

Rodrigo Caetano Soares
William Mesquita de Medeiros

**SISTEMA PARA O TRANSPORTE PÚBLICO URBANO
BASEADO NO CONCEITO DE SMART CITY**

Trabalho de Conclusão de Curso
submetido à Universidade Federal de
Santa Catarina para a obtenção do
Grau de Bacharel em Tecnologias da
Informação e Comunicação.

Orientador: Prof. Dr. Alexandre
Leopoldo Gonçalves.

Araranguá
2017

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Soares, Rodrigo Caetano
Sistema para o transporte público urbano baseado
no conceito de Smart City / Rodrigo Caetano Soares
; orientador, Alexandre Leopoldo Gonçalves, 2017.
92 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus
Araranguá, Graduação em Tecnologias da Informação e
Comunicação, Araranguá, 2017.

Inclui referências.

1. Tecnologias da Informação e Comunicação. 2.
Smart City. 3. Internet das Coisas. 4. Transporte
Público. 5. Beacon. I. Gonçalves, Alexandre
Leopoldo. II. Universidade Federal de Santa
Catarina. Graduação em Tecnologias da Informação e
Comunicação. III. Título.

Medeiros, William Mesquita de
SISTEMA PARA O TRANSPORTE PÚBLICO URBANO BASEADO
NO CONCEITO DE SMART CITY / William Mesquita de
Medeiros ; orientador, Alexandre Leopoldo Gonçalves,
2017.
92 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus
Araranguá, Graduação em Tecnologias da Informação e
Comunicação, Araranguá, 2017.

Inclui referências.

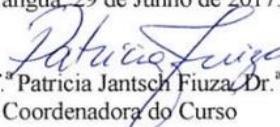
1. Tecnologias da Informação e Comunicação. 2.
Smart city. 3. Internet da coisas. 4. Transporte
publico. 5. Beacon. I. Gonçalves, Alexandre
Leopoldo. II. Universidade Federal de Santa
Catarina. Graduação em Tecnologias da Informação e
Comunicação. III. Título.

Rodrigo Caetano Soares
William Mesquita de Medeiros

**SISTEMA PARA O TRANSPORTE PÚBLICO URBANO
BASEADO NO CONCEITO DE SMART CITY**

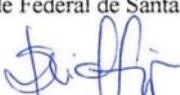
Esta Monografia foi julgada adequada para obtenção do Título de “Bacharel em Tecnologias da Informação e Comunicação”, e aprovada em sua forma final pelo Curso de Graduação em Tecnologias da Informação e Comunicação.

Araranguá, 29 de Junho de 2017.


Prof.^a Patricia Jantsch Fiuza, Dr.^a
Coordenadora do Curso

Banca Examinadora:


Prof. Alexandre Leopoldo Gonçalves, Dr.
Orientador
Universidade Federal de Santa Catarina


Prof.^a Analúcia Schiaffino Morales, Dr.^a
Universidade Federal de Santa Catarina


Prof.^a Olga Yevseyeva, Dr.^a
Universidade Federal de Santa Catarina

Dedicamos este trabalho a todos que estiveram envolvidos direta ou indiretamente no desenvolvimento do mesmo, ou que o apoiaram de alguma forma.

AGRADECIMENTOS

Agradeço a todos que estiveram presentes nesta jornada que começou a 4 anos atrás. Agradeço à oportunidade que a Universidade Federal de Santa Catarina me forneceu. Agradeço ao curso de Tecnologias da Informação e Comunicação e todo seu corpo docente. Agradeço a minha família, principalmente minha mãe e avó. Minha mãe pela garra, e por ter me acompanhado no começo de tudo isso. Minha avó pela determinação, e por ter me dado o empurrão que eu precisava para perder o medo de correr atrás de algo novo. Agradeço ao nosso orientador Prof. Alexandre Leopoldo Gonçalves, por todo o auxílio e por ter apoiado a ideia desde seu princípio. Agradeço à minha parceira e namorada Andressa Ronçani Martinho, pelo apoio, pelas broncas, e principalmente pelo amor.

Rodrigo Caetano Soares

AGRADECIMENTOS

Primeiramente agradeço a minha família, a qual me apoiou ao ingressar no curso e promoveu suporte quanto às escolhas que fiz durante este período acadêmico. Agradeço à minha namorada Raíssa Cardoso Figueiredo a qual me incentivou e deu suporte nos momentos mais difíceis durante este período. Agradeço aos professores do curso de Tecnologias da Informação e Comunicação, os quais compartilharam seus conhecimentos e que através deles me tornei um profissional. Agradeço à ajuda quando necessária do corpo técnico da universidade. Agradeço também ao apoio dos colegas durante o curso. Agradeço também ao professor orientador Alexandre Leopoldo Gonçalves pelo interesse, empenho e suporte neste trabalho.

William Mesquita de Medeiros.

Inteligência é a capacidade de se adaptar à mudança.

Stephen Hawking

RESUMO

Devido ao processo de urbanização, a população das grandes cidades cresce a cada ano. Paralelo a isso, também cresce o desafio de manter os cidadãos dessas cidades satisfeitos com os serviços públicos. Um dos serviços que mais sofrem com o aumento populacional das cidades é o transporte público. Parte considerável da população nas grandes cidades depende do transporte público para trabalhar, fazendo com que este seja um dos serviços mais críticos. Para resolver este problema, a qualidade do serviço deve ser melhorada ao ponto de as pessoas ficarem satisfeitas ao utilizá-lo. Neste contexto, o conceito de uma *Smart City* visa promover satisfação e bem-estar às pessoas. Utilizando o paradigma da Internet das Coisas, a função de uma *Smart City* reside na entrega de serviços públicos mais inteligentes para seus cidadãos. Um dos componentes de uma *Smart City* é o *Smart Mobility*, que foca na satisfação da pessoa com o transporte público da cidade. Baseado nesses conceitos foi desenvolvido neste trabalho um protótipo de um sistema que utiliza sensores sem fio disponíveis em paradas de ônibus e no próprio ônibus para fazer o cálculo de tempo entre paradas, proporcionando aos usuários uma estimativa em tempo real entre uma parada inicial e uma parada final. Os valores dos tempos calculados ficam disponíveis para outros usuários em tempo real, possibilitando a obtenção de uma média de determinada rota. Além disso, os sensores foram utilizados para detectar a presença do usuário e enviar notificações sobre informações de uma determinada parada e horários das linhas de ônibus. Pode-se concluir que os resultados obtidos com este trabalho foram adequados, tanto no cálculo dos tempos médios quando nas notificações aos usuários sobre paradas, horários e rotas.

Palavras-chave: *Smart City*, Internet das Coisas, *Bluetooth Low Energy*, *Beacon*

ABSTRACT

Due to the process of urbanization, the population of large cities grows every year. Parallel to this, the challenge of keeping citizens of these cities satisfied with public services is also growing. One of the services that suffer most from the population increase of cities is the public transport. Considerable part of the population in large cities depends on public transport to work, making it one of the most critical services. To solve this problem, the quality of the service must be improved to the point that people are satisfied using it. In this context, the concept of a Smart City aims to promote satisfaction and well-being for people. Using the Internet of Things paradigm, the function of a Smart City lies in delivering smarter public services to its citizens. One of the components of a Smart City is Smart Mobility, which focuses on the people satisfaction with the city's public transportation. Based on these concepts, was developed a prototype of a system based on wireless sensors available at bus stops and at the bus itself to calculate the time between stops, giving users a real-time estimate between an initial and a stop end. The calculated time values are available to other users in real time, making it possible to obtain an average time value of a certain route. In addition, the sensors were used to detect the presence of the user and to send notifications about information of a certain stop as well as schedules of bus lines. It can be concluded that the results obtained with this work were suitable, both in the calculation of average times and in the notifications to users about stops, schedules and routes.

Keywords: Smart City, Internet of Things, Bluetooth Low Energy, Beacons

LISTA DE FIGURAS

Figura 1 – Fluxograma dos objetivos propostos.....	29
Figura 2 – Representação do conceito de <i>Smart City</i> baseada em eventos	34
Figura 3 - Características de uma <i>Smart City</i>	35
Figura 4 - Plano da cidade de Columbus (Ohio), vencedora do <i>Smart City Challenge</i>	40
Figura 6 - Rede <i>bluetooth</i>	45
Figura 7 – Gráfico de número de dispositivos por pessoa	49
Figura 8 - Arquitetura lógica do MobileBus	54
Figura 9 - Código em Delphi utilizando comandos SQL explícitos	55
Figura 10 - Código em Delphi utilizando o componente Aurelius	56
Figura 11 - Camadas entre a aplicação e o banco de dados	57
Figura 12 - Classe TBusExitTime implementada no sistema, exemplificando o mapeamento da classe	58
Figura 13 - Arquitetura do RemoteDB	59
Figura 14 - Arquitetura de dados	60
Figura 15 - Modelagem de classes do sistema MobileBus	63
Figura 16 - Tela principal do sistema MobileBus	66
Figura 17 - Segunda aba principal do sistema, com a sub aba TESTE selecionada	67
Figura 18 - Tela de cadastro de paradas de ônibus	69
Figura 19 - Ícone utilizado para marcar paradas de ônibus no mapa	69
Figura 20 - Tela de cadastro de horários.....	70
Figura 21 - Notificação de parada encontrada	71
Figura 22 - Descrição da notificação de parada encontrada.....	71
Figura 23 - Botão "Iniciar Teste" da tela principal.....	74
Figura 24 - Tela do segundo usuário, na tela para acompanhar as informações da rota.....	75
Figura 25 - Tela da aplicação mostrando reconhecimento do Ponto A.....	76
Figura 26 - Tela da aplicação mostrando reconhecimento da Linha A	77
Figura 27 - Tela da aplicação mostrando reconhecimento do Ponto B	78
Figura 28 - Tela do segundo usuário apresentando o tempo atualizado.....	79
Figura 29 - Tela da aplicação mostrando dados do <i>beacon</i>	80
Figura 30 - Tela mostrando o sistema depois que usuário seleciona a notificação	81

LISTA DE QUADROS

Quadro 1 - Aspectos da vida urbana relacionado com características de uma Smart City	35
Quadro 2 - Fatores e indicadores para Smart Mobility	37
Quadro 3 - Requisitos funcionais de software	52
Quadro 4 - Requisitos funcionais de <i>hardware</i>	53
Quadro 5 - Requisitos não funcionais de <i>software</i>	53
Quadro 6 - Requisitos não funcionais de <i>hardware</i>	53

LISTA DE ABREVIATURAS E SIGLAS

API – *Application Programming Interface*
ANTT – Agência Nacional de Transportes Terrestres
ARPA – *Advanced Research Projects Agency*
BLE – *Bluetooth Low Energy*
BRT – *Bus Rapid Transit*
HTTP – *Hyper Text Transfer Protocol*
IBGE – Instituto Brasileiro de Geografia e Estatística
IOP – *Internet of People*
IOS – *Internet of Service*
IOT – *Internet of Things*
IPv6 – *Internet Protocol version 6*
FTP – *File Transfer Protocol*
LOD – *Linking of Data*
LTK – *Long Term Key*
MAC – *Media Access Control*
MIT – *Massachusetts Institute of Technology*
NFC – *Near Field Communication*
ORM – *Object Relation Mapping*
P&D – Pesquisa e Desenvolvimento
PLC – *Power Line Communication*
RFID – *Radio Frequency Identification*
RTTI – *Runtime Type Information*
SC4A – *Smart City for All*
RSSI – *Received Signal Strength Indicator*
SQL – *Structured Query Language*
SIG – *Bluetooth Special Interest Group*
STK – *Short Term Key*
TIC – Tecnologias da Informação e Comunicação
TCP/IP – *Transmission Control Protocol/Internet Protocol*
TK – *Temporary Key*
UFSC – Universidade Federal de Santa Catarina
UDP – *User Datagram Protocol*
UN – *United Nations*
UNFPA – *United Nations Population Fund*
US DOT – *United States Department of Transportation*
WLAN – *Wireless Local Area Network*
WMAN – *Wireless Metropolitan Area Network*
WPAN – *Wireless Personal Area Network*
WWAN – *Wireless Wide Area Network*

SUMÁRIO

1 INTRODUÇÃO	25
1.1 PROBLEMÁTICA	26
1.2 OBJETIVOS.....	27
1.2.1 <i>Objetivo Geral</i>	27
1.2.2 <i>Objetivos Específicos</i>	28
1.3 JUSTIFICATIVA E MOTIVAÇÃO	28
1.4 PROCEDIMENTOS METODOLÓGICOS	29
1.5 ORGANIZAÇÃO DO TEXTO	29
2 SMART CITY.....	31
2.1 INTRODUÇÃO.....	31
2.2 CONCEITUALIZANDO <i>SMART CITY</i>	32
2.3 PARAMETRIZANDO <i>SMART CITY</i>	34
2.3.1 <i>Smart Economy</i>	36
2.3.2 <i>Smart People</i>	36
2.3.3 <i>Smart Governance</i>	37
2.3.4 <i>Smart Mobility</i>	37
2.3.5 <i>Smart Environment</i>	38
2.3.6 <i>Smart Living</i>	38
2.4 INICIATIVAS DE <i>SMART CITY</i> AO REDOR DO MUNDO.....	39
3 INTERNET DAS COISAS	40
3.1 INTRODUÇÃO.....	41
3.2 REDES	41
3.2.1 <i>Redes sem Fio</i>	42
3.2.2 <i>Bluetooth</i>	44
3.3 INTERNET DAS COISAS E O IMPACTO SOCIAL	46
3.4 IMPASSES PARA IMPLANTAÇÃO DA INTERNET DAS COISAS	47
3.4.1 <i>Escalabilidade</i>	47
3.4.2 <i>Big data e dados gerados</i>	47
3.4.3 <i>Segurança</i>	48
3.4.4 <i>Privacidade</i>	48
3.5 INTERNET DAS COISAS E O MERCADO.....	48
3.5.1 <i>Crescimento da Internet das Coisas</i>	48
3.5.2 <i>O Valor da Informação</i>	49
4 SISTEMA PROPOSTO.....	51

4.1	INTRODUÇÃO.....	51
4.2	REQUISITOS.....	52
4.2.1	<i>Requisitos funcionais</i>	52
4.2.2	<i>Requisitos não funcionais</i>	53
4.3	ARQUITETURA.....	54
4.3.1	<i>Arquitetura lógica</i>	54
4.3.2	<i>Arquitetura física</i>	55
4.3.3	<i>Sensor sem fio</i>	60
4.3.4	<i>Modelo de dados e classes</i>	62
4.3.5	<i>Funcionamento</i>	64
5	APRESENTAÇÃO DOS RESULTADOS.....	73
5.1	criação do cenário.....	73
5.2	apresentação dos resultados.....	74
6	CONSIDERAÇÕES FINAIS.....	82
6.1	trabalhos futuros.....	83
	REFERENCIAS.....	86

1 INTRODUÇÃO

No século passado, apenas 20% da população mundial vivia em áreas urbanas. Nos países menos desenvolvidos este número chegava a apenas 5%. Em 2008 houve uma convergência na proporção, onde a maioria passou a viver nas cidades, e em 2050 é esperado que essa proporção aumente para dois terços (UNITED NATIONS, 2012).

Segundo Schuurman et al. (2012, p. 50), a crescente urbanização é causada por fatores gerados por processos envoltos na globalização, tais como mudanças tecnológicas, sociais e econômicas. Tais mudanças representam desafios para a gestão pública no que diz respeito à efetividade nas demandas sociais, exigindo novas abordagens para o planejamento, desenvolvimento, execução e operação. Para o enfrentamento desses desafios surgiu o conceito de *Smart City*.

O termo *Smart City* foi adotado em 2005 por algumas empresas de tecnologia (Cisco®, IBM®, Siemens®) para representar sistemas de informação que integram serviços de infraestrutura urbana, ou qualquer tipo de inovação tecnológica para o planejamento, desenvolvimento e operação de cidades (HARRISON; DONNELLY, 2011).

As *Smart Cities* são uma consequência da evolução da tecnologia da informação (HAO; LEI; YAN, 2012). Como exemplo pode-se citar o Padova *Smart City Project*, projeto de *Smart City* na cidade de Padova, Itália. No projeto foram realizadas especificações de alguns serviços de utilidade pública que seriam integrados, tais como: monitoramento de barulhos, congestionamento de trânsito, consumo de energia da cidade, estacionamentos inteligentes, monitoramento da qualidade do ar, entre outros.

Um dos componentes de uma *Smart City* é a *Smart Mobility*, ou Mobilidade Inteligente. Este conceito está relacionado diretamente ao transporte e disponibilidade de infraestruturas de TICs. Para uma *Smart City* o transporte deve ter um sistema sustentável, inovador, seguro e, principalmente, que atenda as demandas urbanas (GIFFINGER, 2007).

Cada serviço de uma *Smart City* pode e, provavelmente, terá tecnologias de acesso diferentes (Ethernet, Wi-Fi, Bluetooth, etc.). Para que o conceito se mantenha, é necessária a capacidade de interconectar tecnologias heterogêneas de forma fácil e transparente. Neste sentido, surge a Internet das Coisas, ou *Internet of Things* (IoT), vista como a chave para essa integração (ZANELLA et al., 2014).

O termo Internet das Coisas surgiu em 1999 no título de um trabalho de Kevin Ashton, cofundador e diretor executivo da Auto-ID Center, laboratório de pesquisa na área de IoT. Kevin havia escolhido a

frase para chamar a atenção dos outros executivos sobre uma nova ideia de RFID que iria apresentar (ASHTON, 2009).

