto "Brasil".

Na base do formulário, há um botão "Criar". No rodapé da página, há o texto "Sistema em Foco Beta - 2019".

Na imagem anterior, é possível ver uma página da internet. A página em questão se refere a uma das funcionalidades do sistema web proposto. São vistos vários campos para a inserção de texto, todos eles etiquetados e ordenados de cima para baixo.

Figura 4 – Imagem mostrando uma das telas da aplicação web do projeto



Na imagem anterior, é possível ver uma página da internet. A página em questão se refere a uma das funcionalidades do sistema web proposto. São vistos vários campos para a inserção de texto, todos eles etiquetados e ordenados de cima para baixo. Também se vê uma área destinada à exibição gráfica de um mapa.

Ainda seguindo os requisitos, o sistema disponibiliza uma *api rest*, que permite a obtenção dos dados cadastrados, sendo estes apresentados utilizando a notação de objetos JavaScript (JavaScript *Object Notation (JSON)*). Esta estrutura é mais leve se comparada a outras como o XML, e por isso foi escolhida para a transferência de dados entre o aplicativo móvel e a aplicação.

Toda a interação com as informações presentes no sistema é feita mediante um processo simples de autorização. Para a utilização da interface web, o processo funciona através da validação de *login* e senha, e a subsequente criação de dados de seção de usuário no navegador. Já para a utilização da *api*, o processo de autenticação utiliza o padrão *JSON Web Token (JWT)*.

Esse padrão define uma forma compacta e independente para transmitir com segurança informações entre duas partes através de um objeto JSON. Essas informações podem ser verificadas devido a elas serem assinadas digitalmente. Os objetos JWT podem ser assinados usando uma senha (com o algoritmo HMAC) ou um par de chaves pública / privada (usando algoritmos como RSA ou ECDSA).

Uma outra funcionalidade da aplicação, é a de gerar códigos de autenticação válidos para que os usuários do aplicativo móvel possam monitorar ônibus específicos. Com base nesses códigos, o sistema também é capaz de gerenciar sobre quais ônibus e para quais usuários ele deve enviar atualizações.

3.4.1 Comunicação com Clientes em Tempo Real

Um dos requisitos essenciais para o sistema é a comunicação em tempo real entre todas as partes envolvidas no processo. Para isso, foi desenvolvida uma pequena aplicação em paralelo utilizando a linguagem de programação Python.

Para a execução do sistema, utilizou-se a biblioteca AIOHTTP. Esta biblioteca é responsável pela implementação de um cliente e servidor Http, com a característica de ser assíncrono, para Python.

Esta aplicação é responsável por receber os dados de GPS transmitidos pelos dispositivos embarcados (abordados mais detalhadamente no próximo capítulo), e enviar para os usuários que indiquem interesse em ser atualizados a respeito de algum ônibus. Para essa finalidade, a aplicação utiliza a implementação dos padrões *WebSocket*. O *WebSocket* trata-se de uma tecnologia que torna possível a abertura de canais de comunicação interativos (através do modelo de rede *full-duplex*) entre o navegador do usuário (ou qualquer outro dispositivo que implemente as especificações necessárias) e um servidor, também implementando os padrões estabelecidos.

Este tipo de comunicação permite que os dois pontos envolvidos na comunicação possam transmitir dados simultaneamente em ambos os sentidos, sem que um dependa do outro. Essa característica remove a necessidade de o aplicativo móvel precisar realizar diversas requisições Http para se manter informado sobre a posição dos ônibus selecionados.

Para gerenciar sobre quais ônibus quais usuários deverão ser informados, optou-se pela utilização da biblioteca Socket.IO. De acordo com a documentação oficial:

“Socket.IO is NOT a WebSocket implementation. Although Socket.IO indeed uses WebSocket as a transport when possible, it adds some meta-data to each packet: the packet type, the namespace and the ack id when a message acknowledgement is needed. That is why a WebSocket client will not be able to successfully connect to a Socket.IO server, and a Socket.IO client will not be able to connect to a WebSocket server either [...]” (SOCKET.IO, 2019).

Esta biblioteca dispõe de recursos que dentre outras funcionalidades, permitem a fácil criação de grupos de destinatários para os quais mensagens específicas devem ser enviadas. Com base nisso, sempre que a posição atual de um ônibus registrado no servidor for atualizada, os novos dados serão enviados para os usuários específicos que desejam se manter informados sobre aquele ônibus.

O recebimento dos dados de GPS enviados pelo dispositivo embarcado se dá através de uma requisição Http. No momento da requisição, é feita também uma busca por possíveis usuários que desejem embarcar no veículo. Caso existam, essa informação é enviada como resposta ao dispositivo.

3.4.2 Cálculo de Distância

A aplicação também realiza estimativa de distâncias com base em pares de coordenadas geográficas. Esses cálculos são realizados nas funções de pesquisa por pontos de ônibus, com o objetivo de retornar todos os pontos mais próximos da posição informada pelo usuário.

Esse cálculo é feito seguindo os seguintes passos. Primeiro, com base na latitude e longitude informadas pelo usuário, obtêm-se pares de coordenadas representando os delimitadores de uma área imaginária. A seguir, realiza-se uma busca no banco de dados, utilizando como parâmetros os pares de coordenadas obtidas na etapa anterior. A busca retorna todos os pontos que estejam dentro da área definida, e esse resultado é enviado ao aplicativo móvel.