De acordo com Atzori, Iera e Morabito (2010), a IoT pode ser definida como um paradigma de comunicação, onde os objetos, ou coisas, estão equipados com transmissores, sensores e micro controladores que se comunicam entre si, e se tornam uma parte integrada da Internet.

No contexto urbano, um sistema de IoT necessita de tecnologias de rede para cobrir uma grande área e suportar o alto tráfego de dados, devido aos vários dispositivos conectados. Tais tecnologias podem se dividir entre dois grandes grupos: não restritas e restritas (ZANELLA et al, 2014).

Ainda segundo Zanella et al. (2014, p. 27), as tecnologias de rede não restritas são as mais comuns, como a *Local Area Network* (LAN), *Metropolitan Area Network* (MAN) e *Wide Area Network* (WAN). Neste grupo se incluem a Ethernet, Wi-Fi, fibra ótica, etc. Todas elas possuem baixa latência e alta transferência de dados.

O segundo grupo, as tecnologias restritas, geralmente são caracterizadas como de baixo consumo de energia e taxas de transferência baixa. Estão incluídas neste grupo as tecnologias *Power Line Communication* (PLC), *Near Field Communication* (NFC), *Radio Frequency Identification* (RFID), *Bluetooth Low Energy* (BLE), entre outras.

1.1 PROBLEMÁTICA

A utilização do transporte público é algo cotidiano e corriqueiro para muitas pessoas. Porém, em algumas cidades a má qualidade do sistema afeta os usuários. Como aponta Antunes e Simões (2013) existem aspectos que devem ser analisados para definir a qualidade do serviço de transporte coletivo por ônibus. O autor então cita alguns deles, tais como, acessibilidade, tempo de viagem, confiabilidade e pontualidade.

Segundo Ferraz e Torres (2004, apud SIMÕES, 2013), a acessibilidade é definida como a distância entre o ponto inicial e final do usuário, levando também em consideração a comodidade. A acessibilidade está diretamente relacionada com as características da rede de transporte, como a distância entre os pontos de parada. O tempo de viagem é definido como o tempo entre um ponto e outro, levando em consideração a distância, a velocidade e as estradas. Pontualidade se refere ao tempo de espera do usuário do transporte público no ponto de

ônibus. Por fim, a confiabilidade é o grau de certeza de que o usuário vai sair do ponto de origem e chegar ao seu destino.

Visto que a qualidade do transporte depende de aspectos como tempo de viagem e pontualidade, é necessário que dados sejam constantemente coletados, analisados e disponibilizados através de sistemas computacionais visando impactar positivamente na experiência de usuários em relação ao transporte público. Por exemplo, o tempo que um ônibus leva de um ponto A até um ponto B provavelmente não será o mesmo em uma segunda-feira ao meio dia ou em uma sexta-feira no final da tarde, principalmente, se a rota inclui vias muito movimentadas.

As manifestações de junho e julho no Brasil em 2013, decorrentes do aumento no preço das passagens de ônibus, demonstrou a insatisfação da população com o transporte público. Além disso, acabou trazendo à tona outros problemas relacionados ao transporte público, como a precariedade do serviço e sua falta de produtividade (MÁRCIO ROGÉRIO SILVEIRA, 2013).

O setor do transporte público urbano é de grande importância para a população, principalmente de grandes cidades, pois é através dele que grande parte das pessoas se desloca para trabalhar, vão para o hospital, ou saem para algum tipo de lazer. Baseado nisso, e na insatisfação dos usuários com este setor, torna-se relevante o desenvolvido constante de inovações tecnológicas que visem auxiliar na melhoria do serviço. A empresa Uber®, que se baseia em um aplicativo para dispositivos móveis para disponibilizar motoristas contratados e atender usuários é um exemplo de inovação tecnológica na mobilidade urbana.

Dentro deste contexto, apresenta-se a pergunta de pesquisa **“Como a Internet das Coisas pode contribuir na melhoria da experiência do usuário em relação ao transporte público levando-se em conta o contexto das *Smart Cities*?”**

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Desenvolver um sistema para dispositivos móveis no contexto de transporte público que, através de sensores sem fio, auxilie usuários com informações úteis para o seu dia a dia

1.2.2 Objetivos Específicos

De modo a atingir o objetivo geral deste trabalho os seguintes objetivos específicos são requeridos:

- Estudar sistemas de comunicação sem fio, envolvendo os conceitos de IoT e *Smart City*;
- Apresentar a tecnologia de sensores sem fio, sua evolução e aplicação em dispositivos móveis;
- Com base na tecnologia discutida desenvolver um protótipo de um sistema que interaja com o usuário através de seu *smartphone* visando demonstrar a viabilidade da proposta;
- Apresentar e analisar os resultados obtidos.

1.3 JUSTIFICATIVA E MOTIVAÇÃO

O meio de transporte coletivo através de ônibus é a realidade de muitas pessoas no Brasil e no mundo, sendo um dos meios de transporte mais acessíveis para a população em geral. Em muitas cidades ele é o único meio de transporte público existente. Segundo dados da Agência Nacional de Transportes Terrestres (ANTT) (2015), através de seu relatório anual referente ao ano de 2013, cada ônibus no Brasil é utilizado por 511 pessoas diariamente.

Este número se torna pequeno quando analisado o crescimento da frota de carros do país. De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE) (2015), a quantidade de habitantes por automóvel já está com uma relação de 1 para 4, e crescendo. Caso este ritmo permaneça, estima-se que o número chegue a 94 milhões de automóveis em 2022.

Esse crescimento é claramente notável em grandes cidades. Notícias sobre congestionamentos de quilômetros são comuns, assim como, o fato de levar horas para percorrer um trajeto de minutos. Isso ocorre principalmente porque a estrutura das estradas não acompanha o ritmo de crescimento da frota de veículos.

Uma possível solução para tal problema seria proporcionar as pessoas uma infraestrutura adequada de transporte público, diminuindo assim, a quantidade de veículos nas estradas. Porém, como mostra Antunes (2013), a decisão do usuário sobre qual meio de transporte utilizar vai depender da qualidade e eficiência do mesmo.

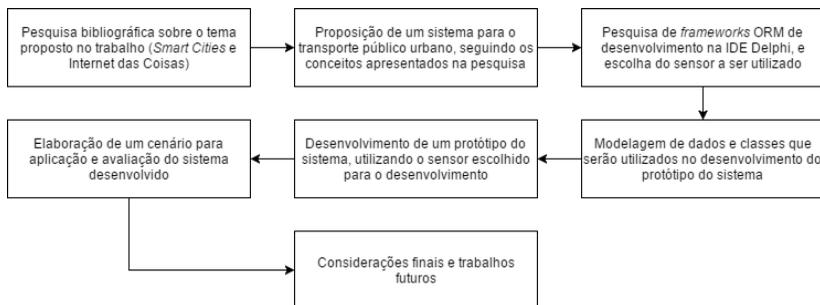
1.4 PROCEDIMENTOS METODOLÓGICOS

Este trabalho consiste em uma pesquisa bibliográfica, que sob o ponto de vista de sua natureza é caracterizado como uma pesquisa tecnológica e aplicada.

Segundo Silva e Menezes (2005), uma pesquisa aplicada caracteriza-se como uma pesquisa que “objetiva gerar conhecimentos para aplicação prática, dirigidos à solução de problemas específicos”. De acordo com a modalidade, caracteriza-se como uma pesquisa tecnológica uma vez que propõe a criação de um artefato tecnológico.

Deste modo, para atingir os objetivos propostos neste trabalho, os seguintes passos foram executados conforme apresentado na Figura 1.

Figura 1 – Fluxograma dos objetivos propostos



Fonte: Autores

1.5 ORGANIZAÇÃO DO TEXTO

Este trabalho, além desta introdução, está dividido em outros cinco capítulos.

O capítulo 2 apresenta uma revisão sobre os *Smart Cities*, suas definições de conceitos, classificações e iniciativas ao redor do mundo.

O capítulo 3 apresenta uma revisão sobre as tecnologias de redes, redes sem fio e Internet das Coisas.

O capítulo 4 apresenta a proposição do sistema para o transporte público urbano e as tecnologias que foram utilizadas para desenvolvê-lo.

O capítulo 5 apresenta o cenário para a avaliação do sistema e os resultados obtidos com o mesmo.

Por fim, no capítulo 6, são apresentadas as considerações finais e os trabalhos futuros.

2 SMART CITY

2.1 INTRODUÇÃO

No século 20, apenas 20% da população mundial residia em cidades. Com o passar dos anos a proporção aumentou consideravelmente, com a expectativa de que em 2050 o número chegue a quase 67%. Na Europa essa porcentagem está em 75%, sendo previsto que chegue a 80% até 2020 (UNITED NATIONS, 2012).

Em paralelo com a população, o tamanho das cidades também aumenta, gerando o que é chamado de processo de urbanização (SCHUURMAN et al., 2012). Este processo pode ser confirmado com o surgimento de megacidades com população de até 20 milhões de pessoas, em continentes como Ásia, África e América (UNITED NATIONS, 2012).

Ainda segundo Schuurman et al. (2012, p. 50), a urbanização é causada por fatores gerados por processos da globalização, tais como mudanças tecnológicas, sociais e econômicas, fazendo com que as cidades cresçam em tamanho e também em número populacional. Isso força a criação de soluções com inovação e sustentabilidade, utilizando as Tecnologias da Informação e Comunicação (TICs).

De acordo com Albino (2015), cidades ao redor do mundo procuram soluções que possibilitem meios de transporte com ligações com outros meios, uso de um transporte terrestre misto, e serviços urbanos de alta qualidade que possam, no longo prazo, impactar positivamente na economia. Albino ainda cita que um dos elementos-chave para o crescimento das cidades é um transporte público mais eficiente que, por exemplo, conecte o trabalho ao trabalhador.

Neste contexto, o conceito *Smart City* foi criado para representar um novo tipo de cidade, com foco em tecnologia e no cidadão. Existem outros termos análogos como *Digital City*, *Intelligent City*, *Virtual City*, *Ubiquitous City*. Embora tais termos tenham em comum a utilização de TICs, são utilizados para representar diferentes tipos de cidades. Eles se referem a cidades mais específicas e com um nível menor de integração que a *Smart City* (SCHUURMAN et al., 2012; ALBINO, 2015).

A *Intelligent City* faz uso das tecnologias da informação para melhorar a vida das pessoas. A utilização da palavra “*intelligent*” se faz pela necessidade de suportar o conhecimento, o desenvolvimento tecnológico e a inovação. Por outro lado, a *Digital City* não inclui o suporte para conhecimento e inovação, apenas a tecnologia. Sendo assim, toda *Intelligent City* tem componentes digitais, porém nem toda

Digital City é necessariamente inteligente. A diferença entre uma cidade “*intelligent*” e “*smart*” é a inclusão do componente “pessoas” na segunda, visto que uma *Smart City* é focada no cidadão (TOWNSEND, 2013 apud ALBINO, 2015).

Diferente das citadas anteriormente, a *Virtual City* passa a ideia da disponibilização de um espaço virtual para as pessoas, onde é possível a visualização de elementos urbanos. E por último, a *Ubiquitous City* é uma extensão da *Digital City* nos termos de acessibilidade em grande escala, com inclusão da computação ubíqua¹ em todos os lugares. A ideia principal é da criação de um ambiente onde os cidadãos poderiam acessar serviços urbanos em qualquer lugar, a qualquer hora e em qualquer dispositivo (TOWNSEND, 2013 apud ALBINO, 2015).

2.2 CONCEITUALIZANDO *SMART CITY*

Nas últimas décadas, o termo *Smart City* se tornou muito popular no meio científico. Essa popularidade é resultado do avanço das TICs em paralelo com o crescimento populacional das cidades. Porém, por mais popular que seja o conceito, ainda não é bem definido pelos pesquisadores, e não tem um significado unificado (ALBINO, 2015). O fato de não ter uma definição única resultou no desenvolvimento de diferentes definições.

O termo surgiu nos anos 90 com o avanço de novas tecnologias no que diz respeito à infraestrutura moderna nas cidades. A ideia foi criticada por ter um significado muito técnico, e também por ser utilizada com o propósito de fazer *marketing* (HOLLANDS, 2008).

Na tradução literal para o português, “*Smart*” significa “inteligente”. Porém, nesse contexto, a palavra tem um significado mais abrangente, moderno e tecnológico. Por exemplo, hoje temos *Smartphones*, *Smart TVs*, *Smart Houses*, etc., todos trazendo tecnologias avançadas ou algum tipo de inovação para os dispositivos.

Harrison et al. (2010) cita a *Smart City* como sendo uma cidade inteligente, interconectada e instrumentada. “Instrumentada” se refere à capacidade de capturar dados do mundo real utilizando sensores, medidores, dispositivos pessoais ou qualquer outro tipo de dispositivo que permita a captura de dados. “Interconectada” no sentido de utilizar os dados capturados em sistemas de computação que permitam a

¹ Computação ubíqua se refere à interação das pessoas com tecnologias de forma transparente, no sentido de elas não perceberem esta interação, pois ela é natural (FLOERKEMEIER; LAMPE, 2004).

disseminação de tais informações entre os vários serviços disponibilizados pela cidade. Por fim, “Inteligente” no sentido de realizar análises complexas desses dados, utilizar serviços de modelagem, otimização e visualização que permitam uma melhor tomada de decisão.

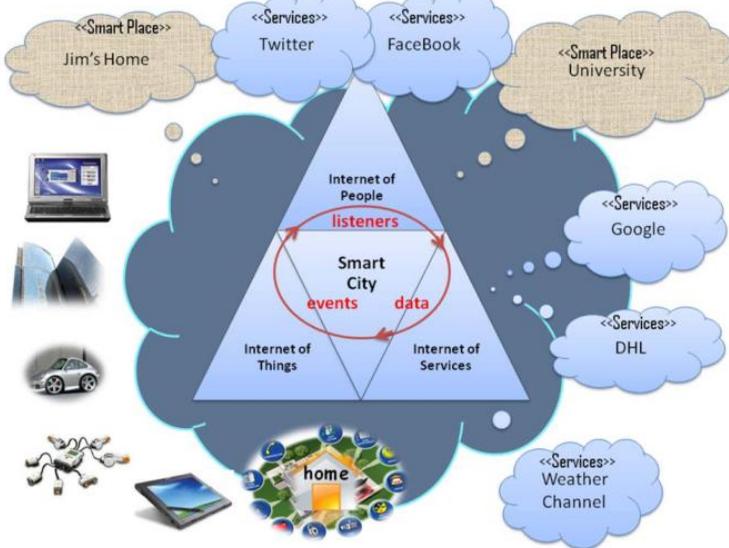
Para Barrionuevo et al. (2012), uma *Smart City* deve utilizar das tecnologias disponíveis para criar centros urbanos integrados, habitáveis e sustentáveis, de forma coordenada e inteligente. Além disso, o autor identifica cinco tipos de indicadores que contribuem na inteligência de uma cidade: econômico (Produto Interno Bruto, investimentos estrangeiros), humano (inovação, criatividade, educação), social (religião, tradições), ambiental (políticas de energia, gerenciamento de desperdícios) e institucional (eleições).

Caragliu et al. (2011) define a *Smart City* como uma cidade que investe em capital humano e social, infraestrutura de comunicação moderna, combustível sustentável, crescimento econômico e qualidade de vida, com um gerenciamento de recursos naturais inteligentes e participação governamental. A definição do autor foi baseada nas seis características de *Smart City* identificadas por Giffenger (2007): *Smart Economy*, *Smart Mobility*, *Smart Environment*, *Smart People*, *Smart Living* e *Smart Governance*.

As cidades que mais se destacam no quesito *smartness* focam no lado humano ao invés de simplesmente investir em tecnologia. As TICs devem ser utilizadas em prol do cidadão, e não apenas para deixar a cidade *smart* (Shapiro, 2006). Segundo Albino (2015), uma *Smart City* deve ser baseada na capacidade de pessoas inteligentes gerarem soluções inteligentes para problemas urbanos.

Cretu (2012) define a *Smart City* como a habilidade em fornecer toda a estrutura requerida pelos componentes da cidade (pessoas, serviços de software, dispositivos e sensores) para produzir, descobrir, entender e processar eventos em tempo real. A Figura 2 representa o *Event-driven Smart City* (EdSC), ou *Smart City* baseada em eventos. Este sistema foi criado por Cretu (2012), e representa um ambiente digital onde os componentes citados anteriormente interagem por meio de eventos e sensores.

Figura 2 – Representação do conceito de *Smart City* baseada em eventos



Fonte: Extraído de Cretu (2012)

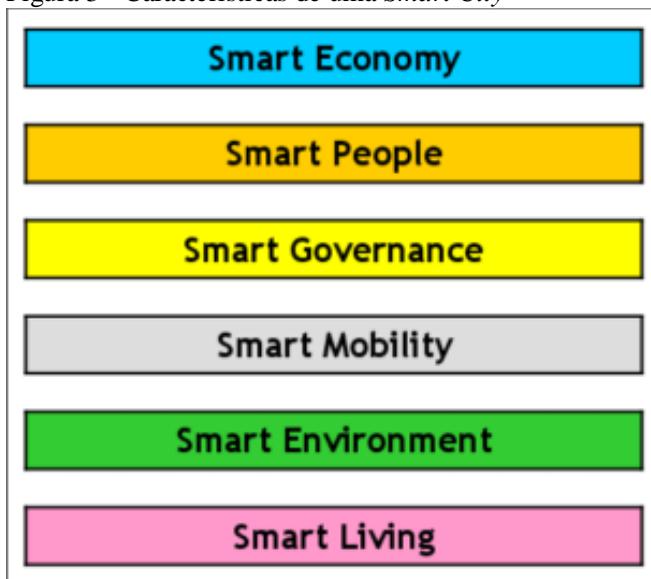
Ainda segundo Cretu (2012), a Internet evoluiu para um mundo multidimensional que abrange quatro grandes áreas: *Internet of People* (IoP, representa a mídia social), *Web of Data* (com o projeto da comunidade *Linking Open Data* - LOD), *Internet of Services* (IoS, se refere a serviços RESTful) e a *Internet of Things* (IoT, se referindo ao mundo de sensores sem fio e dispositivos inteligentes). Interconectar todos estes mundos é um objetivo essencial em qualquer *Smart City*.

Para Giffinger (2017), a *Smart City* deve ter uma boa perspectiva na economia, pessoas, governo, transporte e ambiente, misturando decisões inteligentes e cidadãos independentes e ativos. *Smart City* geralmente se refere ao uso de soluções inteligentes para melhorar a vida do cidadão.

2.3 PARAMETRIZANDO SMART CITY

Para definir se uma cidade é “*Smart*”, ou o quão “*Smart*” ela é, devem existir parâmetros para mensurá-la. Visando isso, Giffinger (2007) elaborou um sistema de ordenação de cidades. Nele foi identificado seis características que podem definir uma *Smart City*, como mostra a Figura 3.

Figura 3 - Características de uma *Smart City*



Fonte: Extraído de Giffinger (2007)

Essas características formam a estrutura para a definição do desempenho de uma *Smart City* como inteligente. O Quadro 1 relaciona alguns aspectos da vida urbana com os componentes de uma *Smart City*.

Quadro 1 - Aspectos da vida urbana relacionado com características de uma *Smart City*

Componentes de uma <i>Smart City</i>	Aspectos relacionados com a vida urbana
<i>Smart Economy</i>	Indústria
<i>Smart People</i>	Educação
<i>Smart Governance</i>	e-Democracia
<i>Smart Mobility</i>	Logística e Infraestrutura
<i>Smart Environment</i>	Eficiência e Sustentabilidade
<i>Smart Living</i>	Segurança e qualidade de vida

Fonte: Adaptado de Lombardi et al. (2012)

2.3.1 *Smart Economy*

Giffinger (2007) relaciona a característica *Smart Economy* com a competitividade econômica, e cita alguns fatores neste sentido, como inovação, empreendedorismo, produtividade, flexibilidade no mercado de trabalho, habilidade de adaptação, entre outros. Lombardi et al. (2012) relaciona *Smart Economy* com a presença de indústrias na área das TICs, ou o uso das TICs na produção de processos. Lombardi et al. (2012) e Lazaroïu e Roscia (2012) propõe alguns indicadores para avaliação de *Smart Cities* dentro do contexto *Smart Economy*, como gastos públicos com pesquisa e desenvolvimento (P&D), gastos públicos com educação, Produto Interno Bruto (PIB) por capita da cidade, índice de desemprego, entre outros. Estes indicadores são levados em consideração em alguns sistemas de avaliação de *Smart City*.

2.3.2 *Smart People*

Albino (2015) cita a educação como um indicador para *Smart People*. Porém, este termo é muito mais amplo que apenas um bom aproveitamento curricular. Ainda segundo Albino, as pessoas são as protagonistas das *Smart Cities*, que as molda em uma contínua interação. Por isso, outros termos acabaram sendo associados com *Smart City*, como a criatividade (uma cidade realmente inteligente deve ser criativa), conhecimento, aprendizado e, finalmente, educação, que foi utilizado para englobar todos os outros.

Alguns indicadores para a avaliação da presença de *Smart People* em uma *Smart City* são: porcentagem da população com ensino médio, línguas estrangeiras, participação em *life-long learning*², nível individual de habilidades com computadores, pedidos de patentes por habitante, entre outros (LOMBARDI et al., 2012; LAZAROÏU; ROSCIA, 2012).

² Em tradução direta, “aprendizado ao longo da vida”, significa uma busca contínua, voluntária e auto motivacional por novos conhecimentos, seja por razões pessoais ou profissionais. Dessa forma, não apenas melhora a inclusão social do indivíduo, cidadania ativa e desenvolvimento pessoal, como também a auto sustentabilidade, competitividade e empregabilidade.

2.3.3 Smart Governance

Smart Governance significa, segundo Albino (2015), “governança mediada pela tecnologia”. Além disso, as decisões do governo devem envolver vários *stakeholders*³, e sempre focadas e direcionadas ao cidadão. A transparência também é fundamental, fazendo com que conceitos como *open data*⁴ se tornem essenciais, criando novos canais de comunicação entre governo e cidadão. *Smart Governance* também pode ser chamado de *e-governance* que, de acordo com Lombardi et al. (2012) se com o conceito de com *e-democracy*.

2.3.4 Smart Mobility

Este componente está relacionado diretamente ao transporte e disponibilidade de estruturas de TICs no setor. Para uma *Smart City* o transporte deve ter um sistema sustentável, inovador, seguro e, principalmente, que atenda as demandas urbanas. Além disso, deve garantir a acessibilidade local e internacional dos cidadãos (GIFFINGER, 2007).

Giffenger (2007) utilizou informações de diferentes bases de dados de cidades europeias e extraiu desses dados alguns fatores (e indicadores para cada fator) para avaliação e classificação das cidades, no contexto de *Smart City*. Infelizmente, tais dados são datados entre 2001 e 2007, ficando bem defasados nos dias de hoje. Porém os indicadores que foram extraídos ainda são utilizados em diversos estudos sobre o assunto. Os fatores e indicadores para *Smart Mobility* são mostrados no Quadro 2.

Quadro 2 - Fatores e indicadores para Smart Mobility

Fator	Indicador
Acessibilidade local	<ul style="list-style-type: none"> • Rede de transporte público por habitante • Satisfação pelo acesso ao transporte público • Satisfação com a qualidade do transporte

³ Qualquer pessoa ou grupo que possui algum tipo de participação em algum processo é um *stakeholder*. Do inglês, *stake* significa interesse, participação, enquanto *holder* significa aquele que possui.

⁴ Vem da ideia de que dados devem estar disponíveis para todos, a qualquer momento, e que pode ser livremente distribuído e compartilhado por qualquer pessoa.

	público
Acessibilidade internacional	<ul style="list-style-type: none"> • Acessibilidade internacional
Disponibilidade de infraestrutura de TICs	<ul style="list-style-type: none"> • Computadores em domicílios • Acesso à banda larga em domicílio
Transporte público sustentável, inovador e seguro	<ul style="list-style-type: none"> • Tráfego individual sem motor (sustentável) • Segurança no trânsito • Uso de carros econômicos

Fonte: Adaptado de Giffinger (2007)

2.3.5 Smart Environment

Segundo Giffinger (2007), *Smart Environment* é descrito como a forma com que a cidade trata seus bens naturais, como clima e áreas verdes, poluição, gerenciamento de recursos, e também seus esforços com proteção ambiental.

Este item está diretamente relacionado aos recursos naturais das cidades. Um de seus indicadores é a presença de estratégias de redução de CO₂ do ambiente. Além disso, também são indicadores o consumo de energia por pessoa da cidade, uso eficiente de energia, consumo de água anual por pessoa, uso eficiente da água, total de área verde, percentual da população indo para o trabalho com transporte público, entre outros (LOMBARDI et al., 2012).

2.3.6 Smart Living

Giffinger (2007) relaciona *Smart Living* com qualidade de vida. Para o autor, alguns dos fatores deste conceito são: instalações culturais, condições de saúde, segurança individual, qualidade habitacional, instalações educacionais, atrações turísticas e igualdade social. Lazarou e Roscia (2012) ainda citam alguns indicadores de *Smart Living*, como: área dedicada à recreação de esportes e leitura, quantidade de bibliotecas públicas, visitas em museus e atendimento de teatro e cinema.

2.4 INICIATIVAS DE *SMART CITY* AO REDOR DO MUNDO

Como discutido anteriormente, uma *Smart City* é um conceito baseado, entre outras coisas, na utilização do paradigma da Internet das Coisas para possibilitar o desenvolvimento de uma estrutura de comunicação capaz de coletar dados de diferentes dispositivos, e os utilizar estes dados para melhorar a qualidade de vida das pessoas. Por mais que interesse comercial e científico na IoT tenha aumentado muito nos últimos anos, experimentos práticos de IoT estão apenas começando (CENEDESE et al., 2014).

O *Padova Smart City* é um exemplo prático da implementação da Internet das Coisas no contexto urbano. Ele foi realizado na cidade de Padova, Itália. Este projeto consiste na utilização de sensores de monitoramento de trânsito, qualidade de ar, medição de temperatura e medição de umidade. Estes sensores são utilizados para capturar dados e disponibilizá-los através de um *Web Service* (CENEDESE et al., 2014).

Outro exemplo de iniciativa é o *Smart Cities for All*⁵ (SC4A), projeto que tem como objetivo definir TICs que sejam acessíveis a pessoas com deficiência e pessoas idosas em *Smart Cities* ao redor do mundo. Ele disponibiliza quatro ferramentas para auxiliar as cidades a atingirem este objetivo social. Estas ferramentas são elaboradas para vários *stakeholders* relacionados a *Smart City*, como políticos, profissionais de TICs, fornecedores de tecnologias, e desenvolvedores de aplicativos e soluções para *Smart City*. Este projeto é apoiado pela Microsoft®.

Em 2015, o Departamento de Transporte dos Estados Unidos (US DOT, 2015) lançou o *Smart City Challenge*⁶, solicitando às cidades de médio porte dos EUA para desenvolverem ideias para um sistema de transporte inovador, integrado, que utilize dados, aplicações e tecnologia para fazer com que o transporte urbano das cidades seja mais rápido, barato e eficiente. O projeto teve 78 cidades participantes, e destas, 7 (sete) foram escolhidas como finalistas. Com estas 7 (sete) ideias, foi criado o relatório *Smart City Challenge: Lessons for Building Cities of the Future*⁷. A Figura 4 mostra a visão de projeto da cidade de

⁵ <http://smarcities4all.org/>

⁶ <https://www.transportation.gov/smartcity>

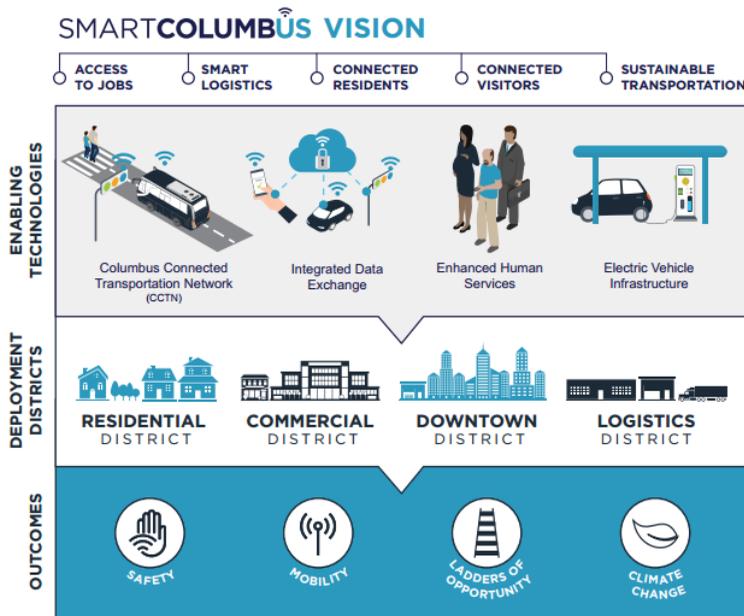
⁷

<https://www.transportation.gov/sites/dot.gov/files/docs/Smart%20City%20Challenge%20Lessons%20Learned.pdf>

Columbus, estado de Ohio nos EUA, vencedora do projeto *Smart City Challenge*.

Figura 4 - Plano da cidade de Columbus (Ohio), vencedora do *Smart City Challenge*

Columbus Smart City Challenge Implementation Vision



Fonte: US DOT (2015)

No projeto de Columbus foram utilizados um conjunto de tecnologias recentes, como infraestrutura conectada (IoT), carregadores de veículos elétricos, uma plataforma de dados interconectada, veículos autônomos, entre outros. Estes projetos começarão a ser implementados em 2017. Um dos projetos é a instalação de uma via inteligente que conectará bairros, que hoje não são atendidos, à empregos e serviços da cidade. Esta via será destinada principalmente ao *Bus Rapid Transit* (BRT), sistema de transporte público via ônibus, que irá contar com um sistema de semáforos inteligentes, informações de trânsito, quiosques de pagamento de passagem, e sinal *Wi-Fi* gratuito por toda a rota.

3 INTERNET DAS COISAS

3.1 INTRODUÇÃO

Com o advento da internet, o mundo vem se transformando um grande sistema interconectado, principalmente quando se leva em consideração a quantidade de informações geradas por dispositivos móveis (*smartphones*, *wearables*, *tablets*). Estes dispositivos se comunicam através de controladores e sensores, visando automatizar processos e gerar serviços para a sociedade (LACERDA; LIMA-MARQUES, 2015). Nesse cenário, a Internet das Coisas, ou *Internet of Things* (IoT), se apresenta como um novo paradigma de comunicação onde vários tipos de objetos estão conectados e gerando dados.

A palavra “coisas” refere-se a objetos cotidianos que podem ser conectados a sensores, de modo a gerar algum tipo de dado. Sendo assim, a IoT objetiva proporcionar novos serviços e produzir bem-estar para as pessoas. Porém, por tratar-se de uma tecnologia recente, os investimentos, ainda que crescentes, são reduzidos devido à falta de conhecimento dos investidores quando se trata sobre o que realmente é este conceito e o que ele pode gerar de valor. Além disso, para aplicá-lo na prática, seria necessário readaptar a internet para suportar esse novo paradigma (ZANELLA et al., 2014).

O crescimento dos smartphones fez com que a proporção de dispositivos conectados à internet por pessoa fosse maior que o número de pessoas no mundo. A Cisco® possui uma expectativa que em 2020 existam aproximadamente 50 bilhões de dispositivos conectados à internet e cerca de 7,6 bilhões de pessoas no mundo, gerando uma proporção de 6,58 dispositivos por pessoa (EVANS, 2011). A Gartner (2013)⁸ é mais otimista, e espera que o número de dispositivos seja de 26 bilhões em 2020. O crescimento do número de dispositivos gera uma demanda da internet como infraestrutura para que todos estes dispositivos trafeguem na rede.

3.2 REDES

Para entender um pouco mais sobre a Internet das Coisas, deve-se entender como funciona a Internet, a qual era conhecida como rede de computadores. Porém, por conta da variação de dispositivos conectados à internet, ela deixou de ser uma rede só para computadores, abrangendo outros dispositivos como: *smartphones*, *tablets*, sensores, transmissores, entre outros dispositivos ou *gadgets* que aparecem no mercado.

⁸ <http://www.gartner.com/newsroom/id/2636073>

A Internet nasceu através do projeto ARPAnet por pesquisadores da ARPA (*Advanced Research Projects Agency*) que lideraram este projeto que visava transmitir dados entre computadores localizados em lugares diferentes. Até então, a ARPA havia projetado uma rede de computadores, porém privada. Apenas na década de 1990 com o fim da ARPA e com a criação da NFSNET houve a popularização da Internet, através da criação de redes públicas (KUROSE; ROSS, 2010).

Segundo Tanenbaum e Wetherall (2013), a Internet como é conhecida hoje não é a mesma de alguns anos atrás. Antes a internet era limitada a utilização de quatro aplicações específicas, as quais são: correio eletrônico, newsgroup, *logon* remoto e transferência de arquivos. O correio eletrônico, também conhecido como *e-mail*, utilizado para redigir, enviar e receber mensagens. *Newsgroup* são fóruns especializados; contêm regras específicas que, caso o usuário desrespeite, este pode ser banido. *Logon* remoto é utilizada através de programas como o *telnet* e sua função é realizar acesso remoto a outras máquinas conectadas à internet. Transferência de arquivos, utilizando um protocolo de transferência, torna possível a cópia de arquivos entre computadores conectados à internet.

A Internet pode ser definida brevemente como: uma rede de dispositivos que se comunicam com roteadores ou comutadores da camada de enlace para enviar dados a determinado destinatário através de um provedor (KUROSE; ROSS, 2010). Estes dispositivos se comunicam entre si através de mensagens que são transmitidas dentro de um protocolo, como por exemplo, o TCP/IP e UDP.