A seguir, segue-se o algoritmo, desenvolvido na linguagem Groovy, que foi utilizado para definir a área de busca, tendo como centro as coordenadas do usuário.

```
1 def minMaxCoordenadas(double lon , double lat , double dist){
2     double R = 6371
3     double radius = dist
4     double longMin = lon - Math.toDegrees(radius/R/Math.cos(Math.
5     toRadians(lat)))
6     double longMax = lon + Math.toDegrees(radius/R/Math.cos(Math.
7     toRadians(lat)))
8     double latMax = lat + Math.toDegrees(radius/R)
9     double latMin = lat - Math.toDegrees(radius/R)
10    def values = [:]
11    values.latmin=latMin
12    values.latmax=latMax
13    values.lonmin=longMin
14    values.lonmax=longMax
15    return values
16 }
```

3.5 Desenvolvimento da Aplicação *IOT* Embarcada

Como mencionado nas seções anteriores, a terceira parte da solução proposta é um dispositivo embarcado. Esse dispositivo é responsável por manter a localização do ônibus atualizada junto a base de dados do servidor, e por alertar ao motorista sobre usuários em pontos de ônibus à frente que desejem embarcar no veículo.

Para a confecção desse dispositivo, utilizou-se a plataforma eletrônica de desenvolvimento de código aberto para *software* e *hardware*, chamada Arduino. Optou-se por esta plataforma devido à variedade de componentes disponíveis, e a facilidade de sua integração e utilização para prototipagem.

Com o objetivo de cumprir as funções pensadas para o dispositivo, utilizou-se os seguintes módulos e componentes do Arduino:

A placa responsável pelo processamento foi o Arduino Mega2560; também incluiu-se um Módulo Acelerômetro de 3 Eixos, um módulo GSM GPRS, um módulo receptor de GPS, além de peças comuns na construção de protótipos, como um Buzzer ativo de 5 volts e luzes de leds.

Nas seções a seguir, são descritos como e para quê cada componente foi utilizado no projeto.

3.5.1 Obtenção de Posição e Velocidade

Para obter a localização atual do dispositivo, utilizou-se o módulo GPS. De acordo com a fabricante, o módulo apresenta baixo consumo de energia, e é indicado e certificado para integração com veículos.

Ainda de acordo com a fabricante, no processo de obtenção da posição, o módulo é capaz de interagir com uma rede de aproximadamente 50 satélites de GPS. Essa interação varia bastante, e a qualidade dos sinais e quantidade de satélites acessíveis ao dispositivo são influenciados por fatores de ambiente diversos, tais como clima e/ou construções próximas.

No Arduino, foi utilizada uma biblioteca destinada ao tratamento das informações obtidas pelo módulo. Ela converte, por exemplo, dados de latitude e longitude para valores numéricos, o que facilita sua manipulação para a realização dos cálculos de distância.

Devido ao aplicativo para dispositivos móveis e à aplicação no servidor por padrão trabalharem com dados numéricos, optou-se por converter as informações geradas no Arduino para este tipo de variáveis.

Já para verificar se o dispositivo encontra-se parado ou em movimento, utilizou-se o módulo de acelerômetro. Com base nos dados informados por Este módulo, é possível estimar uma aproximação no valor da aceleração existente no dispositivo.

Então, com base nos valores obtidos, atualiza-se as informações referentes à posição atual. Isto ocorre sempre que é detectado que o dispositivo encontra-se em movimento.

3.5.2 Conectividade via GPRS

Para que o dispositivo cumpra sua função, é necessário que ele se conecte com a aplicação sendo executada no servidor. Essa conexão se dá através do módulo SIM800L. Esse módulo tem seu funcionamento semelhante ao de um celular. Tendo conectado a ele um chip de telefonia, ele consegue realizar diversas funções, tais como realizar e receber ligações, enviar e receber mensagens de texto, e acessar a internet.

O acesso a internet se dá através do GPRS (*General Packet Radio Services*). Esta tecnologia se refere a sistema de transferência de dados móveis por pacotes, compatíveis

globalmente com a maioria das operadoras de telefonia. A tecnologia é projetada para ter alta velocidade, e pode ser utilizada para provê conectividade a dispositivos além de celulares. Para o funcionamento do módulo, é necessário que se utilize um *chip* com plano de telefonia que inclua dados móveis.

A execução de instruções no módulo foi feita através de comandos AT. Esses comandos se referem a um conjunto de opções aceitas e interpretadas pelo módulo. Esses comandos permitem que o código escrito em Arduino possa executar instruções no módulo SIM800L, e trabalhar com os dados retornados por ele.

O módulo também possui a funcionalidade de obter a localização atual, semelhante ao GPS. No entanto, ele utiliza outras técnicas e fontes para extrair essa informação, tais como sinal de rede, endereço IP, dentre outras. Por essas técnicas atualmente serem menos precisas do que o GPS, não se trabalhou com esses dados na estimativa da distância pela aplicação.

3.6 Considerações Finais

Este capítulo apresentou as etapas percorridas ao longo do desenvolvimento do aplicativo para dispositivos móveis e da aplicação web, juntamente com o que foi necessário para a confecção do dispositivo IOT embarcado. Também descreveu-se como foram planejados e implementados os cálculos para a estimativa de distâncias, tanto por parte do aplicativo quanto por parte da aplicação web.