3.2.1 Redes sem Fio

As redes sem fio surgiram em um cenário onde o objetivo era oferecer mobilidade, já que os padrões cabeados não permitiam muita flexibilidade. As redes sem fio funcionam transmitindo ondas de rádio operando sobre uma faixa de frequência e utilizam o ar como meio de propagação (TELECO, 2017). A partir disso, vários padrões de redes sem fio foram criados e padronizados conforme sua abrangência e extensão, entre eles:

- *Wireless Personal Area Network (WPAN)*: Este tipo de rede serve para comunicação entre dispositivos entre uma distância muito pequena, de 10m até 100m, geralmente realizada entre dispositivos móveis. Um dos protocolos mais conhecidos nesse grupo é o *Bluetooth*;

- *Wireless Local Area Network (WLAN)*: Este grupo é conhecido pelo protocolo *Wi-Fi*, amplamente disseminado e popularizado, podendo ser encontrado em lugares públicos, *shoppings*, residências, entre outros lugares. Pode funcionar em um raio de 100m até 300m;
- *Wireless Metropolitan Area Network (WMAN)*: Este grupo é comumente utilizado por provedores, para fornecer acesso para os pontos de distribuição. Utiliza o protocolo WiMax o qual pode transmitir dados de 4km até 10km de distância;
- *Wireless Wire Area Network (WWAN)*: Neste grupo é comum encontrar operadoras de serviços de celulares operando nesse grupo de rede, já que o mesmo pode abranger um país, ou até mesmo um continente;

Uma das principais desvantagens do uso de redes sem fio é a reflexão dos sinais por objetos sólidos. Caso o ponto de acesso esteja em uma área com muitos obstáculos pode ocorrer a perda de pacotes transmitidos (TANENBAUM; WETHERALL, 2013). Para contornar esse problema, as redes realizam a transmissão dos pacotes por vários caminhos diferentes que consigam chegar ao destinatário.

As tecnologias sem fio para uso em redes de projetos de IoT podem ser diversas existindo várias tecnologias que podem ser utilizadas e testadas, como por exemplo: RFID, *Bluetooth*, *Zigbee*, NFC. Cada uma dessas tecnologias tem suas diferenças e são utilizados em contextos diferentes.

Como mostra Zheng (2006), o *Zigbee* é uma tecnologia que tem características como baixo consumo de energia, baixo custo e baixa transferência de dados.

Segundo Madlmayr et al. (2008), o NFC é uma tecnologia sem fio para transferência de dados até 10cm de distância. A maior vantagem do uso do NFC é a simplicidade, já que, dois aparelhos com NFC podem começar uma transação quando dispositivo chegar próximo suficiente do outro dispositivo.

Segundo Juels (2006), o RFID é uma tecnologia para automatizar a identificação de objetos e pessoas. Um dispositivo com RFID é composto por um microchip para transmissão de dados por rádio frequência.

A próxima seção aborda em mais detalhes a tecnologia *Bluetooth* já que esta tecnologia promove suporte para este trabalho.

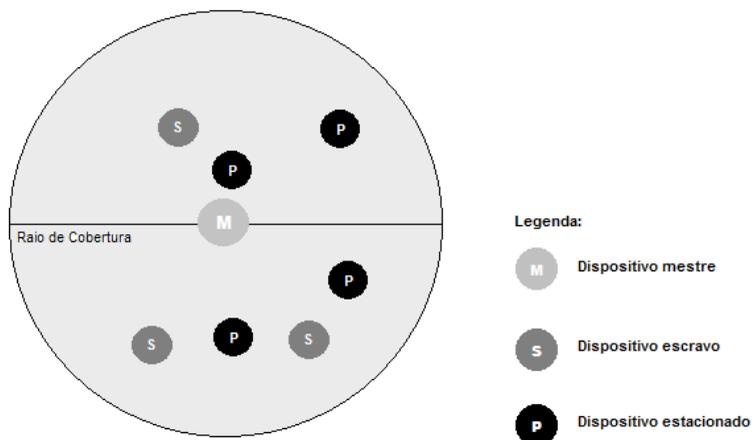
3.2.2 Bluetooth

O *Bluetooth* nasceu como uma alternativa para transferência de dados a curta distância, geralmente realizado entre sistemas finais e periféricos via ondas de rádio pela frequência 2.4 Ghz, sendo normatizado pelo padrão IEEE 802.15.1. Por se tratar de uma rede *ad hoc* não é necessário um ponto de acesso para a conexão dos dispositivos. A rede funciona de forma mestre-escravo (Figura 5), onde um nó é o mestre que comanda a rede, enquanto os nós escravos podem transmitir para o nó mestre apenas quando o mesmo tiver realizado uma transmissão ao nó escravo anteriormente (KUROSE; ROSS, 2010).

Desde o lançamento do *bluetooth* até atualmente, houve uma série de atualizações. A partir da versão 4 em diante foi apresentado o *Bluetooth Low Energy* (BLE).

Segundo Chang (2014), o BLE foi introduzido em 2010 com o objetivo de aumentar o número de aplicações baseadas em *Bluetooth*, visando utilizar em dispositivos que tivessem energia limitada, tais como: sensores sem fio e controladores sem fio. O maior diferencial do BLE para o *Bluetooth Classic* é que não há mais uma interação intensa com um grande fluxo de dados entre dois dispositivos, ou seja, na versão BLE as interações são rápidas e há um baixo fluxo de dados fazendo com que seja necessária menos energia do que a versão anterior do protocolo.

Outra mudança que houve com a chegada do BLE, é que agora existem 40 canais com o dobro de largura de banda comparado ao *Bluetooth Classic*. Três desses canais são utilizados para detecção e iniciação da comunicação com outros dispositivos, e os 37 restantes são utilizados para transmissão de dados. Outra característica é de que neste protocolo os pacotes de transmissão enviados por um dispositivo *bluetooth low energy*, variam entre 10 a 47 bytes. Sendo assim, o tempo de transmissão dos menores pacotes pode levar 80 microssegundos e dos maiores podem levar 300 microssegundos.

Figura 5 - Rede *bluetooth*

Fonte: Adaptado de Kurose e Ross (2010)

Segundo Kwon et al. (2016), o *bluetooth low energy* na versão 4.2 conta com um gerenciador de segurança que utiliza chaves para autenticar os dispositivos em que o sistema de segurança pode ser ativo ou não. São usadas as seguintes chaves:

- *Temporary Key (TK)* é uma chave de 128 bits que é utilizada para criar a *STK* durante o pareamento.
- *Short Term Key (STK)* é a chave utilizada para criptografar a conexão.
- *Long Term Key (LTK)* é utilizada para distribuição de chaves para reuso em uma conexão.

Como mostra Kwon et al. (2016), a autenticação pode ser utilizada ou não para conectar os dispositivos *bluetooth*. Existem basicamente três modos de autenticação, os quais são: *Just Works*, *Passkey entry*, *out of band*. O modo *Just Works* é utilizado para dispositivos que ao serem conectados devem estar se comunicando automaticamente, como *headphones* sem fio. O modo *Passkey entry* gera uma chave de seis dígitos, e o dispositivo o qual irá se conectar deve então informar essa chave para que haja a comunicação. Já o modo *out of band*, não utiliza a interface *bluetooth* para a autenticação. Consegue ser mais seguro que os dois modos anteriores, porém requer que os dispositivos tenham a mesma interface para se comunicarem.

Durante o pareamento é utilizado uma TK para gerar a STK. Com a STK gerada, a conexão se torna criptografada e segura. Porém, caso a conexão seja interceptada durante a geração da STK, o interceptador pode alterar os valores da TK e gerar uma STK, fazendo com que o interceptador seja capaz de descriptografar os dados que serão transmitidos durante a sessão. Isso ocorre, pois, o tamanho da TK é pequena, e pode-se descobrir seu valor em um ataque de força bruta, onde-se testa os valores até descobrir a chave.

Segundo Ray e Agarwal (2016), o *Bluetooth* versão 5 trará novas melhorias com foco na eficiência para projetos voltados à IoT. A seguir são apresentadas algumas melhorias:

- A transmissão de dados terá mais banda, além de ser duas vezes mais rápida que a versão anterior.
- O alcance do *Bluetooth* aumentará cerca de quatro vezes mais.
- Não será necessário aplicativos para tratar informações baseadas na localização, pois com o aumento da capacidade da mensagem que *broadcast*, pode-se enviar informações personalizadas.

3.3 INTERNET DAS COISAS E O IMPACTO SOCIAL

A tecnologia em geral sempre esteve associada a possíveis mudanças positivas objetivando o bem-estar social. Atualmente, as pessoas almejam uma vida com mais bem-estar e saúde, e isto é um aspecto que a IoT pode proporcionar para a sociedade. A internet por si só, já se provou como uma revolução e vem se constituindo em dos fatores principais para a globalização. A informação sempre teve um grande valor para humanidade, desde a descoberta do fogo até os dias atuais. A IoT pretende expandir esse leque de funcionalidades fazendo com que as pessoas utilizem melhor os recursos providos por esta (EVANS, 2011).

Com a IoT, pode-se realizar coisas que ainda não são imagináveis, principalmente, pela abrangência de áreas que ela contempla. Por exemplo, a Internet das Coisas oferece serviços como o rastreamento de gado, o cálculo do custo da água consumida em diferentes áreas de uma cidade, ou ainda o monitoramento de sinais vitais de pessoas idosas (EVANS, 2011). E com essa nova perspectiva é possível criar novos hábitos, criar novas facilidades e novos serviços, ou ainda tornar mais eficientes serviços que já existam.

Há tempos se vem discutindo como seriam as casas inteligentes, ainda quando computadores não eram considerados itens pessoais. Mais recentemente, algumas empresas passaram a vender seus pacotes prontos para *smart houses*, como a Google® e a Apple®. Mas quais os serviços que realmente teriam valor para as pessoas que tivessem uma casa inteligente? Algumas aplicações que poderão agregar valor quanto a isso podem oferecer serviços como: a) usar energia de maneira mais eficiente; b) proteger a casa e as pessoas que nela vivem; c) ter autonomia de poder limpar a casa sozinha; e d) oferecer mais entretenimento (DECUIR, 2015).

3.4 IMPASSES PARA IMPLANTAÇÃO DA INTERNET DAS COISAS

Como um novo paradigma para comunicação atual, a IoT pretende estender muito mais a internet, e fazer com que tudo esteja conectado. Com isso, pode-se trazer mais soluções, mas também por outro lado pode-se trazer ainda mais problemas se não administrado e arquitetado para que esse novo modelo seja de fato eficiente. A seguir são citados e discutidos alguns desses desafios.

3.4.1 Escalabilidade

Segundo Gupta e Shukla (2016), os dispositivos serão implementados em larga escala, e deve-se então fornecer a estes autenticação, manutenção, proteção, utilidade e suporte desses dispositivos em larga escala são o maior problema. Outra questão levantada foi de que os dados gerados, armazenamento e uso podem ser um problema quanto à escalabilidade.

Evans (2011) diz que uma das mudanças primordiais é a transição do IPv4 para IPv6, já que para a implantação da IoT será necessária a alocação de mais endereços para suportar todos os dispositivos. Esta transição está sendo realizada, mas ainda não está completa.

3.4.2. Big data e dados gerados

Os dados que serão transmitidos devem apenas trafegar pela rede se forem realmente dados que poderão se tornar conhecimento válido. Dados primários sem uso não devem utilizar a rede, a fim de torná-la mais eficiente e não gastar recursos com dados que serão inutilizados (GUPTA; SHUKLA, 2016).

3.4.3 Segurança

Segurança é questão primordial quanto a IoT. Neste modelo os dispositivos são mais vulneráveis por diversas razões, como por exemplo: é uma rede heterogênea; contém múltiplos protocolos de comunicação, dispositivos e plataformas; não há nada ainda unificado para esse propósito (BERTINO; ISLAM, 2017). Outra questão dentro da segurança é a autenticação por dispositivo, a qual não seria possível, além da dificuldade que será fornecer trilhares de chaves para todos eles.

3.4.4 Privacidade

A privacidade se diz respeito a todas as formas de manipulação da informação na rede, por poder conter informações privadas. Os dados não podem ser vistos por outras pessoas, além do proprietário da informação. Nesse contexto, ainda há falta de políticas de privacidade para cada sistema, o que poderia ajudar a resolver este impasse (ELBOUANANI; KIRAM; ACHBAROU, 2015).

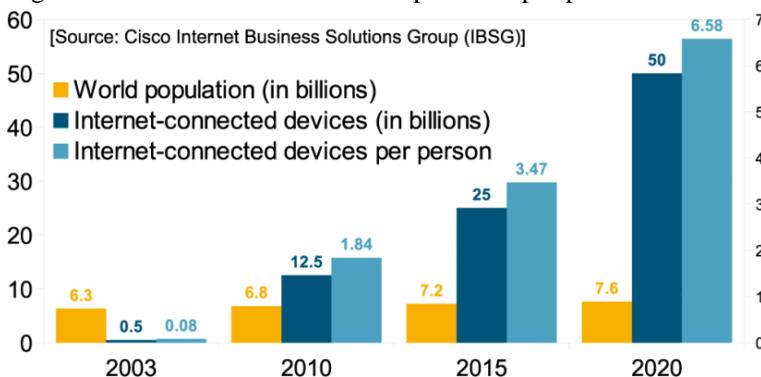
3.5 INTERNET DAS COISAS E O MERCADO

3.5.1 Crescimento da Internet das Coisas

O crescimento da IoT ocorreu pouco após o surgimento do termo, e ganhou atenção de diferentes setores como, indústria, instituições acadêmicas, *startups*, organizações governamentais e das grandes corporações. Segundo Perera, Liu e Jayawardena (2014), a IoT visa a melhoria da vida das pessoas, podendo fazer com que estas possam estar conectadas por todo tempo, em qualquer lugar, a qualquer hora.

O crescimento do número de dispositivos que utilizam a internet no mundo aumentou exponencialmente. Segundo Evans (2011), estima-se que em 2020 existiram cerca de 6,5 dispositivos com internet por pessoa no mundo (Figura 6). Somente o mercado de sensores em 2010 obteve cerca de 56,3 bilhões de dólares e este número tende a aumentar.

Figura 6 – Gráfico de número de dispositivos por pessoa



Fonte: Extraído de Evans (2011)

3.5.2 O Valor da Informação

A informação sempre foi algo de muito valor para a sociedade. A informação tem um papel fundamental, e dependendo da tomada de decisão pode ser valiosa. A Google®, uma das maiores empresas da área de tecnologia, tem usado a informação com seu serviço de busca com a intenção de direcionar a publicidade para uma pessoa em específico, o que vem gerando uma boa parte da receita da empresa (BBC, 2017).

A internet das coisas demandará um fluxo maior de dados trafegando na internet, e será responsabilidade dos sistemas utilizar estes dados para criar informação que possua algum valor no mercado. Há uma área que estuda as questões do valor da informação, nomeada economia da informação. Esta área se dedica ao estudo da economia da informação, onde o valor da informação é provido conforme o custo para obter a informação e o valor da informação gerada, com o objetivo de tirar o máximo de proveito para seu uso (NIYATO et al., 2016).

Quantificar o valor da informação requer estudo, pois a informação é tratada como um bem intangível, sendo difícil de mensurar uma vez que o seu valor pode variar consideravelmente dependendo do uso e utilidade. Algumas questões a serem abordadas quanto o valor da informação segundo Niyato et al. (2016) são:

- **Qualidade da informação:** A qualidade da informação é algo amplo, e depende da confiabilidade e da qualidade da

informação mais do que a quantidade. Depende também de como ela é utilizada para uma tomada de decisão;

- **Estrutura de custos diferentes:** O custo para se desenvolver um projeto que forneça informação como produto são maiores em seus estágios de *design*, desenvolvimento e o de implantação de um sistema do que o custo de armazenamento. Por exemplo, fazer implantação de sensores e uma infraestrutura de comunicação requer um grande investimento;
- **Opções de preços:** Na IoT, existem modelos de custo conforme os dados gerados e os serviços utilizados. No modo de preço por transação o cliente paga pelo acesso às informações. O modo de preço por informação/tempo cobra pela quantidade de informação utilizada ou pelo tempo. E o modo de preço por inscrição, a cobrança é realizada pelo acesso à informação e aos serviços por um determinado período de tempo. Cada um dos modos de cobrança tem alguma área específica onde pode ser melhor utilizada;
- **Lucro e o custo:** Para ter uma melhor eficiência busca-se sempre ter um lucro maior do que o custo para o uso da informação. Para isso devem ser pensadas questões quando se implementa um sistema de IoT, tais como, a amplitude do projeto do sistema, a quantidade de dispositivos que serão utilizados, a otimização da energia e dos recursos utilizados;
- **Preço de competitividade:** É comum ter várias empresas de IoT fornecendo o mesmo tipo de serviço e isso pode impactar no valor final da informação. Para manter-se no mercado a empresa deve sempre procurar outras formas de diminuir o custo da estrutura, bem como, definir estratégias para a venda da informação.

4 SISTEMA PROPOSTO

4.1 INTRODUÇÃO

A proposta deste trabalho consiste em desenvolver um sistema (nível de protótipo) para dispositivos móveis, voltado ao transporte público urbano, baseando-se nos conceitos apresentados neste trabalho (*Smart City* e IoT)⁹. Estes conceitos representam a utilização de sensores, integrados ao sistema desenvolvido, para possibilitar uma interação mais inteligente com o transporte via ônibus, proporcionando comodidade e informações em tempo real para o usuário.

O tipo de sensor escolhido para o desenvolvimento do sistema foi o *beacon*, um dispositivo *Bluetooth Low Energy* (BLE) que serve como um ponto de referência espacial para o sistema. O *beacon* não é pareado com o dispositivo, como a maioria dos dispositivos *bluetooth*. Ele apenas emite o sinal e se comunica utilizando um protocolo específico. Todo o comportamento depende do sistema que captura seu sinal e utiliza as informações que ele disponibiliza através do seu protocolo, como por exemplo, a força do sinal, que é utilizada na medição de distância. A principal justificativa para o uso do *beacon* é justamente o fato de ser possível identificar a distância (com precisão de centímetros) entre o sensor e o dispositivo que está captando seu sinal.

O sistema se baseia em um cenário no qual já exista a presença de *beacon* espalhados pela cidade, mais especificamente um em cada ponto de ônibus, e um no próprio ônibus. Pelo fato de haver sensores nas paradas e no ônibus, pode-se obter algumas informações úteis quando detectados, tais como:

- Tempo entre uma parada e outra. Dessa forma, é possível fornecer em tempo real, através de um servidor *online*, quanto tempo um ônibus está levando de uma parada até a próxima;
- Enviar notificação para o usuário assim que ele se aproximar da parada, enviando os horários de seus ônibus favoritos, economizando tempo e oferecendo praticidade e comodidade para o mesmo;
- Qualquer outro tipo de informação que seja útil para o usuário.

⁹ O código fonte completo do sistema desenvolvido neste trabalho se encontra no repositório Github:

<https://github.com/dicaetano/TCC/tree/master/MobileBus>

4.2 REQUISITOS

Para o desenvolvimento do sistema proposto foi necessário realizar uma análise das necessidades de *hardware* e *software* para o funcionamento do mesmo. Por *hardware* entende-se como todo o conjunto de tecnologias físicas necessárias para atender os requisitos da proposta (sensor e o *smartphone* do usuário), e *software* sendo o próprio sistema desenvolvido.

4.2.1 Requisitos funcionais

Os requisitos funcionais são todos aqueles que dizem o que o *software* ou *hardware* deve ser capaz de fazer. Eles definem o comportamento do sistema. Neste trabalho, como existe uma forte dependência do sistema com o *hardware* utilizado, foram analisados tanto os requisitos funcionais do *software* (Quadro 3) quanto do *hardware* (Quadro 4).

Quadro 3 - Requisitos funcionais de software

Código do requisito	Descrição
RF001	Ler sensores próximos.
RF002	Identificar sensores que estão nos ônibus, para diferenciar dos sensores das paradas.
RF003	Calcular a distância entre sensor e <i>smartphone</i> , para determinar se usuário entrou no ônibus.
RF004	Cadastrar sensor.
RF005	Cadastrar paradas de ônibus.
RF006	Cadastrar linha de ônibus.
RF007	Cadastrar horários de ônibus.
RF008	Listar paradas de ônibus.
RF009	Listar linhas de ônibus.
RF010	Enviar notificação para o usuário quando se aproximar de um sensor na parada de ônibus. Quando o usuário acessar a notificação, devem ser apresentados os próximos ônibus que passarão naquela parada específica, com preferência às linhas favoritas do usuário.
RF011	Apresentar o tempo médio das rotas de acordo com

	informações dos cálculos de tempo realizados através dos dados obtidos a partir dos outros usuários.
--	--

Fonte: Autores

Quadro 4 - Requisitos funcionais de *hardware*

Código do requisito	Descrição
RF001	O sensor deve ser capaz de ser detectado a partir de um <i>smartphone</i> comum.
RF002	O sensor deve ser capaz de disponibilizar a distância entre ele e o <i>smartphone</i> do usuário.
RF003	O <i>smartphone</i> do usuário deve ter a capacidade de se comunicar com o sensor.

Fonte: Autores

4.2.2 Requisitos não funcionais

Os requisitos não funcionais dizem como o *software* ou *hardware* deve funcionar. Tais requisitos podem ser referentes ao desempenho, interface, usabilidade, qualidade, entre outros. Os quadros Quadro 5 e Quadro 6 representam os requisitos não funcionais de *software* e *hardware*, respectivamente.

Quadro 5 - Requisitos não funcionais de *software*

Código do requisito	Descrição
RN001	Sistema deve ter um bom desempenho com relação ao tempo de conexão com o servidor de banco de dados
RN002	Sistema deve ter uma interface atraente para o usuário

Fonte: Autores

Quadro 6 - Requisitos não funcionais de *hardware*

Código do requisito	Descrição
RN001	Sensor deve ter baixo custo de produção e alta disponibilidade, ou seja, sua bateria deve ter alto tempo de vida
RN002	Sensor deve a capacidade de ser instalado em ambiente ao ar livre, e resistir à imprevisibilidade do clima

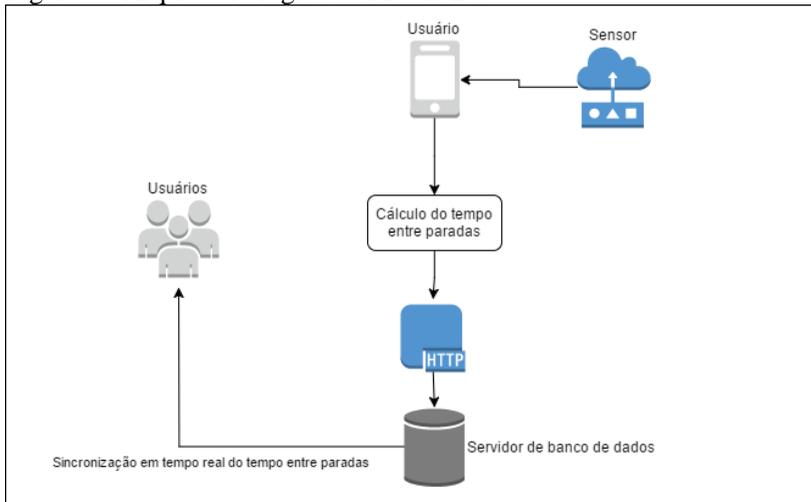
Fonte: Autores

4.3 ARQUITETURA

4.3.1 Arquitetura lógica

Na Figura 7 é apresentada a arquitetura lógica do sistema (denominado MobileBus), que representa um modelo para captura e persistência dos tempos entre as paradas de ônibus que são capturados pelo sistema através dos sensores. O fluxo começa com o componente “Sensor”, que representa um *beacon*. Este sensor fica continuamente emitindo um sinal. Quando o sistema captura esse sinal, é realizado o cálculo entre a parada (sensor) anterior e a atual, sendo o dado então enviado ao servidor de banco de dados utilizando uma camada HTTP. Assim que o servidor recebe esse dado, processa e atualiza sua base, mantendo sempre o tempo entre as paradas atualizado para os demais usuários.

Figura 7 - Arquitetura lógica do MobileBus



Fonte: Autores

Esta arquitetura é dependente dos usuários. De modo geral, quanto mais o sistema é utilizado, maior será a precisão que este poderá atingir.

4.3.2 Arquitetura física

Para a implementação do sistema proposto foram necessárias pesquisas sobre as tecnologias disponíveis para manipulação e persistência de dados em aplicações Delphi®. Esta IDE foi escolhida pela experiência na linguagem *Object Pascal*, e principalmente por permitir o desenvolvimento de sistemas multi-plataforma. A plataforma de desenvolvimento escolhida foi o Android®, pelo motivo de atualmente estar presente em uma parcela considerável de dispositivos pessoais, bem como, pela experiência nesta plataforma.

Como resultado da pesquisa foi encontrado o Aurelius®, um *framework Object-Relation Mapping* (ORM) distribuído pela TMS Software® para aplicações Delphi®. Com o Aurelius® foi possível evitar a preocupação em construir e manipular instruções SQL. Ao invés disso é trabalhado apenas com classes e objetos. Existem várias vantagens em utilizar um ORM como o Aurelius®, tais como:

- Facilidade na manutenção: a lógica do negócio fica mais clara lidando apenas com objetos, escondendo a camada de dados;
- Portabilidade: os objetos e sua utilização são independentes do banco de dados utilizado, facilitando futuras mudanças do banco de dados;

Na Figura 8 é demonstrado um código simples escrito em Delphi® onde são realizadas operações SQL explícitas. Códigos escritos dessa forma são difíceis de manter, porém muito comuns. Nele, é realizada uma instrução SQL, utilizando um componente específico para esta operação, e inserido o número “1” como parâmetro do campo INVOICE_ID. Depois de executar (método “Open”) é apresentada uma mensagem com alguns valores retornados pelo banco.

Figura 8 - Código em Delphi utilizando comandos SQL explícitos

```
Query1.Sql.Text := 'SELECT I.ID AS INVOICE_ID, I.INVOICE_TYPE, I.INVOICE_NO, I.ISSUE_DATE, I.PRINT_DATE, ' +
  'C.ID AS CUSTOMER_ID, C.CUSTOMER_NAME, C.SEX, C.BIRTHDAY, N.ID AS COUNTRY_ID, N.COUNTRY_NAME' +
  'FROM INVOICE AS I INNER JOIN CUSTOMER AS C ON (C.ID = I.CUSTOMER_ID)' +
  'LEFT JOIN COUNTRY AS N ON (N.ID = C.COUNTRY_ID)' +
  'WHERE I.ID = :INVOICE_ID;';
Query1.ParamByName('INVOICE_ID').AsInteger := 1;
Query1.Open;
ShowMessage(Format('Invoice No: %d, Customer: %s, Country: %s',
  [Query1.FieldByName('INVOICE_NO').AsInteger,
  Query1.FieldByName('CUSTOMER_NAME').AsString,
  Query1.FieldByName('COUNTRY_NAME').AsString]));
```

Fonte: Extraído de TMS Aurelius (2017)

Utilizando o *framework* Aurelius, o código acima é traduzido para o que se encontra na Figura 9. A classe TInvoice representa a entidade relacional INVOICE (a mesma do comando SQL da 9), e possui propriedades para cada um dos campos que existem nesta tabela. Por exemplo, na definição da classe TInvoice existe uma propriedade do tipo Integer que é vinculada à chave primária INVOICE_ID. Todas as vinculações entre campos da tabela relacional e as propriedades de classe são feitas através de atributos¹⁰.

Figura 9 - Código em Delphi utilizando o componente Aurelius

```
Invoice := Manager1.Find<TInvoice>(1);
ShowMessage(Format('Invoice No: %d, Customer: %s, Country: %s',
  [Invoice.InvoiceNo, Invoice.Customer.Name, Invoice.Customer.Country.Name]));
```

Fonte: Extraído de TMS Aurelius (2017)

No código acima, o objeto “Invoice” recebe uma instância retornada pelo objeto Manager1 (instância de TObjectManager¹¹). O método “Find” irá procurar por um registro na tabela INVOICE cuja chave primária seja 1 (passada como parâmetro). Todos os campos desta tabela ficam disponíveis para o objeto através de suas propriedades. Pode-se notar que o Aurelius® também suporta a associação de objetos. Por exemplo, na terceira linha, onde é acessado a propriedade “Invoice.Customer.Name” na tabela relacional, existe uma chave estrangeira referenciando outra tabela CUSTUMER, sendo que esta possui um campo chamado NAME.

Este *framework* suporta todos os componentes de acesso a banco de dados disponível no Delphi®, tais como: FireDAC, dbExpress, IBX, entre outros. Estes componentes fazem a ligação entre o banco de dados e o Delphi®. O Aurelius® é uma camada extra entre a aplicação e estes componentes de conexão. Para fazer a conexão entre a aplicação e todos estes tipos diferentes de componentes é utilizada uma *interface* chamada

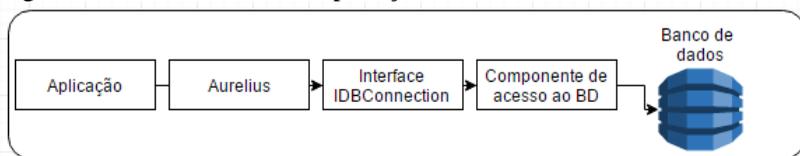
¹⁰ Nas versões mais recentes do Delphi®, existe um recurso chamado de atributos. Este recurso é semelhante às anotações em Java®. Basicamente refere-se a possibilidade de inserir metadados em classes e propriedades de classe. Estes metadados podem ser recuperados em tempo de execução utilizando RTTI (*Runtime Type Information*).

¹¹ A classe TObjectManager faz parte da biblioteca do Aurelius®, e é utilizada para manipular os objetos. A letra “T” antes da definição de tipos (classes, *enumerators*, etc.) é uma convenção no desenvolvimento em *Object Pascal*.

IDBConnection. Utilizando esta *interface*, o tipo de servidor de banco de dados escolhido e componente de conexão são transparentes para a aplicação, podendo ser alterado a qualquer momento sem qualquer modificação no código fonte.

A Figura 10 mostra as camadas entre a aplicação e o banco de dados. A interface IDBConnection implementa métodos que independem do componente de acesso a dados utilizado. O componente de acesso ao banco utilizado foi o FireDAC, por ser nativo do Delphi®.

Figura 10 - Camadas entre a aplicação e o banco de dados



Fonte: Autores

O Aurelius® é multiplataforma, podendo ser utilizado no desenvolvimento de sistemas Windows®, MacOS® e Android®. Além disso, ele suporta grande parte dos servidores de banco de dados disponíveis atualmente, tais como, Firebird®, Interbase®, Microsoft SQL Server®, MySQL®, NexusDB®, Oracle®, SQLite®, PostgreSQL®, DB2®, Elevate DB Server® e Absolute Database®;

Para o desenvolvimento do sistema proposto, foi utilizado o SQLite, pela sua simplicidade e por ser nativo na plataforma Android®.

Outro recurso do Aurelius® utilizado neste trabalho foi a geração automática da estrutura do banco de dados. Significa dizer que foi apenas realizada a modelagem das classes e o mapeamento. A efetiva criação de tabelas e campos foi realizada automaticamente pelo *framework*.

A Figura 11 demonstra como funciona o mapeamento. Ao indicar o atributo “[Automapping]” acima da declaração da classe, esta será mapeada automaticamente. Neste exemplo, a classe TBusExitTime representará uma entidade no banco de dados, e seu nome será gerado automaticamente como “BUS_EXIT_TIME”. Quando se tem uma propriedade com nome “ID” na classe, ela é automaticamente criada como chave primária no banco. No caso do *enumerator* TWeekDay (propriedade “WeekDay” da classe TBusExitTime), será criado um campo “WEEK_DAY” na tabela “BUS_EXIT_TIME”, Sempre que for inserido ou atualizado este campo em um registro, será automaticamente

convertido para o tipo Integer. Isso quer dizer que se inserir um registro na tabela com o campo `WeekDay` como “`wdSunday`”, será inserido o valor “0”, e assim por diante com os outros valores. Todas estas gerações automáticas podem ser desativadas, tendo a possibilidade de fazer o mapeamento todo manualmente. Neste trabalho, foi utilizado este recurso, e todos os campos e entidades foram autogerados.

Figura 11 - Classe `TBusExitTime` implementada no sistema, exemplificando o mapeamento da classe

```

type
  [Automapping]
  TWeekDay = (wdSunday, wdMonday, wdTuesday, wdWednesday,
    wdThursday, wdFriday, wdSaturday);

  [Entity]
  [Automapping]
  TBusExitTime = class
  private
    FWeekDay: TWeekDay;
    FID: Integer;
    FExitTime: string;
    FBusLine: TBusLine;
  public
    property ID: Integer read FID;
    property BusLine: TBusLine read FBusLine;
    property ExitTime: string read FExitTime;
    property WeekDay: TWeekDay read FWeekDay;
  end;

```

Fonte: Autores

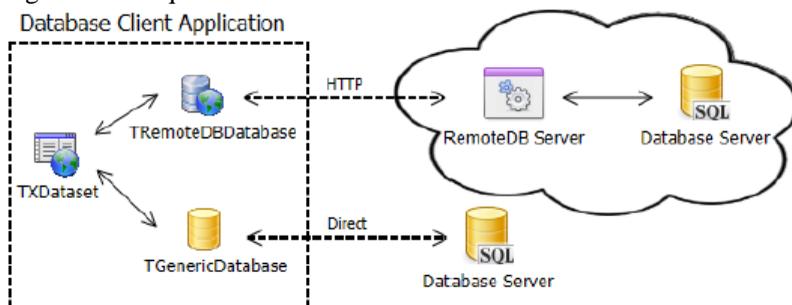
Além da pesquisa sobre o *framework* ORM, foi realizada uma pesquisa visando prover de maneira remota o servidor de banco de dados. Para tal, foi encontrado o RemoteDB®, um componente para a construção e acesso a servidores de banco de dados via HTTP. Este componente também é implementado e distribuído pela TMS Software®, sendo integrado com o *framework* Aurelius®.

Com o RemoteDB® é possível criar um servidor HTTP local ou remoto. Este servidor faz a ligação entre o banco de dados e a aplicação cliente, processando requisições HTTP e retornando os resultados do banco de dados. Desta forma, é possível separar o desenvolvimento da

aplicação do banco de dados e a aplicação cliente, permitindo a aplicação em multicamadas¹².

A Figura 12 mostra a arquitetura do RemoteDB®. No quadro pontilhado ao lado esquerdo da imagem reside a aplicação cliente e três componentes: TXDataset, TRemoteDBDatabase e TGenericDatabase. Eles são utilizados para fazer as requisições ao servidor, receber a resposta, e disponibilizar os dados recebidos. A parte da direita representa o servidor HTTP. O RemoteDB Server se conecta com algum componente de acesso a dados disponível no Delphi®. Para isso foi utilizado, como já mencionado, o FireDAC.