Pôde-se concluir que o desenvolvimento do aplicativo, utilizando o framework, como o esperado, foi relativamente descomplicado, uma vez que a codificação para a composição dos componentes visuais e da parte lógica do aplicativo se dá através de tecnologias web. No que se refere à confecção do dispositivo IOT embarcado, a conectividade entre os módulos escolhidos e a plataforma Arduino decorreu sem imprevistos, e todos os módulos responderam conforme o esperado.

A seguir, são descritos os experimentos que foram planejados e realizados envolvendo todas as partes do sistema proposto. Também são expostos os resultados obtidos, juntamente com alguns problemas encontrados.

4 Experimentos e Resultados

Após a conclusão das partes descritas nos capítulos anteriores, realizou-se alguns experimentos controlados para avaliar a eficiência do conjunto da solução proposta. Buscou-se, através destes experimentos, avaliar o nível e a qualidade da interação entre as partes desenvolvidas, juntamente com a precisão dos dados que elas produziram.

Dentre as técnicas e categorias de experimentos mais comuns e utilizadas na área da Engenharia de Software, foram aplicadas duas dessas modalidades: o In Virtuo, que refere-se a experimentos com base em simulações do mundo real, e o In Vitro, que se refere a experimentos aplicados em laboratório, ou seja, em ambientes controlados.

Os experimentos In Vitro decorreram no campus de Pituvaçu, na Universidade Católica do Salvador, devido à possibilidade de simular o tráfego esperado para a atuação da solução proposta na área externa do campus. Já os experimentos In Virtuo foram realizados utilizando a própria estrutura montada no sistema de computação em nuvem da Google.

Foram planejados experimentos In Vivo, que deveriam ocorrer em trechos aleatórios pertencentes à itinerários de linhas de ônibus em circulação na cidade de Salvador, Bahia. No entanto, devido a instabilidade no dispositivo embarcado, os experimentos In Vivo não puderam ser realizados.

A princípio, já era esperada alguma imprecisão no que se refere aos dados obtidos pelos receptores de GPS, uma vez que o sinal destes são propensos a sofrerem interferências. Estas interferências podem “ser divididas em interferências intencionais e não-intencionais”(SOUSA, 2005).

“As interferências não intencionais são atribuídas às fontes de erros sistêmicos e aos problemas causados por emissões espúrias¹ oriundas de outros serviços, tais como TV via satélite, estações de rádio FM, radioamadores, entre outros.

As interferências intencionais são aquelas que mais preocupam a comunidade usuária. São provenientes de transmissores de diversos tipos, projetados exclusivamente com o propósito de causar danos ao sistema, podendo causar o bloqueio dos canais, impossibilitando a aquisição dos códigos, ou provocar alteração dos mesmos com o objetivo de afetar a precisão do posicionamento.”(SOUSA, 2005).

Além disto, fatores como a qualidade do receptor, erro entre o relógio do receptor e dos satélites, quantidade e nível dos sinais recebidos, quantidade de satélites "visíveis" pelo aparelho, fatores causados pelas camadas da atmosfera, multicaminhamento (reflexão / deslocamento) do sinal devido a quaisquer obstáculos presentes, dentre outros fatores afetam diretamente a qualidade das informações.

¹ Não genuíno; suposto, hipotético; falsificado, alterado, adulterado.

4.1 Descrição do Experimento

Para a realização dos experimentos, tanto In Virtuo quanto In Vitro, instalou-se a solução embarcada em veículos particulares. Para a alimentação elétrica do dispositivo, utilizou-se um inversor portátil para automóveis, capaz de proporcionar uma corrente de até 2,5 amperes. A utilização de um inversor foi necessária devido ao consumo de um dos módulos do dispositivo, que, na falta de uma quantidade mínima de energia, apresenta funcionamento instável, o que compromete sua operação.

O aplicativo para dispositivos móveis foi previamente instalado em um celular com sistema operacional Android, sendo necessário fornecer permissões de uso, tais como acesso à localização, vibração, controle de áudio, dentre algumas outras durante sua instalação, para que o mesmo funcionasse de acordo com o esperado.

Como especificado no decorrer dos capítulos anteriores, todas as partes da solução proposta necessitam de conexão à internet, de modo a interagirem entre si. Para isso, tanto o celular executando o aplicativo quanto a aplicação embarcada possuíam acesso à internet através de cartões pré-pagos de telefonia móvel, devidamente habilitados à conexão 3G.

A aplicação web responsável por intermediar a comunicação entre o aplicativo e o dispositivo embarcado foi hospedada no sistema de máquinas virtuais em nuvem da Google. A máquina virtual criada para os experimentos possuía especificações básicas. O sistema operacional escolhido foi o Ubuntu, versão 14.4, com 3 GB de memória Ram.

Na máquina foram instaladas as ferramentas necessárias a fim de possibilitar a execução dos programas. Para a execução da aplicação feita utilizando o *framework* Grails, instalou-se o Java *Development Kit*, versão 8.1.0. Para os experimentos, não utilizou-se um servidor de aplicação dedicado, comumente empregado em ambientes de produção, como por exemplo o Tomcat, devido a uma funcionalidade do framework, que permite exportar um pacote pronto para a execução "*stand alone*".

Já para a execução da aplicação em tempo real, instalou-se o Python, em sua versão 3.6. Juntamente com a instalação do Python, instalou-se alguns pacotes dos quais a aplicação depende para executar suas funcionalidades, tais como o pacote GeoPy, para manipular dados de geo localização, o servidor assíncrono Http AioHttp, e a implementação da biblioteca Socket.io para Python.

Com todas as dependências estruturadas, o experimento In Virtuo foi conduzido da seguinte maneira. Primeiro, selecionou-se três localizações dentro do campus da universidade. Obteve-se então, as coordenadas geográficas dos pontos escolhidos, através de serviços de mapas disponíveis na internet, e efetuou-se o cadastro dessas informações junto ao sistema. Esses dados também foram utilizados para posteriores experimentos In Vitro.

Nos experimentos In Virtuo, simulou-se o comportamento esperado tanto pelo dispo-

sitivo embarcado, quanto pelo aplicativo para celulares. Foi pressuposto que um usuário ficaria parado em um determinado ponto, e o ônibus estaria se movimentando. Durante a movimentação do ônibus, ele se aproximaria e passaria pelo ponto onde se encontrava o usuário. Ambos estariam constantemente enviando suas respectivas posições à aplicação no servidor, posições estas compostas por latitude e longitude.

Simulou-se o envio das localizações tanto por parte do usuário como por parte do ônibus. Com base nessas informações, compostas por latitude, longitude e um número de identificação único, a aplicação sendo executada no servidor deveria perceber quando o ônibus se aproximava do usuário, e neste momento, deveria retornar para o dispositivo embarcado, como resposta a uma requisição *Get*, informações que indicassem que ele se aproximava de um usuário. O dispositivo, por sua vez, ao receber este tipo de informação, deveria sinalizá-la ao condutor.

Com relação ao dispositivo embarcado, a princípio, utilizou-se funções nativas de acesso à internet através do protocolo HTTP e da transmissão de dados via GPRS padrões do módulo. Seguindo o padrão *REST*, enviou-se dados como a latitude e longitude, extraídas do módulo GPS, através de protocolos de requisições *HTTP/Post*.

O conteúdo da requisição *Post* foi composto por um objeto JSON. O objetivo da utilização dessa representação para as informações foi o de enviar uma quantidade pequena de dados, a fim de agilizar o envio e o processamento das requisições, tanto por parte do servidor, e principalmente por parte do módulo GPRS.

No entanto, percebeu-se uma demora na execução dos comandos necessários pelo módulo para o envio das requisições *Post*. Isso se deu devido ao módulo precisar seguir uma sequência de passos. Primeiro é preciso definir o tipo de requisição, e esperar uma confirmação do módulo. Depois, é necessário definir que se deseja anexar os dados do corpo da requisição juntamente com o tempo que o módulo deverá aceitar o envio desses dados, e novamente, esperar pela confirmação do módulo. Então, é preciso enviar os dados que formarão o corpo da requisição, e esperar uma confirmação do módulo. Por fim, envia-se ao módulo o comando responsável por realizar a requisição em si.

Não se pôde especificar um intervalo muito pequeno para preenchimento do corpo da requisição, pois se ocorresse algum atraso no envio dos dados, o módulo recusaria quaisquer dados seguintes, sendo necessário executar novamente os comandos citados anteriormente. Igualmente, também não se pôde especificar um intervalo muito amplo para o preenchimento do corpo da requisição, pois isto ocasionaria em um espaço de tempo entre as requisições ao servidor muito grande, o que ocasionaria a diminuição na precisão das informações que chegariam ao usuário.

Com base nisso, optou-se pela utilização de requisições *HTTP/Get*, migrando os dados de latitude e longitude para parâmetros na própria *URL*. Isto eliminou a necessidade da utilização dos comandos para o preenchimento do corpo descritos anteriormente, o que, por sua vez, proporcionou um pequeno aumento no tempo necessário para o envio das

requisições, de cerca de 1 segundo e meio.

No entanto, durante os experimentos In Virtuo, percebeu-se uma instabilidade no funcionamento do módulo responsável pela conexão com a internet. O módulo em questão apresentou superaquecimento, constantes travamentos, reinicializações e demora na resposta aos comandos necessários para a execução de suas funções.

Primeiramente, investigou-se as possíveis causas para o problema. Algumas delas, envolveram o fornecimento da alimentação através de corrente elétrica com amperagem maior ou igual a 2 amperes. Tentou-se a utilização de um inversor, a ligação de baterias, e até a conexão de mais de uma fonte de alimentação em paralelo. As correntes elétricas, antes de serem aplicadas direto no módulo foram previamente medidas, mas o módulo continuou instável.

Para o problema do superaquecimento, tentou-se ligar o módulo fornecendo uma refrigeração. Apesar de, com refrigeração, o módulo não aquecer, ele continuou apresentando travamentos e reinicializações.

Tentou-se também, mudar o fornecimento da voltagem enviada do Arduino para o módulo. Experimentou-se com 5 volts e com 3,3 volts. No entanto, isto também não resolveu os problemas identificados.

Devido a esta instabilidade, optou-se por realizar os experimentos In Vitro substituindo o dispositivo embarcado por um outro aplicativo para celulares. O aplicativo em questão foi elaborado tendo como funcionalidades as mesmas inicialmente pensadas para o dispositivo embarcado, e utilizou-se o mesmo framework de desenvolvimento empregado na elaboração do aplicativo destinado ao usuário.