Figura 12 - Arquitetura do RemoteDB



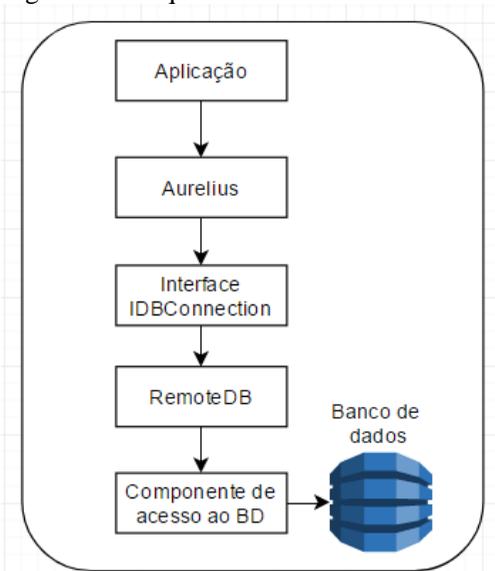
Fonte: Extraído de TMS RemoteDB (2017)

O componente TXDataset implementa a classe TDataset, que provê métodos, propriedades e eventos básicos para manipulação de dados no Delphi®. Ele pode ser utilizado com o TGenericDatabase, que faz a conexão com um servidor local (mesmo dispositivo que está rodando a aplicação), ou com o TRemoteDBDatabase, que faz a conexão com um servidor HTTP (na Internet ou em uma rede local). Este último foi utilizado no desenvolvimento do sistema proposto.

Com o TRemoteDBDatabase é possível utilizar a *interface* IDBConnection para a conexão, a mesma que é utilizado no Aurelius®. Desta forma, a estrutura de dados será constituída como apresentada na Figura 13.

¹² Uma aplicação multicamadas separa o desenvolvimento da *interface* da aplicação, regras de negócio e gerenciamento do banco de dados. Neste conceito, cada um dos itens é uma camada independente. A camada do meio, referente às regras de negócio pode ter ainda outras camadas.

Figura 13 - Arquitetura de dados



Fonte: Autores

4.3.3 Sensor sem fio

A escolha do sensor sem fio para o desenvolvimento do sistema proposto foi baseada nos requisitos funcionais e não funcionais de *hardware*, apresentados nas seções anteriores. Como mencionado, o sensor deve ter a capacidade de identificar o usuário na parada de ônibus (utilizando seu *smartphone*), deve ser resistente ao clima, entre outros. Depois de analisado os requisitos, foi concluído que o sensor que mais se encaixaria no perfil necessário é o *beacon*.

Este sensor foi lançado inicialmente em 2013 pela Apple®, intitulado como *iBeacon*, e lançado mais tarde por outras empresas, passando a ser chamado apenas de *beacon*. O sensor é baseado na tecnologia *bluetooth*, e por esse motivo a interação com os usuários pode ser facilitado visto que todos os celulares atualmente possuem conexão *bluetooth*. A tradução literal da palavra “*beacon*” significa “farol”, e é exatamente dessa forma que ele funciona. O sensor emite um sinal constante, que pode ser identificado por aplicativos em *smartphones*.

A maior vantagem deste tipo de sensor é a possibilidade de identificar a distância em relação ao celular do usuário com precisão de

centímetros. Dessa forma, a aplicação pode saber se o usuário está perto ou longe do *beacon*. Para o desenvolvimento do MobileBus isso foi essencial, pois era necessário saber se o usuário passou por uma determinada parada, ou se entrou em um ônibus. Este segundo objetivo poderia ter sido realizado de maneira mais adequada com o uso de sensores como o NFC (*Near Field Communication*), porém o presente trabalho focou apenas na utilização de *beacons*.

Para a identificação de diferentes *beacons*, cada um possui um UUDI, ou seja, um identificador único com 32 dígitos hexadecimais. Este identificador é dividido em 5 grupos. A primeira seção possui 8 dígitos, a segunda 4, a terceira 4, a quarta 4 e a quinta seção 12 dígitos. Quem define o identificador é o fabricante do sensor.

Além do UUID, o beacon possui outros dois identificadores: o *Major ID* e *Minor ID*. Estes números podem variar entre 0 e 65535, e servem para identificar os *beacons* com maior precisão do que apenas com o UUDI. O *Major ID* identifica um grupo, e o *Minor ID* identifica um *beacon* de um determinado grupo. Exemplificando em um cenário, se um museu possui mil *beacons*, e possui 5 exposições (uma sobre antiguidades, outra sobre obras de arte, etc.), com cada apresentação tendo 200 exposições, o *Major ID* dos *beacons* de cada apresentação teriam um valor entre 1 e 5, e o *Minor ID* de 1 a 200. No desenvolvimento do MobileBus não foi utilizado este conceito. Porém, na aplicação real do sistema, o *Major ID* poderia ser utilizado para dividir os *beacons* por bairro, enquanto o outro identificaria o sensor em cada bairro.

O tempo de duração da bateria de um *beacon* varia entre 18 e 24 meses. Isso ocorre em função do sinal *bluetooth* ser considerando *low energy*. Juntando isso com o fato de seu custo ser baixo, aproximadamente R\$50,00 (ou \$15,00), faz com que ele se torne muito útil em inúmeras situações.

A principal informação que um *beacon* pode fornecer é a distância entre ele e o dispositivo que o descobriu. Como já mencionado anteriormente, pode-se saber a distância com precisão de centímetros. O cálculo da distância é realizado utilizando a força do sinal transmitido (*txPower*, ou *Measured Power*) e o RSSI (*Received Signal Strength Indicator*). O *beacon* apenas transmite essas informações, o cálculo fica por conta da aplicação. Normalmente, é utilizada uma biblioteca implementada especificamente para isso. No caso deste trabalho, como já mencionado, foi utilizada a biblioteca disponível no Delphi®.

O valor de *txPower* é a força do sinal esperada a 1m de distância do *beacon*. Este valor é normalmente configurado pelo fabricante. Se o

smartphone reconhece que o valor da força do sinal recebido RSSI pelo *beacon* é igual ao valor do *txPower*, ele sabe que o *beacon* está a 1m de distância. Se o valor é maior, está mais perto, caso contrário mais longe. É importante, antes do uso real do *beacon*, calibrar esse valor. Isso porque as ondas podem sofrer interferências externas (reflexão, refração, difração). Se o *beacon* for instalado dentro de uma sala, ou atrás de uma parede, seu sinal enfraquece. Se for colocado atrás de uma parede de metal, talvez o sinal se fortaleça (reflexão do sinal). Por isso é recomendado que se realize a calibração do *txPower* depois do *beacon* instalado.

Alguns fabricantes permitem a alteração da força do sinal do *beacon*. Por força do sinal, entende-se o alcance do sinal, ou seja, o quão longe o sinal do *beacon* pode chegar. Quanto maior a força do sinal, maior o gasto de bateria do dispositivo. É importante ressaltar que não seja confundido a força do sinal (*transmit power*) com a força do sinal transmitido (*transmitted power*, ou *txPower*).

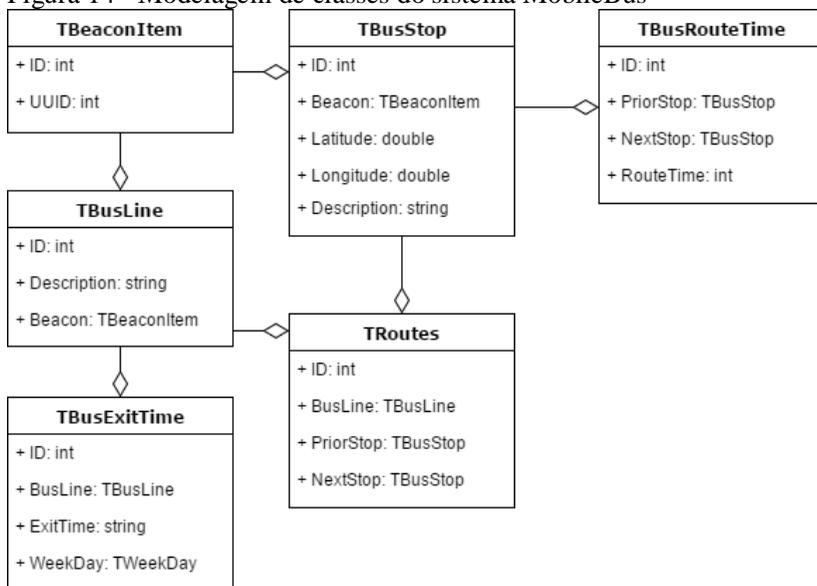
O RSSI indica a força do sinal no dispositivo receptor, que no caso deste trabalho é representado pelo *smartphone* do usuário. Ele depende da distância entre os dois e do valor *txPower* do *beacon*.

4.3.4 Modelo de dados e classes

Como já mencionado anteriormente, a modelagem da estrutura de dados do banco foi realizada automaticamente por meio do *framework* Aurelius®. Será apresentado aqui a modelagem de classes do sistema MobileBus e como o Aurelius® estruturou estas classes no banco de dados.

Na Figura 14 é apresentado o modelo das principais classes do sistema. Para cada uma destas classes existirá uma tabela correspondente no banco de dados, e para cada interação entre elas existe um relacionamento (chave estrangeira) com o ID da mesma.

Figura 14 - Modelagem de classes do sistema MobileBus



Fonte: Autores

Para cada uma destas classes foi criada uma outra classe controladora. Por exemplo, para a classe TBeaconItem, existe a classe TBeaconItemController. Nas classes controladoras existem métodos para salvar, eliminar e selecionar registros da entidade correspondente. Por este motivo, as classes aqui apresentadas não possuem métodos, nem mesmo *gets* e *sets*. Foi preferível implementar dessa forma para se manter a simplicidade no sistema. Abaixo segue as considerações de cada classe:

- TBeaconItem – esta classe mantém o cadastro do *beacon*. Ela guarda apenas um ID para o registro no banco e o UUID do dispositivo, para a identificação posterior.
- TBusStop – esta classe mantém o cadastro da parada de ônibus. Ela possui um ID para identificação no banco, uma propriedade Beacon, do tipo TBeaconItem, que guarda o *beacon* vinculado a esta parada, bem como, a latitude e a longitude para saber onde mostrá-la no mapa e uma descrição.
- TBusRouteTime – esta classe mantém o tempo de um ônibus entre duas paradas. Possui um ID, as propriedades PriorStop e NextStop do tipo TBusStop, para identificar quais são as

paradas e o tempo entre elas. Quando um ônibus passar da parada A para a parada B, levando 5 minutos, será inserido um registro nessa tabela. Quando passar da B para C, irá inserir outro registro. Quando outro ônibus passar da A para B novamente, e levar 7 minutos, é feita a média do que já está no banco com o tempo atual e atualizado o tempo desse registro, mantendo sempre a média dos tempos.

- TBusLine – esta classe mantém o cadastro da linha de ônibus. Possui um campo ID, uma descrição e uma propriedade BeaconItem, para identificar o *beacon* que está nesta linha de ônibus. Este *beacon* é utilizado para duas coisas: identificar que o usuário entrou no ônibus e saiu do ônibus, e fazer o cálculo entre as paradas. Como o cálculo é feito todo no aplicativo que está no celular do usuário, apenas é possível fazer o cálculo quando ele entra e sai do ônibus.
- TRoutes – esta classe mantém a rota de um ônibus. Possui um campo ID, uma propriedade BusLine para identificar o ônibus dessa rota e as propriedades PriorStop e NextStop do tipo TBusStop. Estas últimas servem para manter a parada anterior e a próxima parada da rota do ônibus. Então, se um ônibus 1 possui a rota A->B->C, terá 2 registros na tabela:
 - ID: 1 – BusLine: 1 – PriorStop: A – NextStop: B;
 - ID: 2 – BusLine: 1 – PriorStop: B – NextStop: C;
- TBusExitTime – esta classe mantém os horários de saída do ônibus. Possui um ID, a linha de ônibus (BusLine), um horário de saída e o dia da semana.

4.3.5 Funcionamento

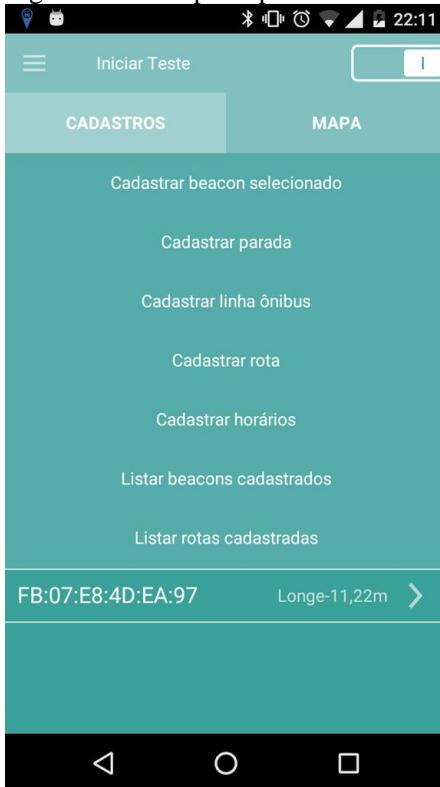
O principal recurso do sistema é o cálculo de tempos entre paradas. Para que isso seja possível deve existir um *beacon* em cada parada e um no ônibus. Dessa forma, quando o usuário se aproxima de uma parada, essa parada é persistida temporariamente. Se o usuário entra no ônibus, ou seja, se o sistema identificar que o usuário está perto de um *beacon* referente a um ônibus, é persistida a hora atual e a parada temporária anterior. O sistema realiza continuamente a procura por outras paradas. Quando ele acha a próxima parada, independente se o usuário saiu ou não do ônibus, é realizado o cálculo da parada anterior até a atual, sendo então persistido o tempo no banco (tabela ROUTE_TIME). Abaixo segue algoritmo:

1. Iniciar o escaneamento dos beacons;
2. Se encontrar uma parada, armazenar temporariamente;
3. Se encontrar determinado ônibus (*beacon*) e o usuário entrar (verificado através da proximidade), então é armazenada a hora atual temporariamente e o escaneamento continua;
 - a. Se encontrar uma parada e a parada temporária não for nula e o usuário ainda estiver no ônibus, é realizada a diminuição da hora atual em relação a hora inicial. Após isso, é realizada a atualização no banco, campo RouteTime, entidade ROUTES, com a média do tempo. O cálculo da média é feito da seguinte forma: $(\text{tempoDoBanco} + \text{mediaAtual}) / 2$. Se o campo do banco estiver vazio, insere a média atual (campo *mediaAtual*).
4. Guardar a parada atual temporariamente;
5. Voltar para o passo 4 até o usuário sair do ônibus.

O funcionamento do algoritmo utilizado é relativamente simples. A ideia é fazer a verificação por *beacons* constantemente, e quando achar algum realizar determinada ação. No passo 2 a parada é guardada em uma variável do sistema temporariamente. Isso significa que é criado uma instância de TBusStop recuperando do banco de dados o registro referente a esta parada, utilizando para a pesquisa, o UUID do *beacon* referente a esta parada. No passo 4 é verificado se a parada temporária não é nula para saber se esta não é a primeira parada encontrada. Outro ponto importante é parar a execução do sistema quando o usuário se afaste do ônibus para evitar cálculos indevidos caso o mesmo vá andando até outra parada de ônibus.

A Figura 15 mostra a tela principal do sistema. A partir desta tela são criadas outras, para cadastro, listagem, e outros testes do sistema. Para facilitar a implementação foi adicionado um botão para cada funcionalidade. Não foi levado em consideração como seria a *interface* gráfica com o usuário real da aplicação. A *interface* criada no sistema foi projetada para fazer testes que atendam aos requisitos do sistema.

Figura 15 - Tela principal do sistema MobileBus



Fonte: Autores

A tela principal do aplicativo é dividida em duas abas. A primeira, intitulada CADASTROS possui as funcionalidades do sistema. A segunda foi criada para auxiliar no teste do sistema, e possui mais duas abas, TESTE e HORÁRIOS, como mostra a Figura 16. A primeira tem a finalidade de auxiliar no teste do sistema, apresentado no Capítulo 5 deste trabalho. A outra lista os horários das linhas de ônibus de uma parada específica.