Com os dois aplicativos devidamente instalados em celulares diferentes e funcionando corretamente, realizou-se os experimentos In Vitro. Estes, por sua vez, foram semelhantes aos experimentos In Virtuo, onde, uma pessoa (com o celular executando o aplicativo destinado ao usuário) se posicionou em um dos pontos escolhidos na área externa do campus, e um voluntário (com um celular executando o aplicativo substituto ao dispositivo embarcado), utilizando um veículo particular, percorreu a área do campus da universidade.

4.2 Resultados dos Experimentos In Vitro

Em um dos experimentos, o veículo se deslocou até uma distância aproximada de 177 metros, com velocidade constante entre 15 a 20 km/h. Não foram realizadas frenagens bruscas durante o deslocamento, e o aplicativo do usuário registrou a distância entre ele e o carro diminuir e aproximar em intervalos de 3 em 3 metros. Acredita-se que isto se deu devido à velocidade constante do veículo. No entanto, com relação ao resultado final exibido, quando o veículo parou próximo ao usuário, simulando a parada de um ônibus, a distância registrada no aplicativo era de 18 metros, quando deveria marcar no máximo 5 metros.

Em outro experimento, utilizando pontos diferentes, o carro se afastou a uma distância de aproximadamente 230 metros, mantendo uma velocidade constante entre aproximadamente 20 a 25 km/h. Também não foram realizadas frenagens bruscas, e as distâncias exibidas para o usuário não seguiram um intervalo fixo. Desta vez, a distância exibida para o usuário quando o veículo encontrava-se parado próximo a ele era de 4 metros, o que indicou uma boa precisão.

Em um terceiro experimento, em pontos também diferentes, o veículo se distanciou do usuário cerca de 300 metros. A velocidade durante o trajeto foi de aproximadamente 30 km/h, e também não houveram frenagens bruscas. O intervalo nos valores das distâncias exibidas para o usuário variaram, como no experimento descrito anteriormente. No entanto, quando o veículo parou próximo ao usuário, a distância exibida na tela foi de 2 metros.

Os objetivos dos experimentos foram, primeiramente, validar a integração das partes que compõem a solução proposta, medir a precisão dos cálculos e das informações resultantes dos mesmos, e avaliar o comportamento do sistema como um todo. Com base nos resultados observados, concluiu-se que o sistema apresenta condições de atender às especificações desejadas, uma vez que as estimativas iniciais apresentaram um grau de precisão bom, mesmo levando em conta o nível de acurácia dos dois GPS's.

Figura 5 – Imagem mostrando o protótipo do dispositivo embarcado

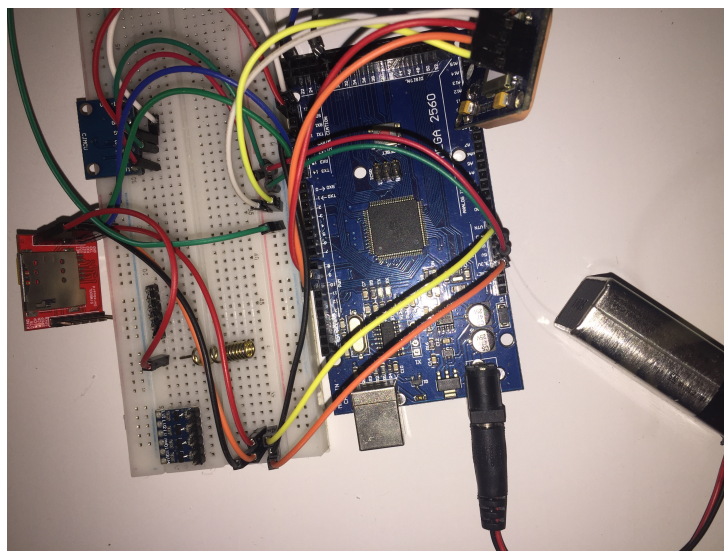
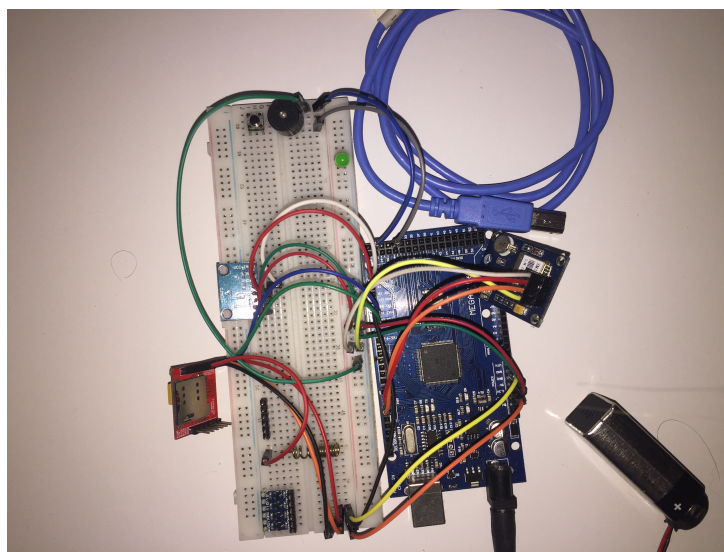


Figura 6 – Imagem mostrando o protótipo do dispositivo embarcado



As imagens acima mostram o protótipo do dispositivo IOT embarcado que foi confeccionado. Nelas é possível ver vários cabos e fios, juntamente com os módulos que compõem o dispositivo, e uma bateria de 9 volts, todos eles conectados ao Arduino ou a uma protoboard.

5 Conclusões

Com a finalização deste trabalho, chegou-se a algumas conclusões, tanto relacionadas a outras aplicações, como com relação aos resultados obtidos através dos experimentos com o sistema construído.

Percebeu-se que, apesar de no Brasil, até o período do desenvolvimento e conclusão desta pesquisa, haverem poucas soluções disponíveis no mercado que se proponham a atuar sobre o problema específico, descrito no capítulo de introdução, existem vários trabalhos acadêmicos voltados a esta problemática. Em outros países, existem soluções já implantadas, aparentemente funcionando.

Notou-se também, a existência de vários dispositivos jurídicos, tais como leis, decretos, convenções, fóruns, programas de incentivos, dentre outros, todos destinados a assegurar direitos das pessoas com deficiência e promover uma melhor acessibilidade na mobilidade urbana. No entanto, apesar de sua existência, ainda são poucas as soluções disponíveis, que atendam com eficiência e eficácia o simples ato de uma pessoa com deficiência visual identificar com autonomia um ônibus, e a mobilidade urbana ainda é encarada por órgãos especializados como um problema, sobre o qual ainda são feitos estudos e propostas de ações na tentativa de amenizá-lo.

Com relação à solução proposta, os experimentos comprovaram que, apesar da estrutura planejada, implementada e testada ser relativamente simples se comparada com a maioria das aplicações disponíveis no mercado (aplicações estas que apresentam pouca ou nenhuma acessibilidade), o sistema descrito neste trabalho cumpriu bem as funções para as quais ele foi desenvolvido.

5.1 Trabalhos Futuros

Como descrito nos resultados dos experimentos, o sistema apresentou em alguns instantes, inconsistências no que diz respeito aos cálculos da estimativa da distância entre o usuário e o ônibus. Estas inconsistências podem ser devido à precisão dos dados obtidos pelos GPS's dos celulares nos quais os aplicativos estavam sendo executados, uma vez que todos os dispositivos que atuam com o Sistema de posicionamento global apresentam um nível de acurácia variável.

Como explicado por especialistas na área e, indicado por fabricantes de receptores de GPS, existem vários fatores que interferem na qualidade do sinal recebido, fatores estes que vão desde a própria atmosfera terrestre, até a localização onde o receptor se encontra. Então, como um dos objetivos do sistema é atingir uma ótima precisão em seus cálculos, a fim de fornecer uma indicação mais acurada ao usuário, ainda é necessário o estudo e

a implementação de técnicas que auxiliem na redução e/ou compensação na diluição de precisão, uma vez que, em ambiente real, a qualidade dos receptores GPS nos celulares dos usuários variaria entre os fabricantes.

Também foi descrita a instabilidade em um dos módulos utilizados pelo dispositivo embarcado para realizar a conexão com a internet. A fim de proporcionar um melhor funcionamento do sistema, é necessário a utilização de módulos mais estáveis no dispositivo.

Por fim, apesar do aplicativo possuir funções básicas, que permitem o usuário selecionar dados referentes ao início de uma viagem, planejou-se a expansão de suas funcionalidades, a fim de possibilitar ao usuário uma gama maior de informações ao longo do seu trajeto. Informações como o local atualizado onde ele se encontra, sinalização de próximas paradas de ônibus em seu trajeto, indicação da proximidade de seu destino e alerta sobre quando desembarcar são algumas das funcionalidades planejadas para o futuro do projeto.

Todas estas funcionalidades, juntamente com o trabalho que se desenvolveu, foram pensadas com o objetivo de proporcionar às pessoas com deficiência visual ou não, uma ferramenta que as ajude em seu dia a dia, e contribua para uma melhor comodidade, autonomia e independência em suas vidas.

Referências Bibliográficas

- ALAMY, L. E. et al. Bus identification system for visually impaired person. *2012 Sixth International Conference on Next Generation Mobile Applications, Services and Technologies*, 2012. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/6327924>>. 20
- BHOWMICK, A.; HAZARIKA, S. An insight into assistive technology for the visually impaired and blind people: state-of-the-art and future trends. *Journal on Multimodal User Interfaces*, v. 11, p. 1–24, 01 2017. 14
- BIANCHI, E. A. Sistema para controle de frota do transporte coletivo com acessibilidade para deficientes visuais. *Inserir*, 2014. 19
- BRASIL. Lei brasileira de inclusão. *Disponível em: http://www.planalto.gov.br/ccivil_03/ato2015-2018/2015/Lei/L13146.htm*. Acesso em 04/04/2019, 2015. 17
- CANADA. Site oficial. 2014. Disponível em: <http://www.stm.info/en/about/major_projects/ibus-real-time>. 19
- CARVALHO, A. J. de. A importância do transporte público e da acessibilidade como meios de acesso a direitos de cidadania das pessoas com deficiência : o caso dos cadeirantes de Franca-SP. *Disponível em https://repositorio.unesp.br/bitstream/handle/11449/150733/carvalho_ajmefran.pdf?sequence=3*. Acesso em 30/03/2019, 2017. 17
- DEVELOPERS, A. *Wi-Fi scanning overview*. 2019. Disponível em: <<https://developer.android.com/guide/topics/connectivity/wifi-scan>>. 26
- DISCHINGER, C. S. M. Orientação e mobilidade de pessoas com deficiência visual no transporte público: discussões através de grupo focal nacional. *Disponível em <https://periodicos.ufrn.br/revprojetar/article/download/16575/11029/>*. Acesso em 01/04/2019, 2000. 18
- DOUKAS, C. *Building Internet of Things with the Arduino*. USA: CreateSpace Independent Publishing Platform, 2012. ISBN 1470023431, 9781470023430. 22, 24
- ESTATÍSTICA, I. I. B. de Geografia e. *IBGE | Censo 2010 | Resultados*. 2010. Disponível em: <<https://censo2010.ibge.gov.br/resultados.html>>. 13
- ESTATÍSTICA, I. I. B. de Geografia e. *IBGE | Censo 2010 | Resultados*. 2010. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religiao_deficiencia.pdf>. 13
- FIELDING, R. T. Architectural styles and the design of network-based software architectures. 2000. Disponível em: <<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>>. 31