Figura 16 - Segunda aba principal do sistema, com a sub-aba TESTE selecionada



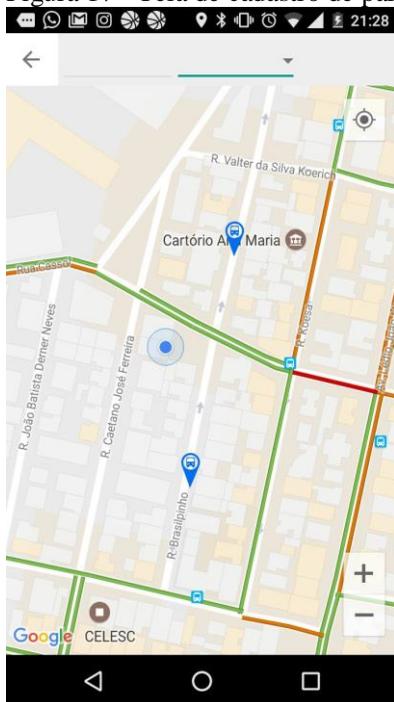
Fonte: Autores

Acima das abas existe o botão “Iniciar Teste”, que foi utilizado para avaliação do sistema, abordada no Capítulo 5 deste trabalho, e um *switch* ao lado do botão. Este *switch* irá ativar ou desativar o escaneamento de *beacons*. Quando o *scanner* detectar um *beacon* este adiciona algumas informações no lista (*listview*) abaixo dos botões. Este *listview* possui duas colunas; a primeira mostra o endereço *Media Access Control (MAC)* e a segunda a distância entre o *smartphone* e o dispositivo. Abaixo segue a lista de botões e suas funções:

- Cadastrar um *beacon* selecionado: cadastra o *beacon* selecionado no *listview* para ser utilizado posteriormente no cadastro de linhas de ônibus ou paradas. Para este cadastro é utilizada a classe *TBeaconItem*;

- Cadastrar uma parada de ônibus: abre a tela apresenta na Figura 16. Para mostrar as paradas de ônibus foi utilizado um componente disponibilizado pelo Delphi® chamado MapView. Este componente utiliza internamente a API do *Google Maps*® disponibilizada pela Google®. Para cada parada, foi adicionada uma marcação no mapa representado pelo ícone na Figura 18. O cadastro da parada é realizado através da classe TBusSop;
- Cadastrar uma linha de ônibus: cadastra uma linha de ônibus, com uma descrição e o *beacon* (TBeaconItem) desta linha;
- Cadastrar uma rota: nesta tela é cadastrada a rota de uma determinada linha. Para isso é selecionada uma determinada linha de ônibus e as paradas pela qual este ônibus passa. Este cadastro é realizado utilizando a classe TRoute. O cadastro das rotas é feito em pares de paradas, com exceção da primeira e última parada. A primeira parada tem o campo PriorStop com valor nulo, e a última com o campo NextStop com valor nulo;
- Cadastrar horários: aqui é realizado o cadastro de horários de uma determinada linha de ônibus, como demonstrado na Figura 19;
- Listar os *beacons* cadastrados: abre uma tela para mostrar a lista de *beacons* (TBeaconItem) cadastrados no sistema;
- Listar rotas cadastradas: lista as rotas cadastradas no sistema.

Figura 17 - Tela de cadastro de paradas de ônibus



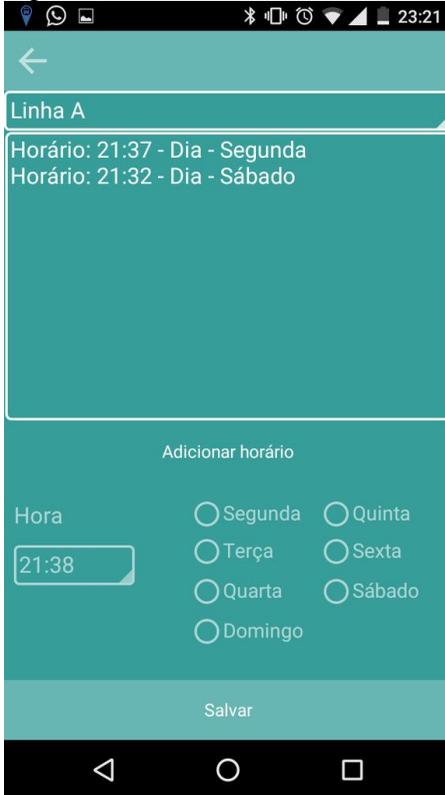
Fonte: Autores

Figura 18 - Ícone utilizado para marcar paradas de ônibus no mapa



Fonte: Autores

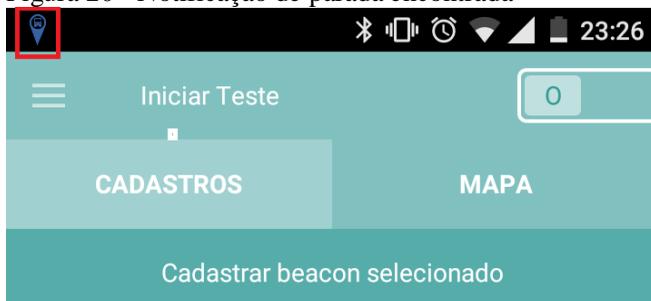
Figura 19 - Tela de cadastro de horários



Fonte: Autores

Para não ter a necessidade de o usuário estar com o sistema aberto para detectar os *beacons*, foi criado um serviço (*Android Service*) para fazer a verificação constante de *beacons*, e enviar uma notificação quando encontrar uma parada de ônibus próxima. A execução do serviço independe da execução da aplicação, não tem interface gráfica e nenhuma interação do usuário. A Figura 20 demonstra a notificação no *smartphone* do usuário, e na Figura 21 a notificação com a descrição “Parada de ônibus detectada. Clique para visualizar horários”.

Figura 20 - Notificação de parada encontrada



Fonte: Autores

Figura 21 - Descrição da notificação de parada encontrada



Fonte: Autores

5 APRESENTAÇÃO DOS RESULTADOS

5.1 CRIAÇÃO DO CENÁRIO

Para a avaliação do sistema proposto neste trabalho foi criado um cenário para o teste da aplicação. Este cenário foi aplicado em um ambiente fechado, visto que o servidor de aplicação foi desenvolvido localmente. Foram utilizados três *beacons*: um para simbolizar o que estaria no próprio ônibus, denominado LINHA A, e outros dois para representar as paradas, denominados, PONTO A e PONTO B, respectivamente.

O cadastro dos *beacons*, da linha de ônibus, das paradas, rotas e horários foi realizado utilizando suas respectivas telas no sistema. Ao cadastrar as linhas de ônibus e as paradas deve-se vincular o *beacon* com sua respectiva função dentro do sistema (parada de ônibus ou ônibus).

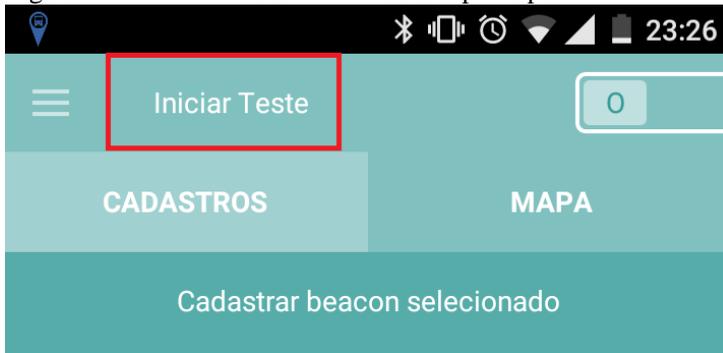
O cenário implementado objetiva simular o uso do aplicativo no dia-a-dia do usuário que utiliza o transporte público por ônibus. Com o aplicativo instalado em seu *smartphone*, quando o usuário chegar perto de uma parada de ônibus que esteja cadastrada no sistema (e tenha o *beacon* físico nela), é enviado uma notificação para o celular avisando que uma parada foi encontrada. Quando o usuário seleciona a notificação, o aplicativo apresenta todas as linhas e horários de ônibus que passam por aquela determinada parada. Como o cenário é simulado, e o banco de dados local, os testes não foram realizados em uma parada de ônibus real. Neste sentido, foi efetuado um teste de aproximação, bem como, a consulta no banco de dados das linhas e horários. Porém, o comportamento do sistema em uma parada de ônibus real teria o mesmo efeito.

Além da notificação, foi criada uma área específica no sistema para realizar o teste de cálculos de tempo entre paradas. Nesta área, apresentada na opção MAPA, subopção TESTE da tela principal, deve-se selecionar dois pontos cadastrados para a realização do teste. Depois de selecionado os pontos, o sistema irá atualizar as informações da rota a cada cinco segundos. Para o teste foram utilizados dois usuários, um para simular o comportamento de um usuário utilizando o ônibus, e o outro para monitorar as informações da média de tempo da rota sendo atualizadas a cada cinco segundos na opção TESTE.

Para iniciar o teste, ou seja, para começar a execução do algoritmo apresentado na seção 4.3.5 deste trabalho, foi criado um botão “Iniciar Teste” na tela principal, como mostra a Figura 22. Conforme o

algoritmo, quando a opção “Iniciar Teste” é selecionada, o sistema procura por *beacons* que estejam cadastrados visando detectar as paradas e realizar os cálculos de tempo.

Figura 22 - Botão "Iniciar Teste" da tela principal



Fonte: Autores

5.2 APRESENTAÇÃO DOS RESULTADOS

O teste do sistema focou em dois pontos principais: o primeiro foi o cálculo do tempo médio entre paradas e a apresentação desta média para outros usuários do sistema; e o segundo referente a notificação que é enviada ao usuário quando o mesmo se aproxima de uma parada.

Para o primeiro objetivo, foram dispostos dois *beacons* em lugares onde o sinal de um *beacon* não interferisse no sinal do outro, ou seja, para que fosse detectado apenas um *beacon* de cada vez. Para tal, o teste foi realizado dentro de uma residência (para que os dois usuários ficassem na mesma rede local), sendo que um dos *beacons* foi colocado em um cômodo, e o segundo em outro cômodo com ao menos 3 paredes separando os dois. A única regra era que o sinal dos dois *beacons* não poderia ser detectado ao mesmo tempo, simulando uma parada de ônibus real.

O usuário responsável pelo percurso da rota foi identificado de “primeiro usuário”. O outro, que estará monitorando as informações da rota na aba TESTE foi chamado de “segundo usuário”.

Para iniciar o teste, o segundo usuário escolhe a rota que quer acompanhar, na aba TESTE. Como só foi cadastrado uma rota, chamada de “LINHA A”, dos pontos PONTO A à PONTO B, esta foi selecionada, como mostra a Figura 23.

Figura 23 - Tela do segundo usuário, na tela para acompanhar as informações da rota



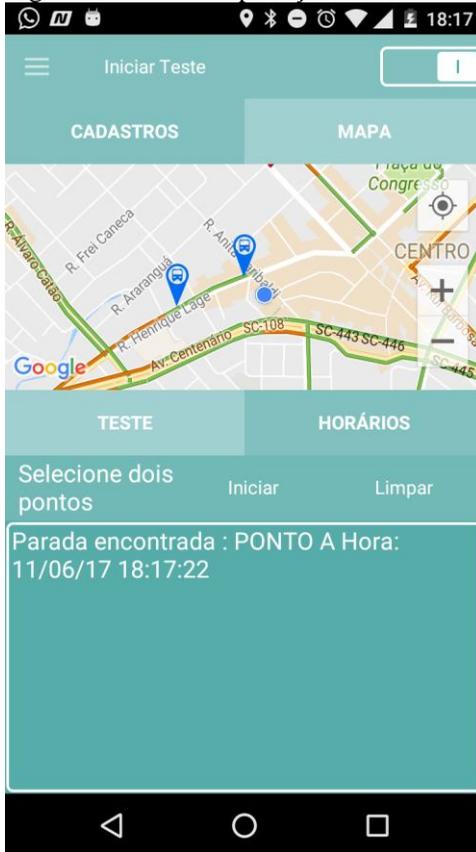
Fonte: Autor

A partir deste momento, as informações (tempo médio entre as paradas) apresentadas na Figura 24 são atualizadas a cada cinco segundos. Enquanto isso, o primeiro usuário pode iniciar o teste em “Iniciar teste” na tela principal, e deste modo, o sistema inicia a busca por *beacons*.

Ao iniciar o teste e ao aproximar-se de um *beacon* que esteja cadastrado como ponto de ônibus é disparado um evento no sistema para inserir informações na aba TESTE com o nome da parada e a data e a hora de quando esta foi encontrada, como mostra a Figura 24.

Internamente no sistema, estas informações são gravadas em variáveis temporárias para fazer os cálculos posteriormente.

Figura 24 - Tela da aplicação mostrando reconhecimento do Ponto A



Fonte: Autor

Após encontrar a parada, o sistema então espera até encontrar um *beacon* que esteja vinculado a um ônibus. É feita uma análise de proximidade entre o celular do usuário e o *beacon* para saber se ele realmente entrou no ônibus. Como dito anteriormente, esta verificação seria mais correta ser feita com outras tecnologias, por exemplo, através de NFC. Após chegar próximo suficiente do dispositivo, um evento é disparado no sistema, em que um *timer* é iniciado visando marcar o tempo do percurso até outro ponto da rota. Além disso, o aplicativo

mostra uma mensagem ao usuário dizendo que foi encontrado um ônibus com o seu nome, data e hora como mostra a Figura 25.

Figura 25 - Tela da aplicação mostrando reconhecimento da Linha A



Fonte: Autor

Após detectar que o usuário entrou no ônibus, simulou-se o percurso que o ônibus faria até a segunda parada, PONTO B. Assim que o sistema detecta a segunda parada (Figura 26), é atualizado o tempo médio da rota utilizando o tempo que levou entre as duas paradas. Ou seja, é utilizada a classe `TRuote` para atualizar o campo `BusRouteTime`, cuja rota tenha como `PriorStop` o PONTO A, e `NextStop` o PONTO B. Para o cálculo da média é realizada a soma do tempo atual da rota com o valor que está no banco de dados, dividido por 2. Esta média não é aritmética por conta da escalabilidade do sistema. Em um sistema em produção, por exemplo, após pouco tempo de funcionamento haveria a necessidade de realizar a divisão na casa dos milhares, visto que para a

média aritmética é utilizada a quantidade. Depois de calcular e atualizar os valores no banco de dados, é apresentada a mensagem para o usuário com o valor da média atualizado, como mostra a Figura 27.

Figura 26 - Tela da aplicação mostrando reconhecimento do Ponto B



Fonte: Autores

Figura 27 - Tela do segundo usuário apresentando o tempo atualizado



Fonte: Autores

Enquanto isso, a cada cinco segundos, o sistema atualiza o valor da média do segundo usuário, que está acompanhando a rota, como demonstrou a Figura 28. Desta forma, todos os usuários têm o valor da média atualizada de uma determinada rota.

Durante os testes do aplicativo, notou-se que por estar em um ambiente com muitas paredes, o cálculo da distância dos *beacons* com o aplicativo demonstrou pouca precisão. Segundo Dong e Dargie (2012) esta imprecisão é por fatores como reflexão, refração, difração e espalhamento de ondas causadas por objetos próximos. Uma parede, por exemplo, pode reduzir o sinal em aproximadamente 3db (HUANG; BARRALET; SHARMA, 2009). Estas interferências no sinal podem ser observadas na Figura 28, onde o dispositivo estava a poucos metros de distância, mas por conta das interferências o cálculo foi para 50 metros. Em um ambiente sem muitas paredes não foi notado o problema.

Figura 28 - Tela da aplicação mostrando dados do *beacon*



Fonte: Autores

Para o segundo objetivo, o envio da notificação para o usuário, foi utilizado um serviço do Android®, como já mencionado. Este serviço possui apenas duas funções: fazer a busca constante por *beacons* e enviar a notificação caso encontre algum *beacon*. A classe utilizada pelo Delphi® para mostrar a notificação possui algumas propriedades, dentre elas a *Description*, para a descrição da notificação para o usuário, e a *AlertAction*, que é uma propriedade do tipo *string*, utilizada para enviar o UUID do *beacon* para a aplicação. Quando o usuário seleciona a notificação, e o sistema abre a mesma, é disparado um evento no objeto de notificação do sistema e os valores das propriedades descritas acima ficam disponíveis para serem utilizados.

Com o valor do UUID do *beacon* é então realizada uma pesquisa no banco de dados para determinar e apresentar para o usuário os horários de todas as linhas cujas rotas passem por determinada parada.

A Figura 29 apresenta o sistema depois que determinada notificação é selecionada. Neste momento, é habilitada a opção

HORARIOS a partir da opção MAPA. A cor do marcador no mapa do ponto que foi detectado é alterada para verde, e os horários são adicionados na lista, agrupados pela linha de ônibus. Foi também adicionada uma caixa de texto para fazer buscas na lista.

Figura 29 - Tela mostrando o sistema depois que usuário seleciona a notificação



Fonte: Autor

O processamento da atualização é realizado em *threads* e *tasks*, fazendo com que o sistema não fique travado enquanto é realizada a pesquisa pelos horários. Esta programação paralela é vital no desenvolvimento de aplicações na plataforma Android®, pois caso a aplicação trave durante a interação com o usuário, o sistema operacional apresenta uma mensagem informando que a aplicação parou de funcionar.

6 CONSIDERAÇÕES FINAIS

O objetivo principal deste trabalho reside no desenvolvimento de um sistema voltado ao usuário de transporte público urbano via ônibus. Este sistema se comunica com sensores disponibilizados nas paradas de ônibus e no próprio ônibus, fazendo com que dados sejam capturados e disponibilizados ao usuário no aplicativo, como por exemplo, a média de tempo de percursos dos ônibus entre as paradas.

As cidades atualmente estão passando por um intenso processo de urbanização, fazendo com que os gestores públicos tenham dificuldade em definir políticas capazes de acompanhar este crescimento. Como consequência, o provimento de serviços públicos como o transporte público urbano, foco deste trabalho, se mostra pouco satisfatório. Manifestação como a ocorrida no Brasil em 2013 demonstram que as pessoas não estão satisfeitas com o transporte, seja pela qualidade, seja pelos preços das passagens.

Para atingir os objetivos propostos neste trabalho, foi realizada uma pesquisa bibliográfica sobre *Smart Cities* e Internet das Coisas. A ideia do sistema foi baseada nestes dois conceitos, que estão fortemente ligados, como foi demonstrado durante a pesquisa. Já os sensores dispostos nas cidades vêm do paradigma da Internet das Coisas. Pelo fato de serem utilizados em prol dos cidadãos de uma cidade para melhorar de alguma forma o dia a dia destes, no caso deste trabalho o uso no transporte público, ocorre a interconexão com o conceito de *Smart City*, mais especificamente o componente *Smart Mobility*.