- FLORES, G. H.; MANDUCHI, R. A public transit assistant for blind bus passengers. *IEEE Pervasive Computing*, IEEE Computer Society, Los Alamitos, CA, USA, v. 17, n. 01, p. 49–59, jan 2018. Disponível em: <<https://www.computer.org/csdl/magazine/pc/2018/01/mpc2018010049/13rRUILtJok>>. 20
- GIL, A. C. Como elaborar projetos de pesquisa. *Disponível em* http://docente.ifrn.edu.br/mauriciofacanha/ensino-superior/redacao-cientifica/livros/gil-a.-c.-como-elaborar-projetos-de-pesquisa.-sao-paulo-atlas-2002./at_download/fileAcessoem30/03/2019, v. 4, 2002. 17
- HOLLIER, S.; ABOU-ZAHRA, S. Internet of things (iot) as assistive technology: Potential applications in tertiary education. In: . [S.l.: s.n.], 2018. p. 1–4. 14
- KUMARI, A. Assistive technology for the visually impaired children for their academic excellence. *Global Journal of Medical Case Report*, Vol-1, p. 1–9, 11 2015. 14
- LTD, A. Site do desenvolvedor. 2010. Disponível em: <<http://www.apex-jesenice.cz/tyfloset.php?lang=en>>. 19
- MONTEIRO, R. dos R. *A Real-Time Reasoning Service for the Internet of Things*. 85 p. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, 2018. 22, 23
- MULFARI, D. et al. Achieving assistive technology systems based on iot devices in cloud computing. *EAI Endorsed Transactions on Cloud Systems*, v. 1, 02 2015. 14
- NOPPA. Personal navigation and information system for users of public transport. 2004. Disponível em: <<http://virtual.vtt.fi/virtual/noppa/noppaeng.htm>>. 20
- NOPPA. Site oficial. 2004. Disponível em: <<http://virtual.vtt.fi/virtual/noppa/mvenue/navigationssystemforthevisuallyimpairedbasedonaninformation.pdf>>. 19
- RAO., D. M. D. G. V. N. S. P. C. S. R. P. B. P. Bus identification system for the visually impaired: evaluation and learning from pilot trials on public buses in delhi. 2015. Disponível em: <<http://assistech.iitd.ernet.in/doc/TRANSED%202015-OnBoard.pdf>>. 19
- SILVA, J. P. da. Ecocit: Uma plataforma escalável para desenvolvimento de aplicações de iot. 2018. Disponível em: <https://repositorio.ufrn.br/jspui/bitstream/123456789/23779/4/JorgePereiraDaSilva_DISSERT.pdf.txt>. 23
- SILVA, R. C. M. A. N. R. da. A percepção do especialista sobre o tema mobilidade urbana. *Disponível em: https://revistatransportes.org.br/anpet/article/download/13/10* Acesso em 01/04/2019, 2008. 18
- SOCKET.IO. *Socket.IO Overview*. 2019. Disponível em: <<https://socket.io/docs/>>. 35
- SOUSA, C. R. M. de. Interferidores de gps: Análise do sistema e de potenciais fontes de interferência. 2005. Disponível em: <http://www.pgee.ime.eb.br/pdf/carlos_sousa.pdf>. 39
- SRI, T. A review on the state of art of internet of things. *INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER AND COMMUNICATION ENGINEERING*, v. 5, p. 38, 07 2016. 23, 24

URBANOS, C. N. do Transporte; Associação Nacional das Empresas de T. *Pesquisa mobilidade da população urbana 2017*. 2017. Disponível em: <<https://www.ntu.org.br/novo/upload/Publicacao/Publicacao/Pub636397002002520031.pdf>>. 15

WANG, H.-L. et al. An interactive wireless communication system for visually impaired people using city bus transport. *Int J Environ Res Public Health*, MDPI, v. 11, n. 5, p. 4560–4571, Apr 2014. Disponível em: <<https://www.ncbi.nlm.nih.gov/pubmed/24776720>>. 19

(WHO), W. H. O. *Blindness and vision impairment*. 2018. Disponível em: <<https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>>. 13

6 Apêndice

Figura 7 – Etapa de amadurecimento do protótipo de Hardware na FabLAB UCSAL.



Identificador	Nome	Categoria	Prioridade
RF01	Validação de Campos	Usabilidade	Importante
Rf02	Pesquisas	Usabilidade	Essencial
RF03	Estimativa de Distâncias	Usabilidade	Desejável
RF04	Notificações	Usabilidade	Essencial
RF05	Informações sobre Veículos	Usabilidade	Essencial
RF06	Continuidade na Execução	Usabilidade	Desejável
RF07	Alertas em Tempo Real	Usabilidade	Essencial

Tabela 1 – Requisitos funcionais do aplicativo

Identificador	Descrição
RF01	Todos os campos de entrada do usuário devem ser validados antes de serem enviados para a API externa
Rf02	O aplicativo deverá permitir a pesquisa por pontos, filtrando por bairro, rua, descrição, dentre outros
RF03	O aplicativo deverá informar as distâncias aproximadas entre a localização do usuário, pontos e ônibus em tempo real
RF04	O aplicativo deverá permitir que o usuário selecione os ônibus sobre os quais ele deseja ser notificado
RF05	O aplicativo deverá exibir informações sobre os ônibus, tais como nome ou código da linha, rota, etc
RF06	O aplicativo deverá continuar funcionando mesmo que esteja em segundo plano
RF07	O aplicativo deverá alertar ao usuário gradativamente à medida que um ônibus selecionado se aproxime de sua localização

Tabela 2 – Requisitos funcionais do aplicativo

Identificador	Nome	Categoria	Prioridade
RNF01	Multi-plataforma	Compatibilidade	Essencial
RNF02	Acesso à API	Interoperabilidade	Essencial
RNF03	Acessibilidade	Usabilidade	Essencial

Tabela 3 – Requisitos não funcionais do aplicativo

A seguir seguem os requisitos resultantes da etapa descrita anteriormente.

1. RNF01: O aplicativo deverá funcionar em dispositivos com sistema operacional Android ou Ios. Ele também deverá funcionar em navegadores como Google Chrome e Mozilla Firefox.
2. RNF02: O aplicativo deverá acessar as funções da api externa via internet.
3. RNF03: O aplicativo deverá seguir padrões de acessibilidade sugeridas por organizações como a W3C, tais como descrições em possíveis imagens, botões facilmente alcançáveis e auto-explicativos, dentre outros.

A seguir, são listados os requisitos não funcionais da aplicação:

Identificador	Nome	Categoria	Prioridade
RNF01	Plataforma	Compatibilidade	Essencial
RNF02	Acessibilidade	Usabilidade	Essencial
RNF03	Servidor de Execução	Disponibilidade	Essencial
RNF04	Armazenamento de Dados	Interoperabilidade	Desejável
RNF05	Acesso Autenticado	Segurança	Importante

Tabela 4 – Requisitos não funcionais da aplicação

Identificador	Nome	Categoria	Prioridade
RF01	Autenticação de Usuário	Segurança	Importante
RF02	Exibição	Usabilidade	Importante
RF03	Cadastro de Informações	Usabilidade	Importante
RF04	Cadastro de Endereços	Usabilidade	Importante
RF05	Composição de Itinerário	Usabilidade	Importante
RF06	Disponibilização de API	Usabilidade	Importante
RF07	Autenticação de Api	Segurança	Importante
RF08	Dados de Entrada da Api	Usabilidade	Desejável
RF09	Atualização em Tempo Real	Usabilidade	Essencial
RF10	Gerenciamento de Dispositivos	Segurança	Importante
RF11	Validação de Campos	Usabilidade	Importante

Tabela 5 – Tabela 7: requisitos funcionais da aplicação

1. RNF01: A aplicação deverá ser acessada e compatível com navegadores como Google Chrome e Mozilla Firefox.
2. RNF02: A interface da aplicação deverá seguir padrões de acessibilidade sugeridas por organizações como a W3C, tais como descrições em possíveis imagens, botões facilmente alcançáveis e auto-explicativos, dentre outros.
3. RNF03: A aplicação deverá rodar em servidor web.
4. RNF04: Os dados da aplicação devem ser armazenados em banco de dados PostgreSQL.
5. RNF05: A manipulação dos dados presentes na aplicação só deve acontecer mediante autenticação.

Identificador	Descrição
RF01	O sistema deve possuir funcionalidades de login, utilizando um código e uma senha
RF02	O sistema deve exibir telas distintas para cada tipo de informação
RF03	O sistema deve possibilitar o cadastro de ônibus e informações relacionadas aos mesmos
RF04	O sistema deve ser capaz de cadastrar endereços, juntamente com suas coordenadas geográficas
RF05	O sistema deve ser capaz de associar endereços seqüencialmente aos ônibus
RF06	O sistema deve expor uma api, que permita a consulta aos dados cadastrados
RF07	O acesso a Api precisa ser autenticado mediante a JWT
RF08	A api só deve responder a requisições em formato JSON
RF09	O sistema deve ser capaz de enviar informações em tempo real aos dispositivos conectados
RF10	O sistema deve ser capaz de gerenciar uma lista de dispositivos que desejem receber informações em tempo real
RF11	Todos os campos de entrada do usuário devem ser validados antes de serem persistidos no banco de dados

Tabela 6 – Requisitos funcionais da aplicação