O desenvolvimento do sistema é basicamente composto por uma aplicação móvel para o sistema operacional Android® e um servidor de banco de dados local. A aplicação em questão foi nomeada MobileBus, e foi desenvolvida utilizando a plataforma de desenvolvimento Delphi®. Foram utilizadas tecnologias disponíveis para esta plataforma, como foi o caso dos *frameworks* Aurelius® e do RemoteDB®. Estes *frameworks* permitiram manter o foco no comportamento do sistema através de classes, ao invés de ter que se preocupar, por exemplo, com as estruturas do banco de dados e com consultas SQL. O primeiro *framework* proveu um sistema de mapeamento de classes em estruturas relacionais (ORM) e o segundo possibilitou a utilização de um banco de dados remoto (LAN) por uma estrutura HTTP.

A aplicação desenvolvida identifica sensores *bluetooth* para gerar dados que são apresentados ao usuário, e também mantém cadastradas paradas de ônibus, linhas de ônibus e horários. Além disso, o sistema também é capaz de enviar notificações ao usuário quando o mesmo se

aproxima de uma parada, e apresenta todos os horários de todas as linhas de ônibus que passam por esta determinada parada. Essa praticidade pode fazer diferença para o usuário, economizando tempo e estimulando os mesmos a utilizarem mais este meio de transporte.

Após a análise do aplicativo os resultados obtidos podem ser considerados adequados. O serviço *Android Service*® desenvolvido permitiu a descoberta de *beacons* sem a interação com o usuário, porém faltou a realização do controle da quantidade de vezes que a notificação é enviada. Outra deficiência refere-se ao cálculo de tempo entre as paradas sendo este estabelecido na aplicação principal ao invés de um serviço. Para os testes foi elaborado um cenário para a aplicação, sendo utilizados três *beacons*, um para simular o ônibus e os outros dois simulando paradas de ônibus. Desta forma, a rota no cenário é representada por dois pontos de ônibus. Também foram utilizados dois *smartphones* no teste, um para simular o usuário e fazer os cálculos de tempo, e o outro para acompanhar a atualização dos tempos. A média de tempo ficou disponível para o outro usuário como esperado, sendo atualizada a cada iteração, ou seja, cada vez que o primeiro usuário terminava a rota.

Um dos principais pontos do projeto foi a pesquisa em uma área que vem se desenvolvendo e possui potencial para crescimento. É notável a falta de investimento em transporte público em grandes cidades. Por isso, projetos e pesquisas nessa área são de grande importância para a população.

6.1 TRABALHOS FUTUROS

Como trabalhos futuros foram identificadas possibilidades de utilizar outras tecnologias para comunicação além dos *beacons*, como por exemplo, NFC ou RFID. Estas tecnologias são as mais indicadas, por exemplo, para identificar que o usuário está no ônibus. Em um cenário ideal, o usuário já pagaria sua passagem utilizando uma destas tecnologias e esta informação poderia ser utilizada no contexto deste trabalho.

Além disso, também seria importante realizar a média do tempo entre paradas de forma mais inteligente. Como mencionado no trabalho, o tempo que um ônibus leva entre uma parada e outra não é o mesmo em diferentes dias da semana e entre diferentes horários do dia.

Também seria importante adaptar o sistema para o usuário final, com integração com redes sociais como o Facebook® para realizar o cadastro na aplicação. Com o cadastro do usuário, seria possível realizar

o cadastro de pontos e linhas de ônibus favoritas, criar configurações para, por exemplo, apenas enviar notificações para o usuário de paradas que estivessem em sua lista de favoritos.

Com as linhas favoritas do usuário cadastradas no sistema, poderia ter uma configuração para, na hora de abrir o sistema a partir da notificação da parada de ônibus, mostrar apenas os horários das suas linhas favoritas. Além desta configuração, seria possível também definir um valor máximo de tempo para mostrar os horários a partir da notificação. Como exemplo, se a notificação foi enviada às 13:00h e o usuário estivesse configurado para mostrar os horários com no máximo 30 minutos de intervalo, seria realizada uma pesquisa no banco de dados por horários de linhas entre 12:30h e 13:30h. Não faria muito sentido apresentar os horários de ônibus que passam por esta parada às 18:00h se o usuário está ali às 13:00h.

Outro ponto importante seria a utilização de um dispositivo GPS no ônibus, além do *beacon*, para mostrá-lo no mapa para o usuário. Desta forma, o usuário poderia acompanhar onde o ônibus está além das médias de tempo que ele leva na rota escolhida.

Seria interessante também utilizar o conceito de Sistemas de Recomendação para recomendar rotas e linhas de ônibus alternativas para os usuários de acordo com suas rotas diárias.

Por fim, identifica-se a possibilidade de utilização de técnicas de *Machine Learning* para realizar previsões de tempo de determinada rota escolhida pelo usuário. Neste sentido, a medida que o aplicativo fosse utilizado, os dados de tempo entre as paradas poderiam ser utilizados para fazer previsões de acordo com determinada hora do dia, ou determinado dia da semana, e assim por diante.

REFERENCIAS

ALBINO, V.; BERARDI, U.; DANGELICO, R. M. Smart Cities: Definitions, Dimensions, Performance, and Initiatives. **Journal of Urban Technology**, v. 22, n. 1, p. 3–21, 2015.

ANTUNES, E. M. Engenharia urbana aplicada: um estudo sobre a qualidade do transporte público em cidades médias. *Revista Brasileira de Gestão Urbana (Brazilian Journal of Urban Management)*, v. 5, n. 2, p. 51–62, 2013.

Arquitetura multicamadas. Disponível em: <https://en.wikipedia.org/wiki/Multitier_architecture>. Acesso em 14 de Maio de 2017.

ASHTON, K. That “Internet of Things” Thing. **RFID Journal**, p. 4986, 2009.

ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A survey. **Computer Networks**, v. 54, n. 15, p. 2787–2805, 2010.

BARRIONUEVO, J. M.; BERRONE, P.; RICART, J. E. Smart Cities, Sustainable Progress. **IESE Insight**, v. 14, n. 14, p. 50–57, 2012.

BBC. **Como o Google ganha dinheiro?** Disponível em: <http://www.bbc.com/portuguese/noticias/2016/03/160329_google_dinheiro_fn>. Acesso em: 18 abr. 2017.

BERTINO, Elisa; ISLAM, Nayeem. Botnets and Internet of Things Security. **Computer**, [s.l.], v. 50, n. 2, p.76-79, fev. 2017. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/mc.2017.62>.

Bluetooth Low Energy. Disponível em: <<https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/low-energy>>. Acesso em 25 de Agosto de 2016.

BLUETOOTH. Disponível em: <<https://en.wikipedia.org/wiki/Bluetooth>>. Acesso em: 01 mar. 2017.

CARAGLIU, A.; DEL BO, C.; NIJKAMP, P. Smart Cities in Europe. **Journal of Urban Technology**, v. 18, n. 2, p. 65–82, 2011.

CENEDESE, A. et al. Padova smart City: An urban Internet of Things experimentation. **Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, WoWMoM 2014**, 2014.

CHANG, Kuor-hsin. Bluetooth: a viable solution for IoT? [Industry Perspectives]. **Ieee Wireless Communications**, [s.l.], v. 21, n. 6, p.6-7, dez. 2014. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/mwc.2014.7000963>.

CRETU, L.-G. Smart Cities Design using Event-driven Paradigm and Semantic Web. **Informatica Economica**, v. 16, n. 4, p. 57–67, 2012.

Dados abertos. Disponível em: <https://pt.wikipedia.org/wiki/Dados_abertos>. Acesso em 06 de Março de 2017.

DECUIR, Joseph. The Story of the Internet of Things: Issues in utility, connectivity, and security.. **Ieee Consumer Electronics Magazine**, [s.l.], v. 4, n. 4, p.54-61, out. 2015. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/mce.2015.2463292>.

DONG, Q.; DARGIE, W. Evaluation of the reliability of RSSI for indoor localization. **2012 International Conference on Wireless Communications in Underground and Confined Areas, ICWCUCA 2012**, p. 2–7, 2012.

ELBOUANANI, Salim; KIRAM, My Ahmed El; ACHBAROU, Omar. Introduction to the Internet of Things security: Standardization and research challenges. **2015 11th International Conference On Information Assurance And Security (ias)**, [s.l.], p.32-37, dez. 2015. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/isiias.2015.7492741>.

EVANS, D. A Internet das Coisas Como a próxima evolução da Internet está mudando tudo. p. 5–7, 2011.

EVANS, Dave. INTERNET DAS COISAS: como a próxima evolução da Internet está mudando tudo. **Cisco Internet Business Solutions Group (IBSG). San Jose, Canada**, 2011.

FLOERKEMEIER C., LAMPE M. Issues with RFID Usage in Ubiquitous Computing Applications. **Lecture Notes in Computer Science**, vol 3001. Springer, Berlin, Heidelberg, 2004.

Frota municipal de veículos. Disponível em: < <http://cidades.ibge.gov.br/painel/frota.php>>. Acesso em 29 de Agosto de 2016.

GIFFINGER, R. Smart cities Ranking of European medium-sized cities. **October**, v. 16, n. October, p. 13–18, 2007.

GOMEZ, Carles et al. **Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology.** Sensors, [s.l.], v. 12, n. 12, p.11734-11753, 29 ago. 2012. MDPI AG. <http://dx.doi.org/10.3390/s120911734>.

GUPTA, Krishnakanth; SHUKLA, Sapna. Internet of Things: Security challenges for next generation networks. **2016 International Conference On Innovation And Challenges In Cyber Security (iciccs-inbush)**, [s.l.], p.315-318, fev. 2016. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/iciccs.2016.7542301>.

HARRISON, C. AND DONNELLY, I. A. A Theory of Smart Cities. **Proceedings of the 55th Annual Meeting of the ISSS - 2011, Hull, UK**, n. Proceedings of the 55th Annual Meeting of the ISSS, p. 1–15, 2011.

HUANG, X.; BARRALET, M.; SHARMA, D. Accuracy of Location Identification with Antenna Polarization on RSSI. n. March 2009.

iBeacon Parameters. Disponível em: < <https://support.kontakt.io/hc/en-gb/articles/201620741-iBeacon-Parameters-UUID-Major-and-Minor>>. Acesso em 15 de Maio de 2017.

KUROSE, John F.; ROSS, Keith W.. **Redes de computadores e a Internet: Uma abordagem top-down.** 5. ed. São Paulo: Pearson Education do Brasil, 2010. 576 p.

KWON, Giwon et al. Bluetooth low energy security vulnerability and improvement method. **2016 Ieee International Conference On**

Consumer Electronics-asia (icce-asia), [s.l.], p.1-4, out. 2016. IEEE. <http://dx.doi.org/10.1109/icce-asia.2016.7804832>.

LACERDA, F.; LIMA-MARQUES, M. Da necessidade de princípios de Arquitetura da Informação para a Internet das Coisas. **Perspectivas em Ciência da Informação**, v. 20, n. 2, p. 158–171, 2015.

LACERDA, Flavia; LIMA-MARQUES, Mamede. Da necessidade de princípios de Arquitetura da Informação para a Internet das Coisas. **Perspectivas em Ciência da Informação**, [s.l.], v. 20, n. 2, p.158-171, jun. 2015. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/1981-5344/2356>.

LAZAROIU, G. C.; ROSCIA, M. Definition methodology for the smart cities model. **Energy**, v. 47, n. 1, p. 326–332, 2012.

Lifelong learning. Disponível em: <https://en.wikipedia.org/wiki/Lifelong_learning>. Acesso em 01 de Março de 2017.

Linking Open Data cloud diagram 2017, de Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak. Disponível em <<http://lod-cloud.net/>>. Acesso em: 09 de junho de 2017.

LOMBARDI, P. et al. Modelling the smart city performance. **Innovation: The European Journal of Social Science Research**, v. 25, n. 2, p. 137–149, 2012.

MÁRCIO ROGÉRIO SILVEIRA, R. G. C. Transporte público, mobilidade e planejamento urbano: contradições essenciais. **Estudos Avançados**, v. 27, n. 79, p. 41–53, 2013.

NIYATO, Dusit et al. Economics of Internet of Things: an information market approach. **Ieee Wireless Communications**, [s.l.], v. 23, n. 4, p.136-145, ago. 2016. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/mwc.2016.7553037>.

ONU projeta que população mundial chegue aos 8,5 milhões em 2030. Disponível em: <<http://www.unric.org/pt/actualidade/31919-onu-projeta-que-populacao-mundial-chegue-aos-85-mil-milhoes-em-2030>>. Acesso em 15 de Agosto de 2016.

Patent application. Disponível em: < https://en.wikipedia.org/wiki/Patent_application>. Acesso em 01 de Março de 2017.

PERERA, Charith; LIU, Chi Harold; JAYAWARDENA, Srimal. A Survey on Internet of Things From Industrial Market Perspective. **Ieee Access**, [s.l.], v. 2, p.1660-1679, 2014. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/access.2015.2389854>.

PRADO, K. C. D.; SANTOS, P. E. DOS. Smart Cities: Conceito, Iniciativas E O Cenário Carioca. p. 123, 2014.

Projeção da população do Brasil e das Unidades da Federação. Disponível em: < <http://www.ibge.gov.br/apps/populacao/projecao/>>. Acesso em 29 de Agosto de 2016.

RASOOLIMANESH, S. M.; BADARULZAMAN, N.; JAAFAR, M. Achievement to Sustainable urban Development using City Development Strategies (CDS): A Comparison between Cities Alliance and the World Bank definitions. **Journal of Sustainable Development**, v. 4, n. 5, 2011.

RAY, Partha Pratim; AGARWAL, Sneha. Bluetooth 5 and Internet of Things: Potential and architecture. **International conference on Signal Processing, Communication, Power and Embedded System (SCOPE)-2016**, Paralakhemundi, Odisha, India, India, p. 1461-1465, 2./jul. 2017.

RemoteDB Documentation. Disponível em: < <http://www.tmssoftware.com/site/remotedb.asp>>. Acesso em 14 de Maio de 2017.

SCHAFFERS, H.; RATTI, C.; KOMNINOS, N. Special issue on smart applications for smart cities - new approaches to innovation: Guest editors' introduction. **Journal of Theoretical and Applied Electronic Commerce Research**, v. 7, n. 3, 2012.

SCHUURMAN, D. et al. Smart ideas for smart cities: Investigating crowdsourcing for generating and selecting ideas for ICT innovation in a city context. **Journal of Theoretical and Applied Electronic Commerce Research**, v. 7, n. 3, p. 49–62, 2012.

SHAPIRO, J. Smart cities: quality of life, productivity, and the growth effects of human capital. **The review of economics and statistics**, v. v88(2,May), p. 324–335, 2006.

SURESH, P.; DANIEL, J. V.; ASWATHY, R. H. A state of the art review on the Internet of Things (IoT) History , Technology and fields of deployment. 2014.

TANENBAUM, Andrew S.; WETHERALL, David. **Redes de computadores**. 5. ed. São Paulo: Pearson Education do Brasil, 2013.

TELECO. **Redes Sem Fio: Tecnologias de Redes Sem Fio**. Disponível em:

<http://www.teleco.com.br/tutoriais/tutorialredespaid/pagina_3.asp>.

Acesso em: 16 mar. 2017.

TMS Software. Aurelius. Disponível em: <<http://www.tmssoftware.com/site/aurelius.asp>> Acesso em: 15 de maio de 2017.

UN-Habitat Global Activities Report 2015. Disponível em: <<https://sustainabledevelopment.un.org/content/documents/1726Habitat%20Global%20Activities%202015.pdf>>. Acesso em 15 de Agosto de 2016.

UNITED NATIONS. World Population Prospects: The 2012 Revision. Highlights and Advance Tables. **Population and development review**, v. 36, p. 775–801, 2013.

US DOT. Smart City Challenge: Addressing the Challenges of Today and Tomorrow. 2016. Disponível em: <<https://www.transportation.gov/sites/dot.gov/files/docs/Smart%20City%20Challenge%20Lessons%20Learned.pdf>>. Acesso em: 10 de junho de 2017.

ZANELLA, A. et al. Internet of Things for Smart Cities. **IEEE Internet of Things Journal**, v. 1, n. 1, p. 22–32, 2014.

ZANELLA, Andrea et al. Internet of Things for Smart Cities. **Ieee Internet Of Things Journal**, [s.l.], v. 1, n. 1, p.22-32, fev. 2014. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/jiot.2014.2306328>.

ZHENG, Li. ZigBee Wireless Sensor Network in Industrial Applications. **2006 Sice-icase International Joint Conference**, [s.l.], p.1067-1070, 2006. IEEE. <http://dx.doi.org/10.1109/sice.2006.315751>.

ZHUHADAR, L. et al. The next wave of innovation - Review of smart cities intelligent operation systems. **Computers in Human Behavior**, v. 66, p. 273–281, 2017.

MADLMAYR, Gerald et al. NFC Devices: Security and Privacy. **2008 Third International Conference On Availability, Reliability And Security**, [s.l.], p.642-647, mar. 2008. IEEE. <http://dx.doi.org/10.1109/ares.2008.105>.

JUELS, A.. RFID security and privacy: a research survey. **Ieee Journal On Selected Areas In Communications**, [s.l.], v. 24, n. 2, p.381-394, fev. 2006. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/jsac.2005.861395>.